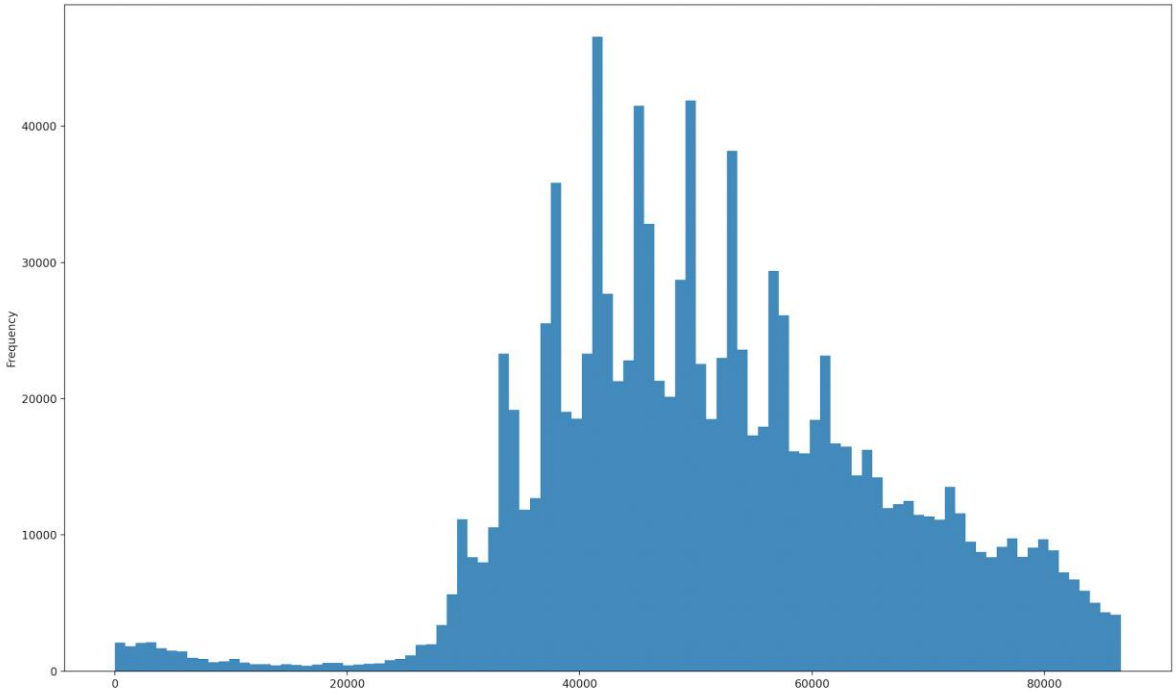
CIS 4930 -- Mobile Networks Experiment 2
Group 9: Leeson Chen, Andy Liu, Ricky Clarke
March 11 2020

Part I.

1. Produce the time series plot of events in "outputwireless-logs-20120409.DHCP_ANON"
•x-axis: Time (15-minute bins)
•y-axis: number of events in the bin



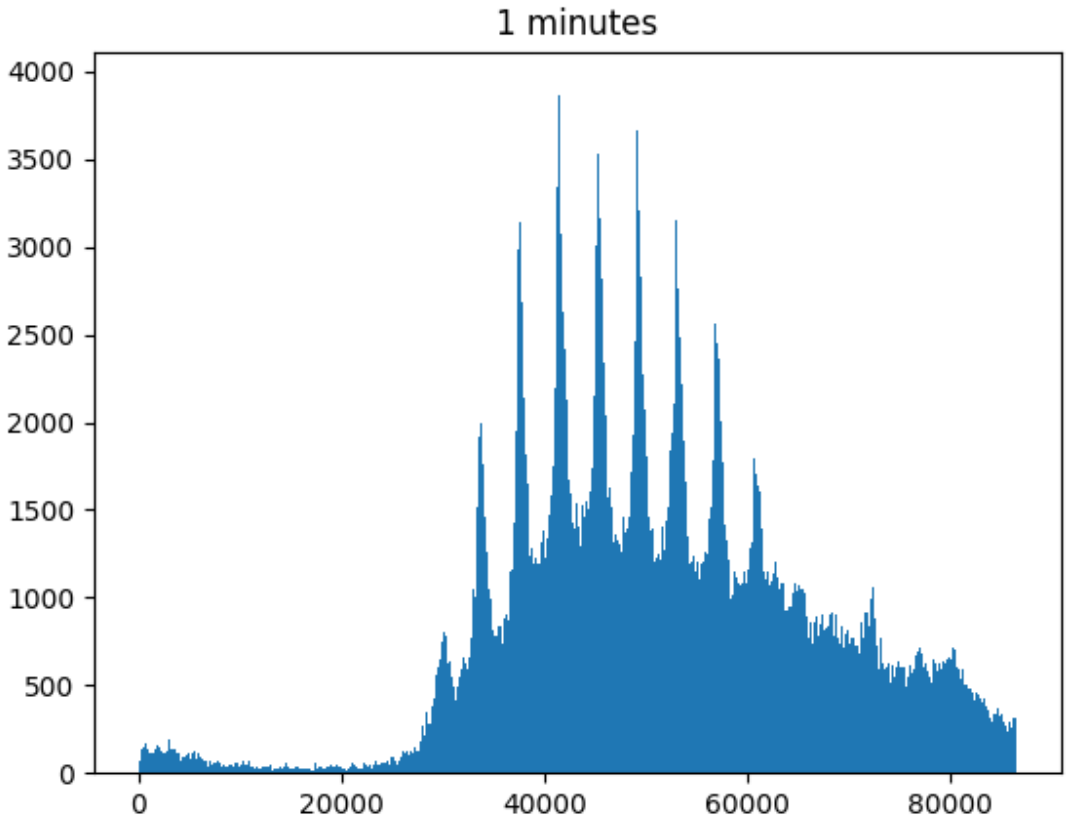
2. What time of the day is the most active (in terms of number of events)?
When divided into 15-minute bins, the most active time of day is 11:30-11:45 AM. This is likely when most people have on-campus classes and are walking to-and-from their classrooms.

3. What time of the day is the least active (in terms of number of events)?
When divided into 15-minute bins, the least active time of day is 4:00-4:15 AM and 5:00-5:15 AM. This is likely because most on-campus facilities have closed by then, and the late-night studying students in the libraries have left (Marston Library closes at 2 AM).

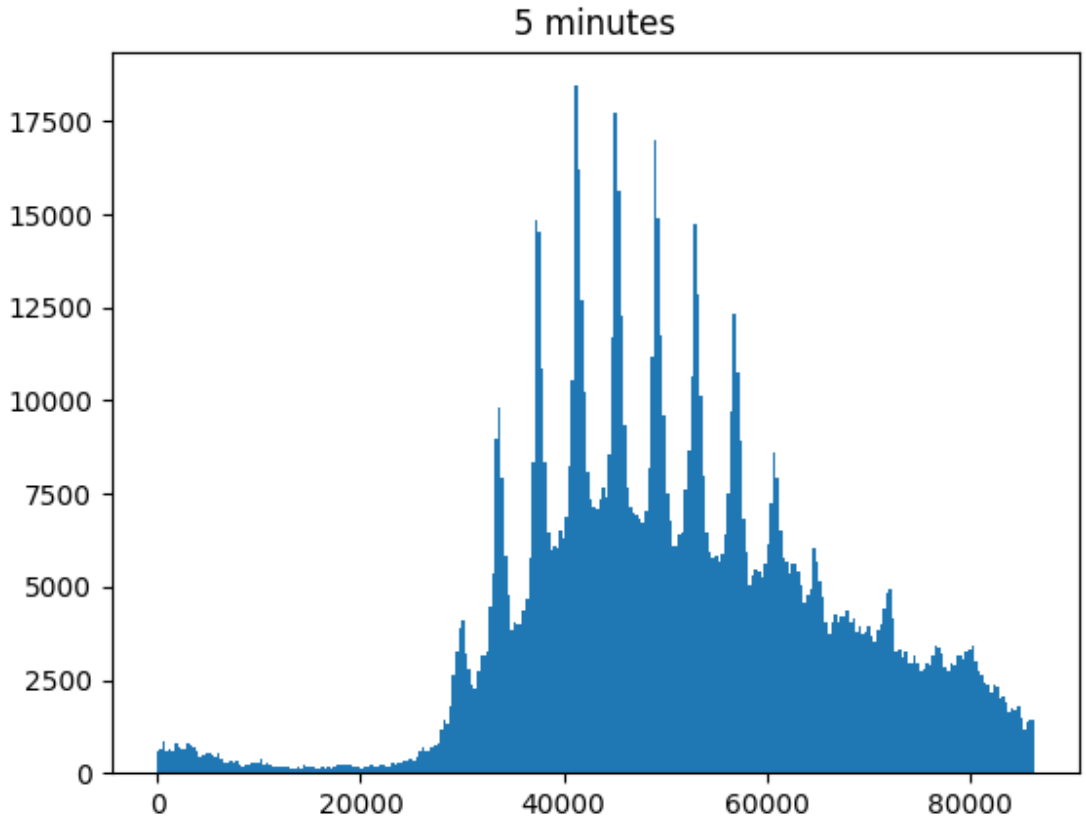
4. Considering the class periods (start/end times), please explain your observations in the time series around beginning or end of classes.
Every hour there is a sharp spike in the number of students. This can be explained by classes ending and beginning, and students walking between classrooms.

5. Try different bins (1m, 5m, 10m, 30m, 60m) and discuss how it affects your analyses.
As the bin interval decreases, the spikes in the number of students become easier to distinguish and more specified in time period. As the interval increases, the specific intervals of spikes are lost and the only data that is easy to distinguish is the busiest times of the day, rather than the class periods.

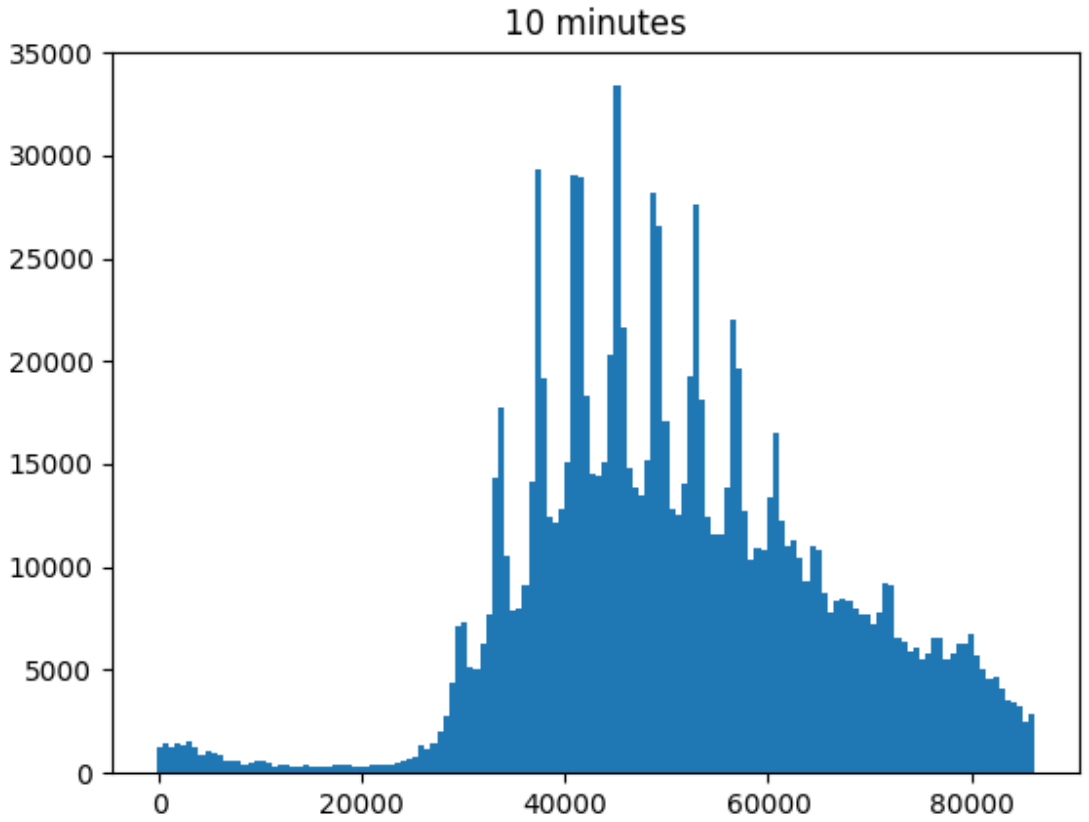
1 minute graph:



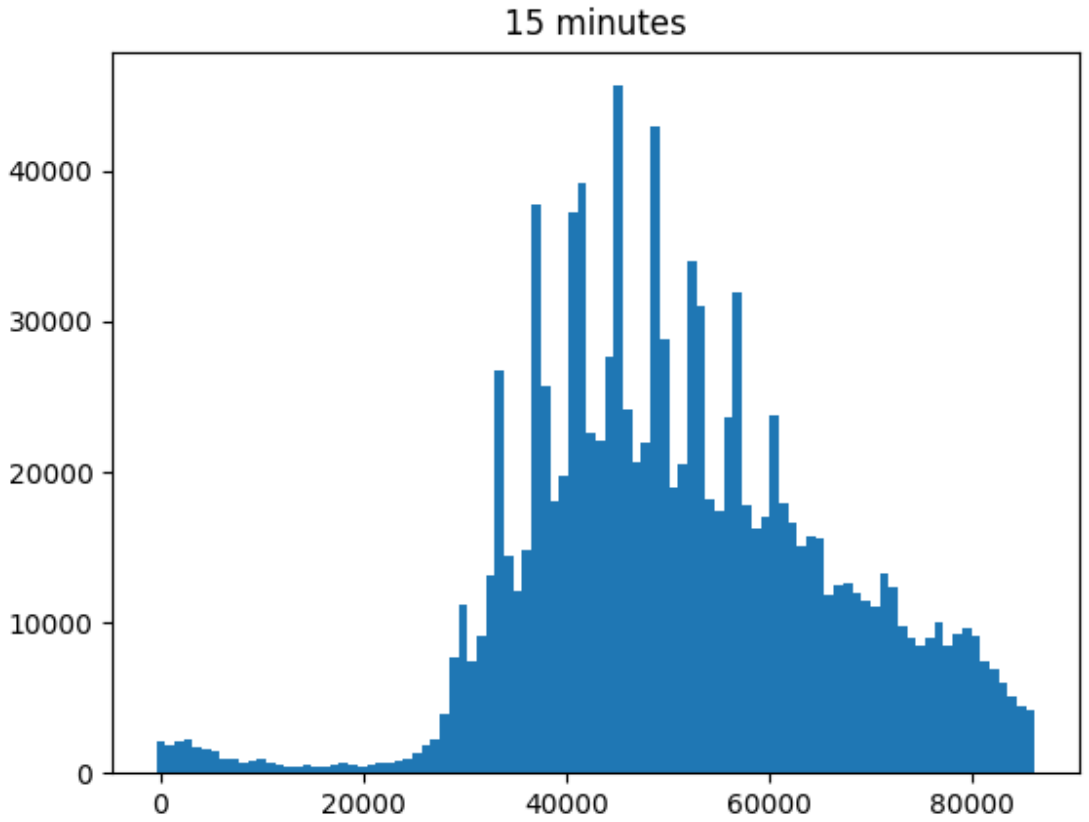
5 minute graph:



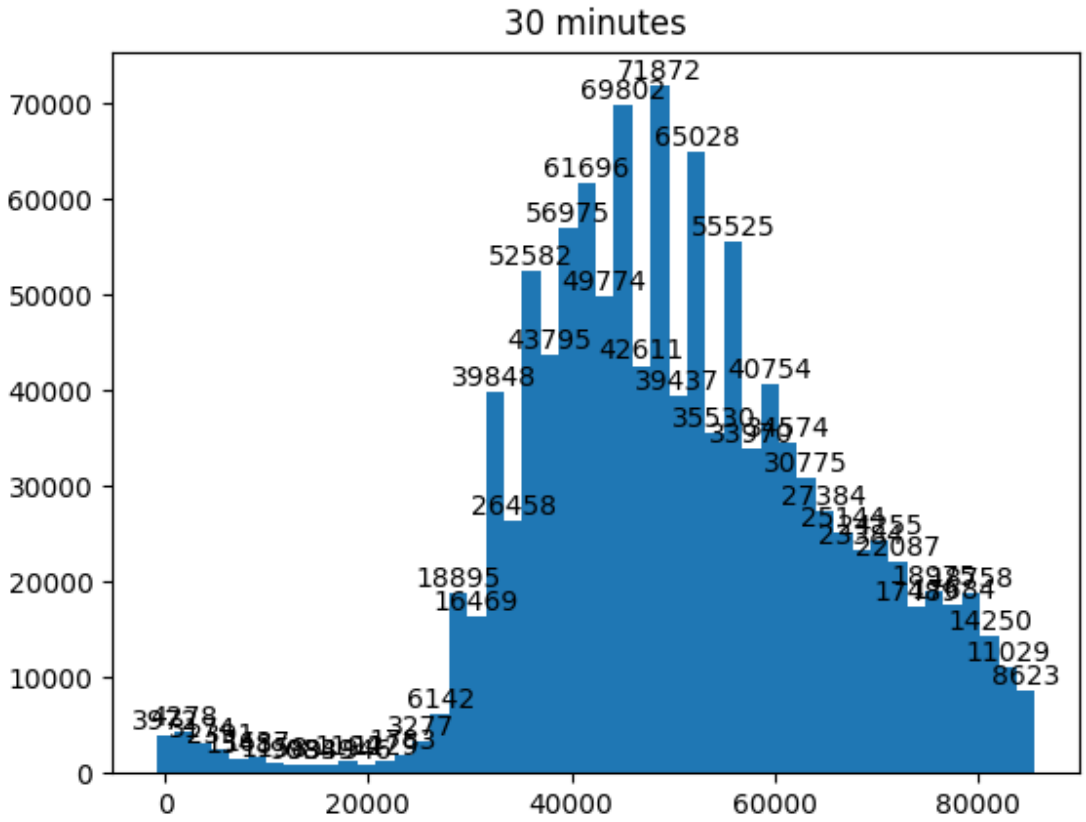
10 minute graph:



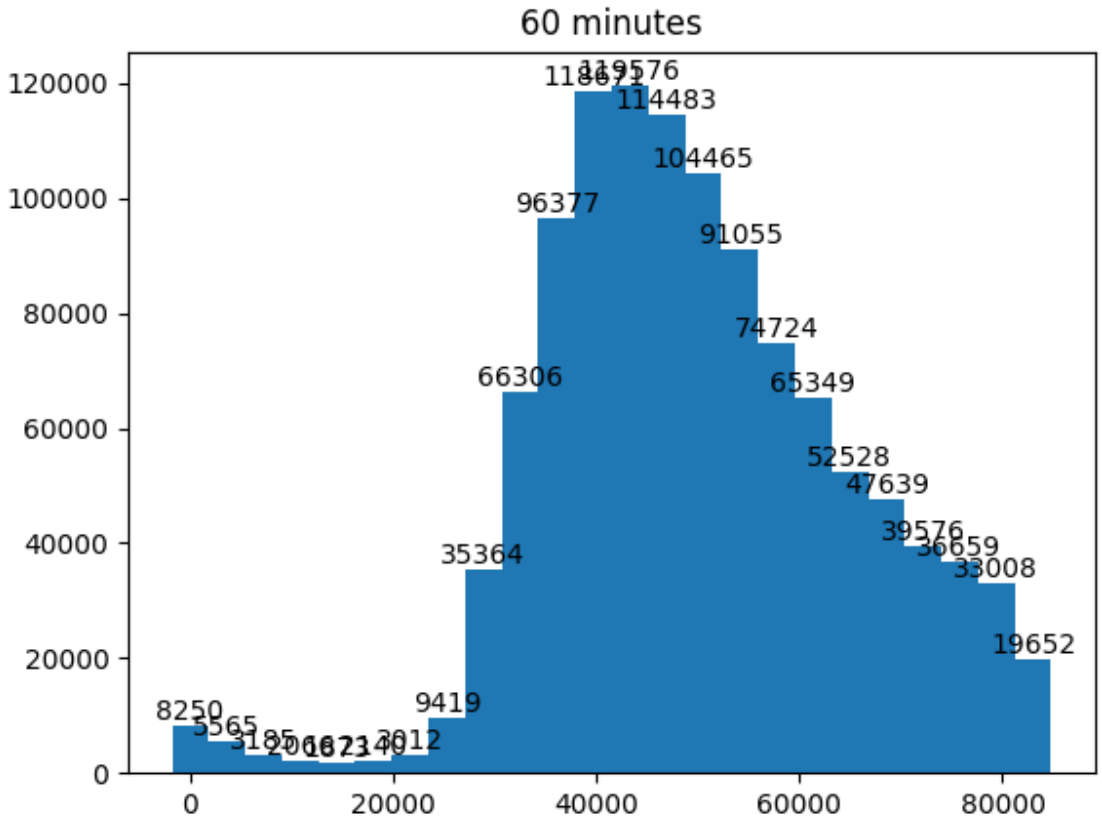
15 minute graph:



30 minute graph:



60 minute graph:



Part II.

1. Repeat the same tasks of Task 1, on "outputwireless-logs-20120407.DHCP_ANON"

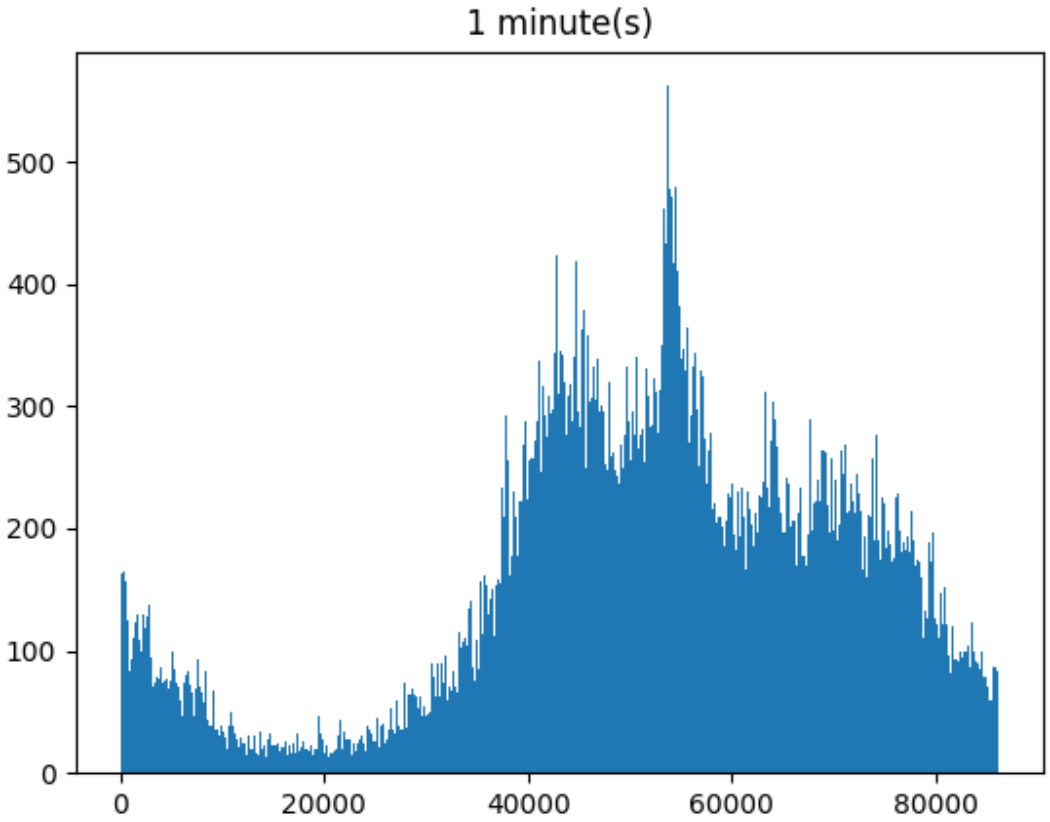
2. In addition, do you notice any similarities or differences?

• Please explain how you observed them and why you think these similarities and differences exist.

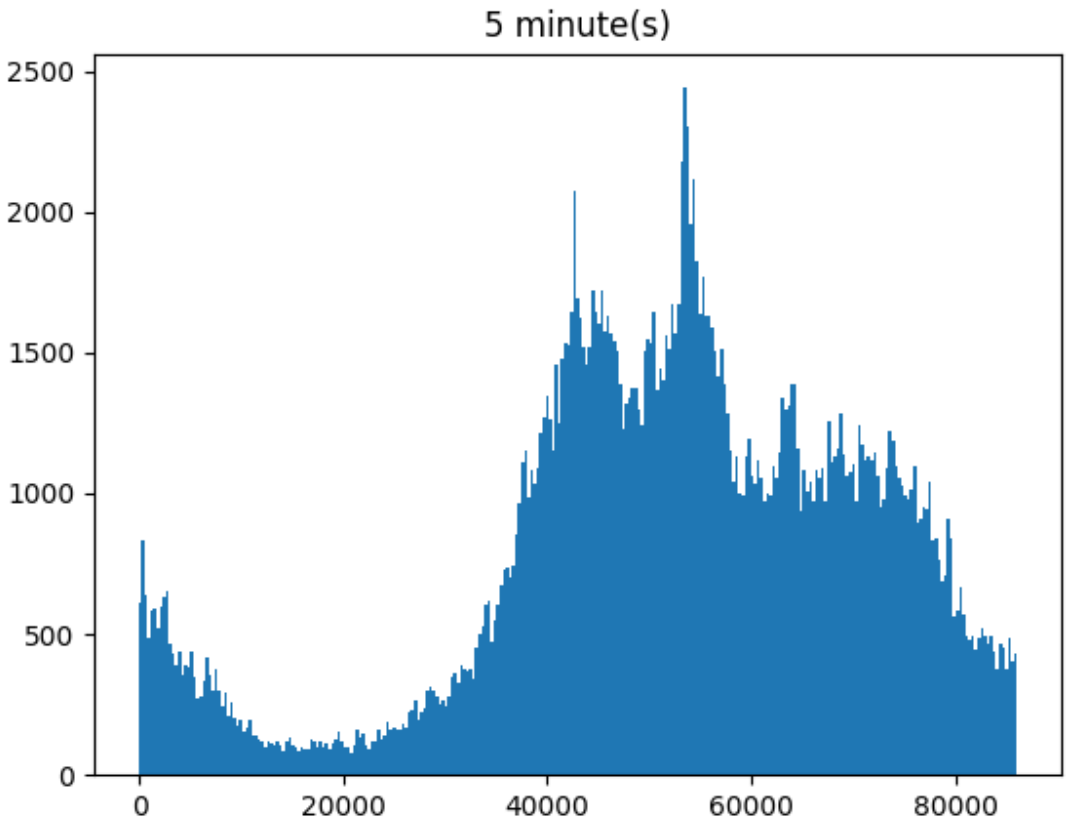
• Include any supporting graphs and plots

The previous data for Part 1 was collected on 4/9/2012, which was a Monday--explaining why the frequency of students correlates to class schedule. However, this data set is from 4/7/2012, which is a Saturday, when there is no class. This explains the lack of "spikes" indicating when class periods end and students appear in large numbers. When looking at the data graphed in part 1, the student spikes appear to indicate certain periods that class starts and ends. The data in part 2 do not have spikes at the class periods, indicating that the collection date must have been on the weekend where there are no classes being held. Similarities that the data share display a major peak at the center of the day (noon) most likely being an time interval that most students work within. However, the smoothed curve still shows a similar daily overall schedule, where students appear on campus more frequently at midday (about 2 hours later than a weekday though), and disappear as 4:00 AM approaches. A notable difference is the quantity of students: there are about 1/7the number of students on campus at peak hours.

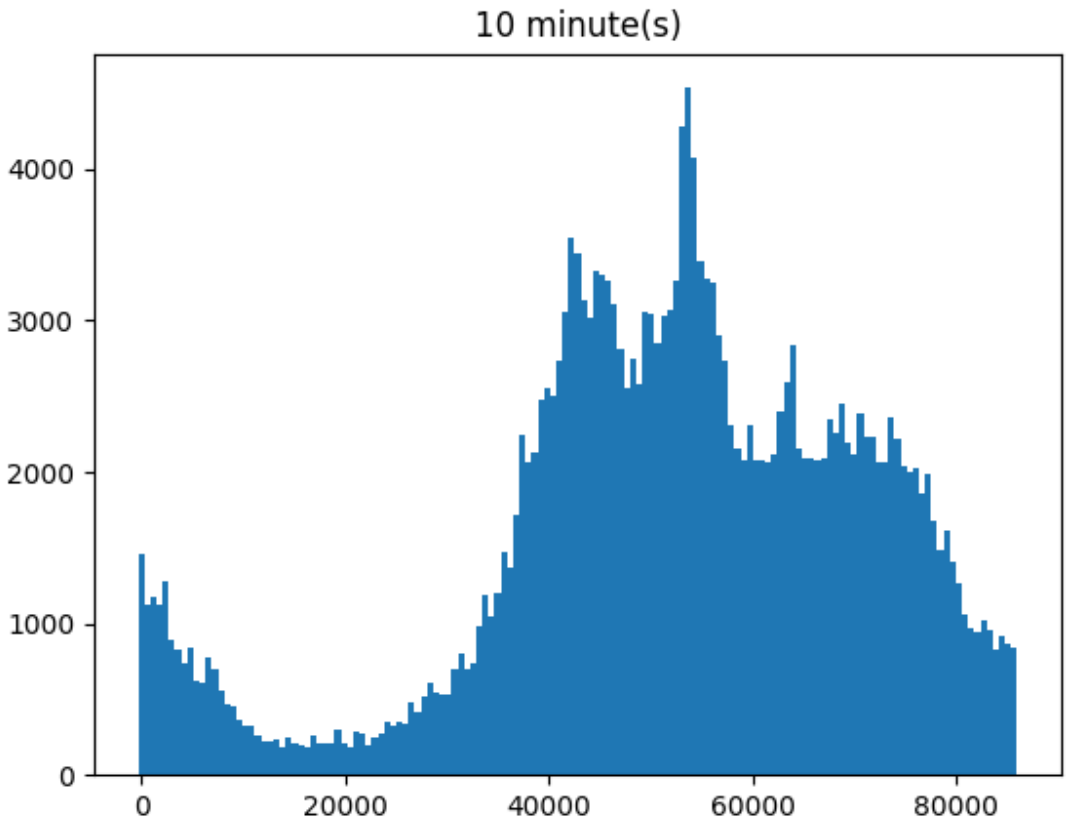
1 minute graph:



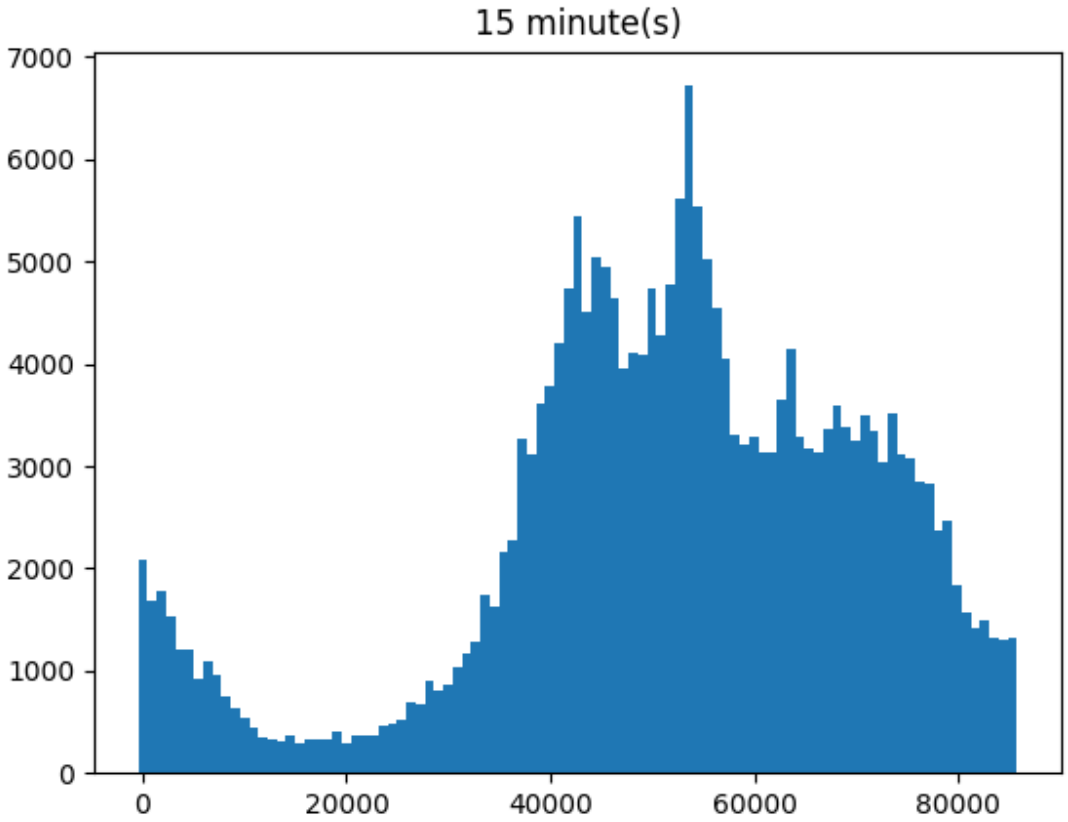
5 minute graph:



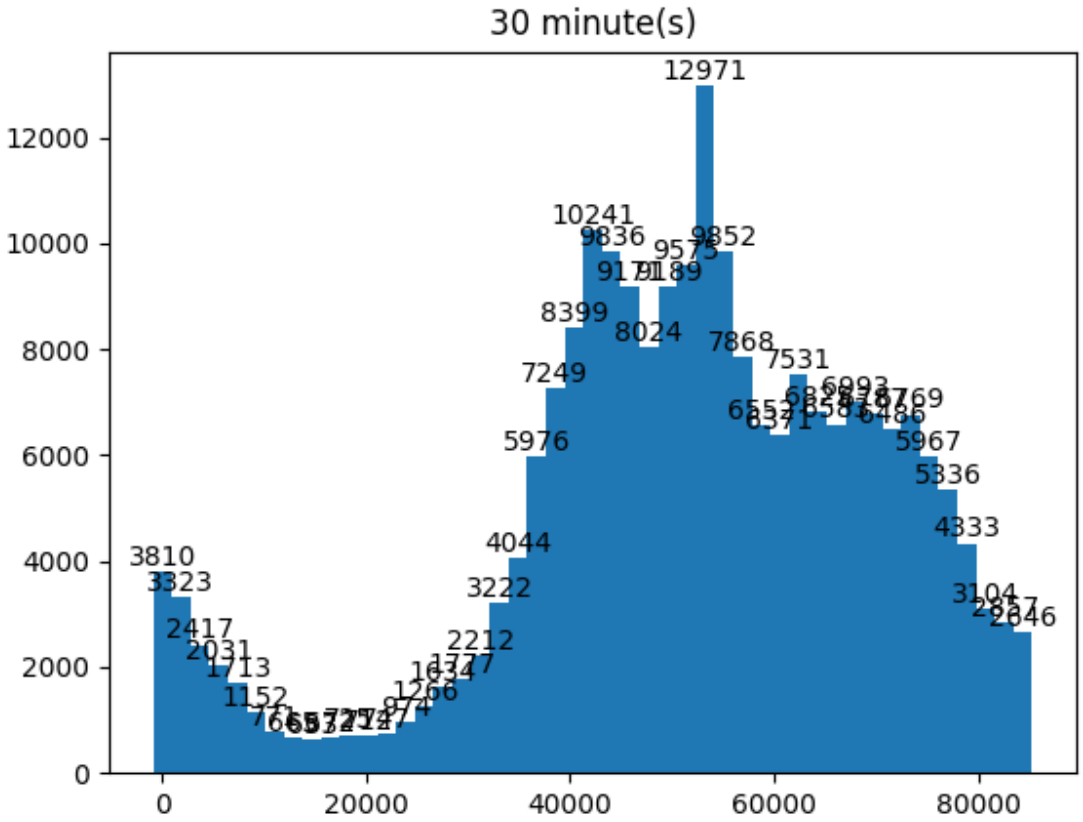
10 minute graph:



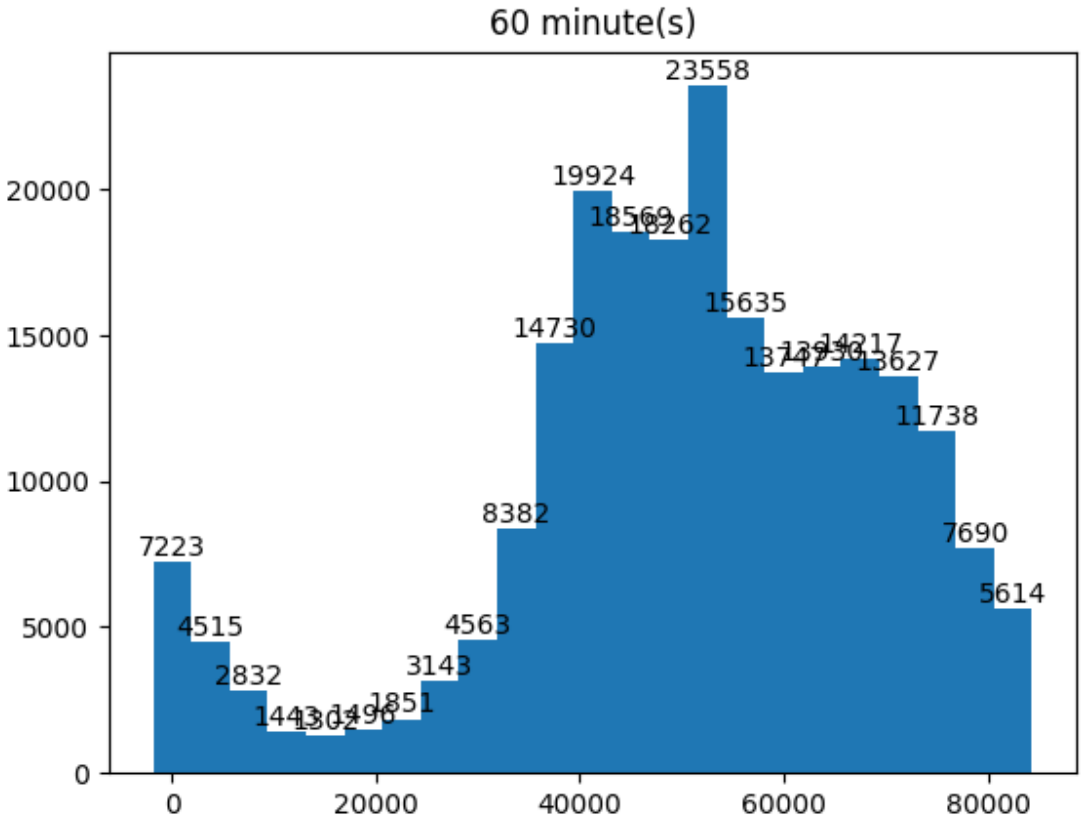
15 minute graph:



30 minute graph:



60 minute graph:

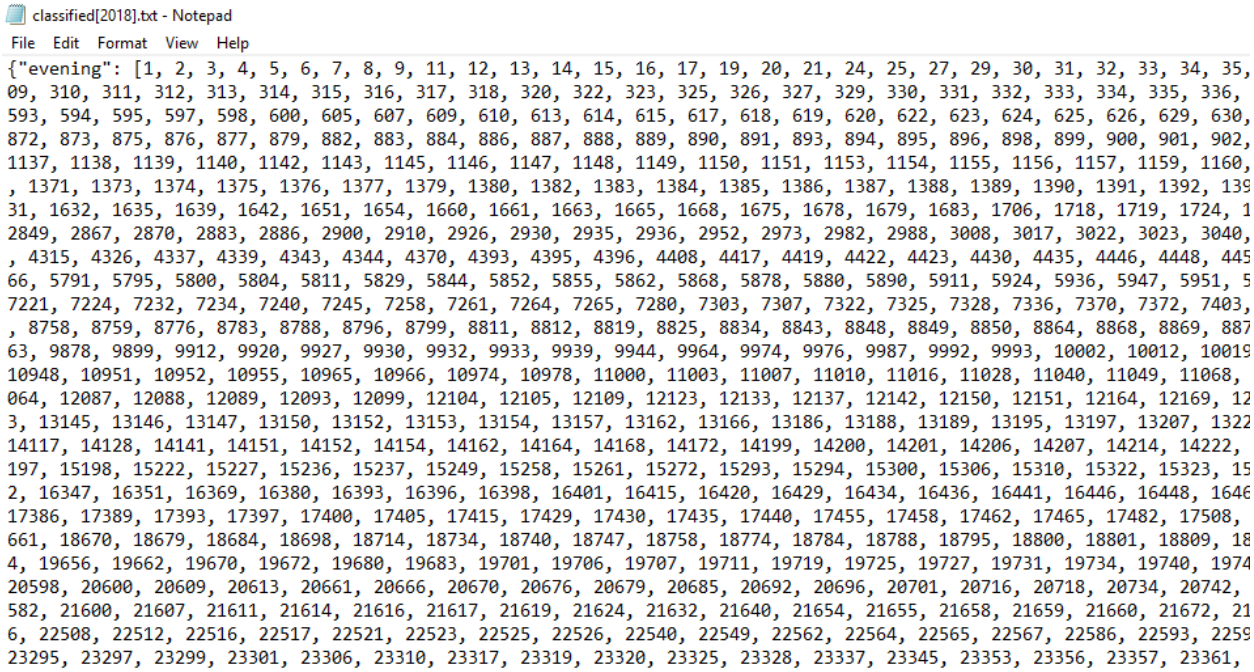


Part III.

Using "outputwireless-logs-20120409.DHCP_ANON"

• Can you identify the user devices that have many sessions in early morning hours (the early-birds)? How about during lunch time (the munchers)? Or the evening (the stompers)?

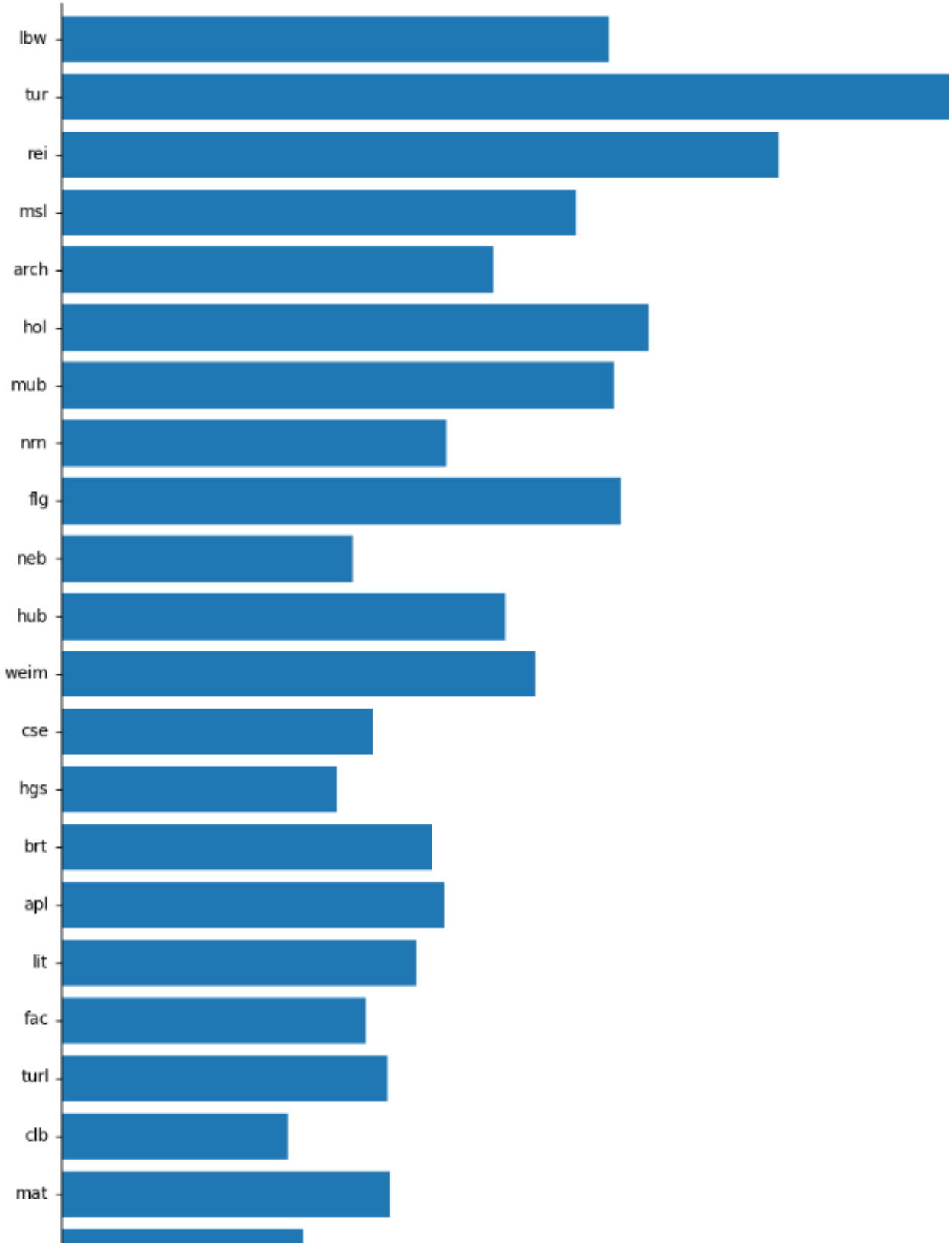
Yes, by looking at the frequency of time intervals of user devices, each individual can be labeled as an early-bird, muncher, or stomper based on how frequently they visit buildings on campus at each time interval. We were able to write a script that grouped all the userMACs into whether they mostly visited campus in the morning, lunch, or evening. The script finds all occurrences of a userMAC and tallies the number of hits for each time bin. The output of this script very large, however, and is not suited for being pasted into this document. A screenshot showing some of this example, where all the userMACs who appear predominantly in the evening, is shown here:



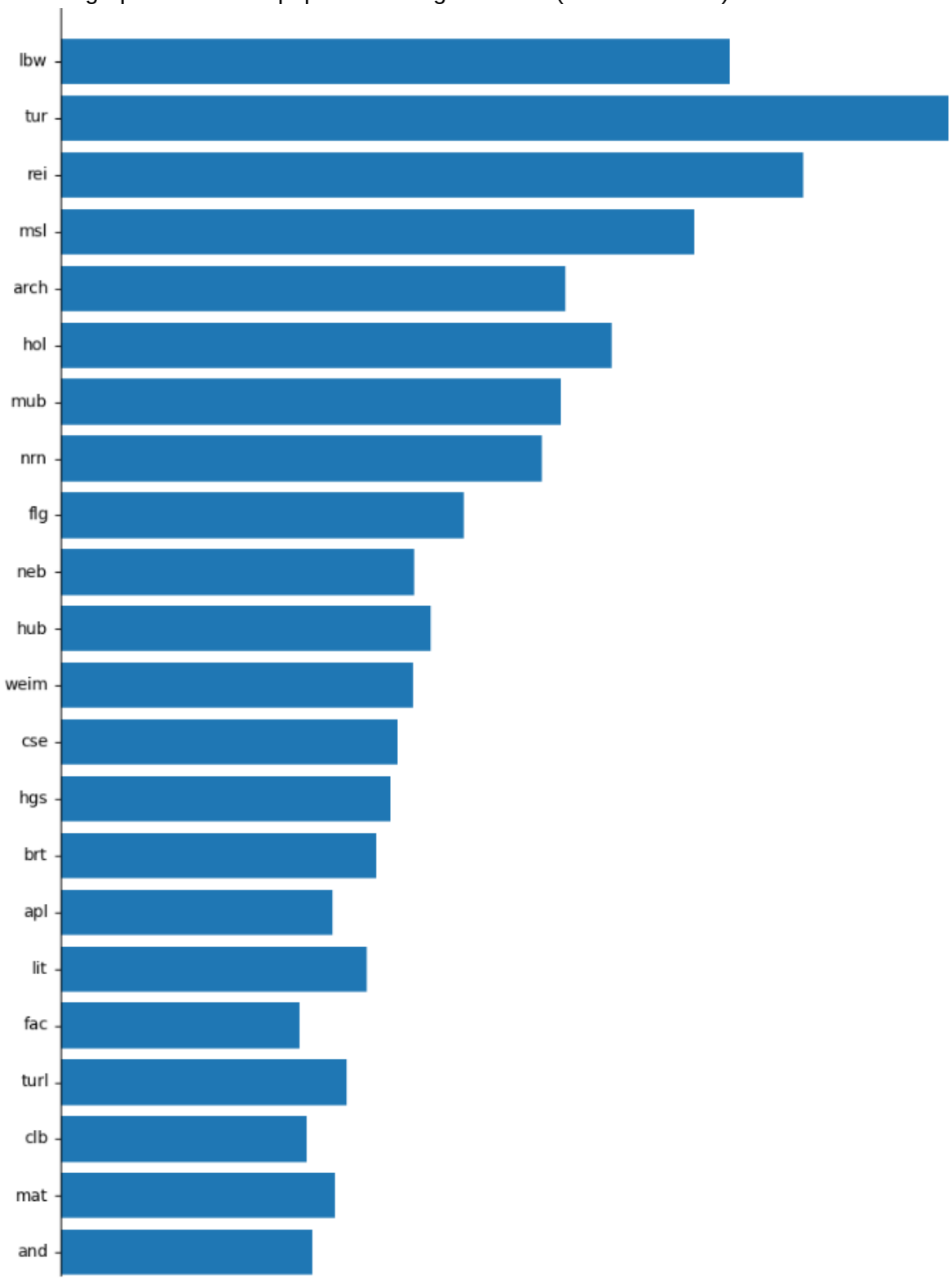
• *Use Input.b to map these devices (and corresponding sessions) to locations. Which buildings are the most popular among 'early-birds'? Or 'munchers? Or 'stompers'?*
Among "early birds" (defined as people with the most occurrences between 5 AM to 11 AM), the most popular buildings were Turlington (tur), Reitz Union (rei), and the Holland Law Center (hol). Turlington would have many people in the early morning due to morning classes and people getting off the bus. Reitz Union would also have people arriving from the bus, as well as restaurant employees opening the restaurant.
Among "munchers" (defined as people with the most occurrences between 11 AM to 4 PM), the most popular buildings were Turlington (tur), Reitz Union (rei), and Library West (lbw). Turlington is obviously very popular at this time due to many people having classes in the middle of the day, and walking between popular areas. Reitz Union is also popular at this time due to the Union's bus circle, multiple restaurants for eating lunch and socializing, and sitting areas for group projects and clubs.
Among "stompers" (defined as people with the most occurrences between 4 PM and afterwards), the most popular buildings were Library West (lbw), Reitz Union (rei), and Marston Science Library (msl). This makes sense, as many people would be studying late in these three areas--Marston and Reitz stay open until 2 AM, and Library West is open 24 hours which is why it dominates the category by a large margin.

• *Does the most popular building of different users change as the day progresses? What are the most popular building categories in the morning? How about in the evenings?*
Yes. In the morning, the most popular building is Turlington. By lunchtime, the most popular building stays being Turlington but with the Reitz Union behind by a small margin. In the evening, the most popular building changes to Library West by a large margin. This is most likely due to the lack of classes in the evening and more students arrive at Library West to study.

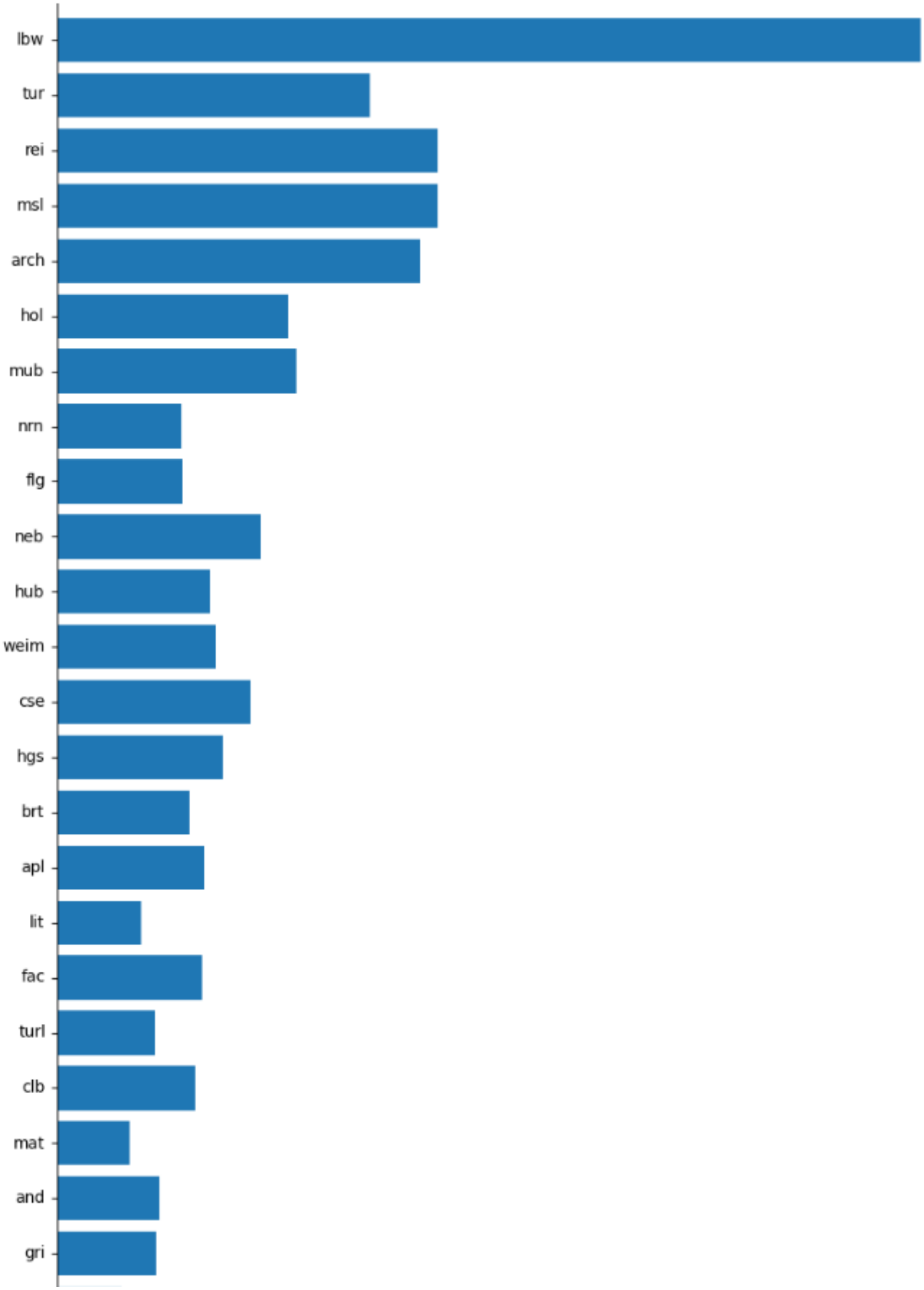
Top 20% of graph of the most popular buildings in the morning (5 AM to 11 AM):



Top 20% of graph of the most popular buildings at lunch (11 AM to 4 PM):



Top 20% of graph of most popular buildings in the evening (4 PM and afterwards):



Part IV.
We could not complete the fieldwork for Part IV due to the coronavirus.

Code for Part I-II:

```
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_csv(
    r'outputwireless-logs-20120409.DHCP_ANON.csv')['endTime'].subtract(1333929823)
fig, axs = plt.subplots(ncols=3, nrows=2)
timeFrames = [15, 1, 5, 10, 30, 60]
for time in timeFrames:
        bins = 86606 / (time * 60)
    fig.add_subplot(pd.DataFrame.hist(df, bins=bins))
plt.show()
#
# fig.add_subplot(df.hist(bins=97))  # 15min
# fig.add_subplot(df.hist(bins=1444))  # 1min
# fig.add_subplot(df.hist(bins=289))  # 5min
# fig.add_subplot(df.hist(bins=144))  # 10min
# fig.add_subplot(df.hist(bins=49))  # 30min
# fig.add_subplot(df.hist(bins=24))  # 60min
# fig.tight_layout()

# plt.show()

df.plot.hist(bins=97)
plt.show()
```

Alt Code for Part I-II:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

df = pd.read_csv(
    r'outputwireless-logs-20120407.DHCP_ANON.csv')['endTime'].subtract(1333929823)

#fig, ax = plt.subplots(ncols=6, nrows=1)
timeFrames = [15, 1, 5, 10, 30, 60]
for index, time in enumerate(timeFrames):
    binAmt = 86238 // (time * 60)
    heights, bins = np.histogram(df, bins=binAmt)
    fig = plt.figure()
    plt.bar(bins[:-1], heights, width=bins[1] - bins[0])
    ax = plt.gca()
    ax.set_title(str(time) + ' minute(s)')
    if time >= 30:
        rects = ax.patches
        for rect in rects:
            height = rect.get_height()
            ax.text(rect.get_x() + rect.get_width() / 2, height + 5, height,
                    ha='center', va='bottom')
    plt.savefig('plt' + str(index))
```

Code for Part III:

```
import pandas as pd
from collections import defaultdict, Counter
import matplotlib.pyplot as plt
import numpy as np
from datetime import datetime
import re
import json

prefix_df = pd.read_csv(r'prefix_lat_lon_name_category.csv')
df = pd.read_csv(r'outputwireless-logs-20120409.DHCP_ANON.csv')

mapping = dict()
for index, row in prefix_df.iterrows():
    mapping[row['prefix']] = row['name']

peopleLocs = defaultdict(Counter)
people = defaultdict(Counter)
building = defaultdict(Counter)
building_tot = Counter()
for index, row in df.iterrows():
    locargs = row['APNAME'].split('-')
    locap = locargs[0]
    knownMatch = re.match(r'([A-Za-z]+)(?:\w+)*$', locap)
    location = locap
    if knownMatch:
        prefix = knownMatch.group(1)
        if prefix in mapping:
            location = prefix
    dt = datetime.utcfromtimestamp(row['startTime'])

    # morning (5am <= m < 11am)
    # lunch (11am <= l < 4pm)
    day = 'evening'
    if 5 <= dt.hour < 11:
        day = 'morning'
    elif 11 <= dt.hour < 16:
        day = 'lunch'

    uid = row['userMAC']
    peopleLocs[uid][location] += 1
    people[uid][day] += 1
    building[day][location] += 1
    building_tot[location] += 1
```

```
# classify into bins (morning, lunch, evening)
types = defaultdict(list)
for uid in people:
    data = people[uid]
    typeOfPerson = (None, 0)
    for day in data:
        if data[day] > typeOfPerson[1]:
            typeOfPerson = (day, data[day])
    types[typeOfPerson[0]].append(uid)
with open('classified.txt', 'w') as outfile:
    json.dump(types, outfile)
# which buildings are popular certain times of the day
plt.figure(figsize=(50, 10))
buildings = [loc for loc, val in sorted(
    mapping.items(), key=lambda item: item[0] in building_tot and building_tot[item[0]] or 0,
reverse=True)]
index = 0
width = 0.25
for day in building:
    y = []
    for prefix in buildings:
        y.append(building[day][prefix])
    ind = np.arange(len(buildings))
    plt.bar(ind + width * index, y, width=width, label=day)
    plt.ylabel('Frequency')
    plt.xlabel('Buildings', rotation=90)
    plt.yticks(rotation=90, va="center", ha="right")
    plt.xticks(ind + width, buildings, rotation=90, va="top", ha="center")
    plt.legend()
    index += 1
plt.tight_layout()
plt.savefig("buildingpopular")
# which buildings are more popular are among the types of people
for t in types:
    plt.figure(figsize=(100, 10))
    people = types[t]
    yp = Counter()
    for uid in people:
        for prefix in buildings:
            yp[prefix] += peopleLocs[uid][prefix]
    y = yp.values()
    plt.bar(buildings, y)
    plt.ylabel('Frequency')
    plt.xlabel('Buildings', rotation=90)
```

```
plt.xticks(rotation=90)
plt.tight_layout()
plt.savefig(t)
```