MAT259 - Assignment 1

Dongyu Meng
Jan 15 - 2020

In this assignment I tried to find out if there is a correlation between the theme of books people borrow and the time in the day these books get borrowed. Like, is it true that people tend to borrow technical books in the morning and novels at night? Inquiries like this give insight to the activity patterns of different readers or even professions.

The basic SQL query is the following:

```
1 select deweyClass, count(*), hour(cout) from spl_2016.outraw
2 where cout Between '2015-01-01' AND '2016-01-01'
3 and deweyClass != ""
4 and itemtype = "acbk"
5 group by deweyClass, hour(cout)
```

In the records in the year of 2015, I counted the number of books borrowed group by the book's Dewey class and the time in the day (discretized to hours) the book got borrowed. This query gave me data like the following:

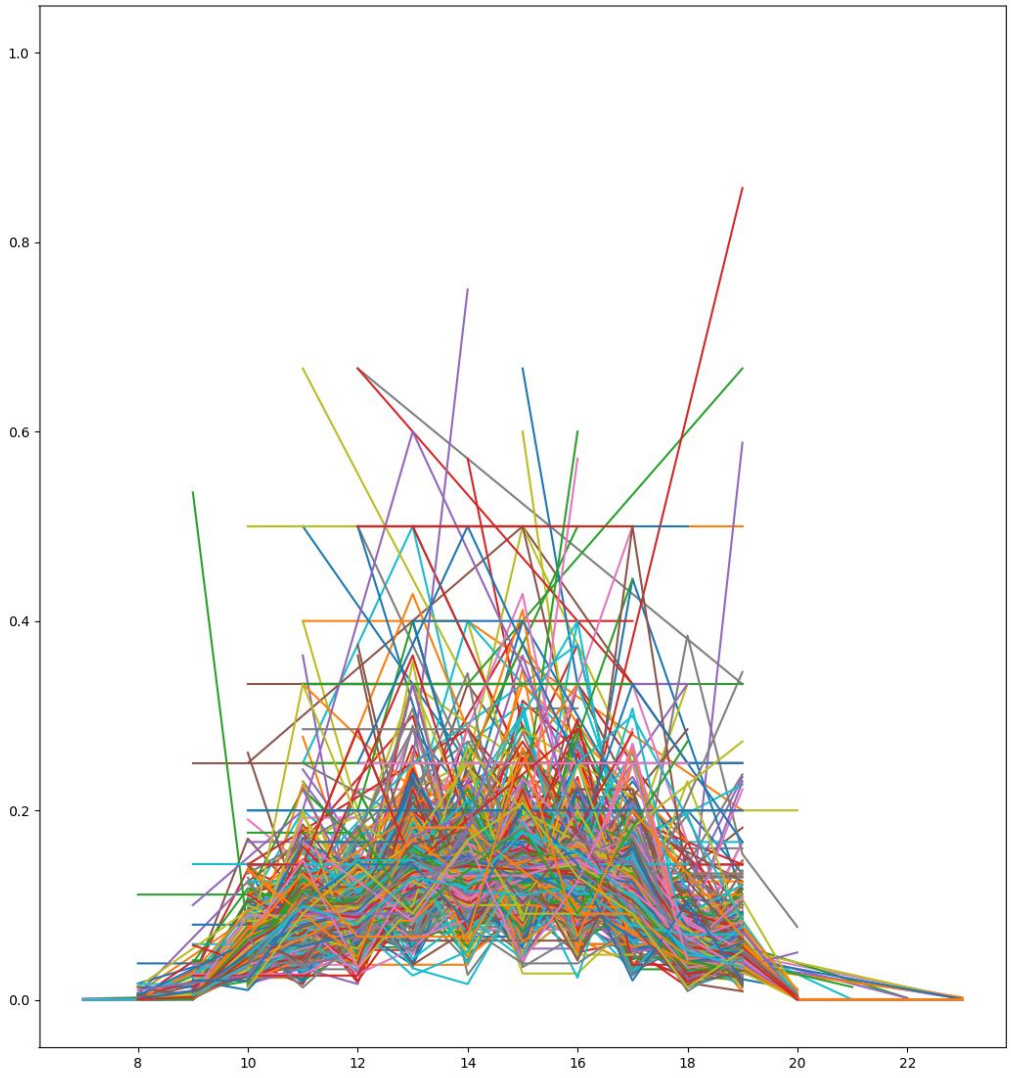
	deweyClass	count(*)	hour(cout)
1	001	5	10
2	001	9	11
3	001	9	12
4	001	15	13
5	001	17	14
6	001	10	15
7	001	11	16
8	001	11	17
9	001	7	18
10	001	4	19
11	001.01	1	12
12	001.01	2	13

I further processed the data with Python.

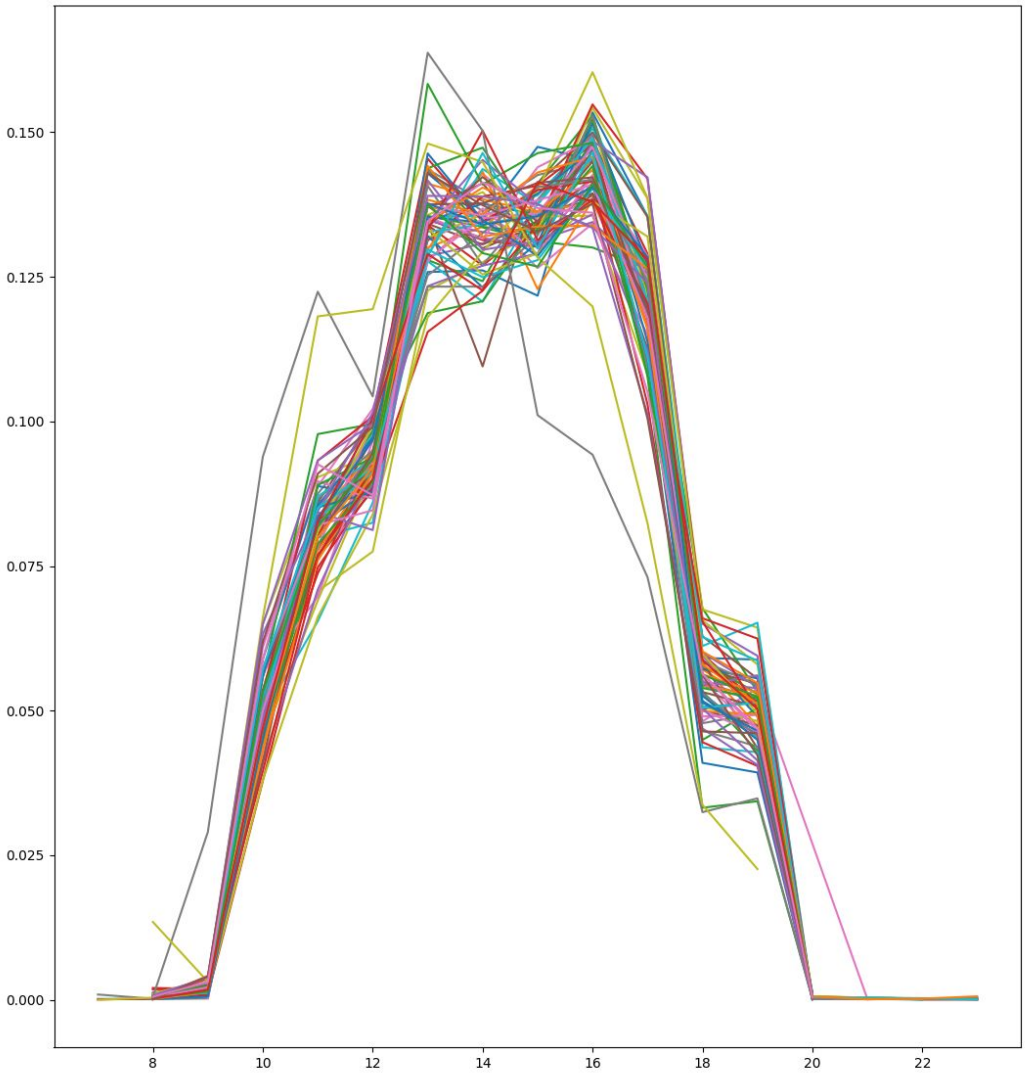
First, I unified the Dewey class number as shown above to the major class number only (001.01 -> 001) and aggregated the count aligning to the (deweyClass, hour) tuple.

Then, I normalized the per hour count to per hour percentage for each Dewey class by dividing the per hour count with the total number of books borrowed under that category.

If we plot the numbers now, with one line for each Dewey class, we get the following graph (x axis being time in a day (hour), y axis being percentage):



Quite messy, huh? We can see some radical outliers too. To make the graph more readable and more smooth, we now consider only classes with more than 3650 cout transactions (more than 10 times per day in the year). We get the following graph:

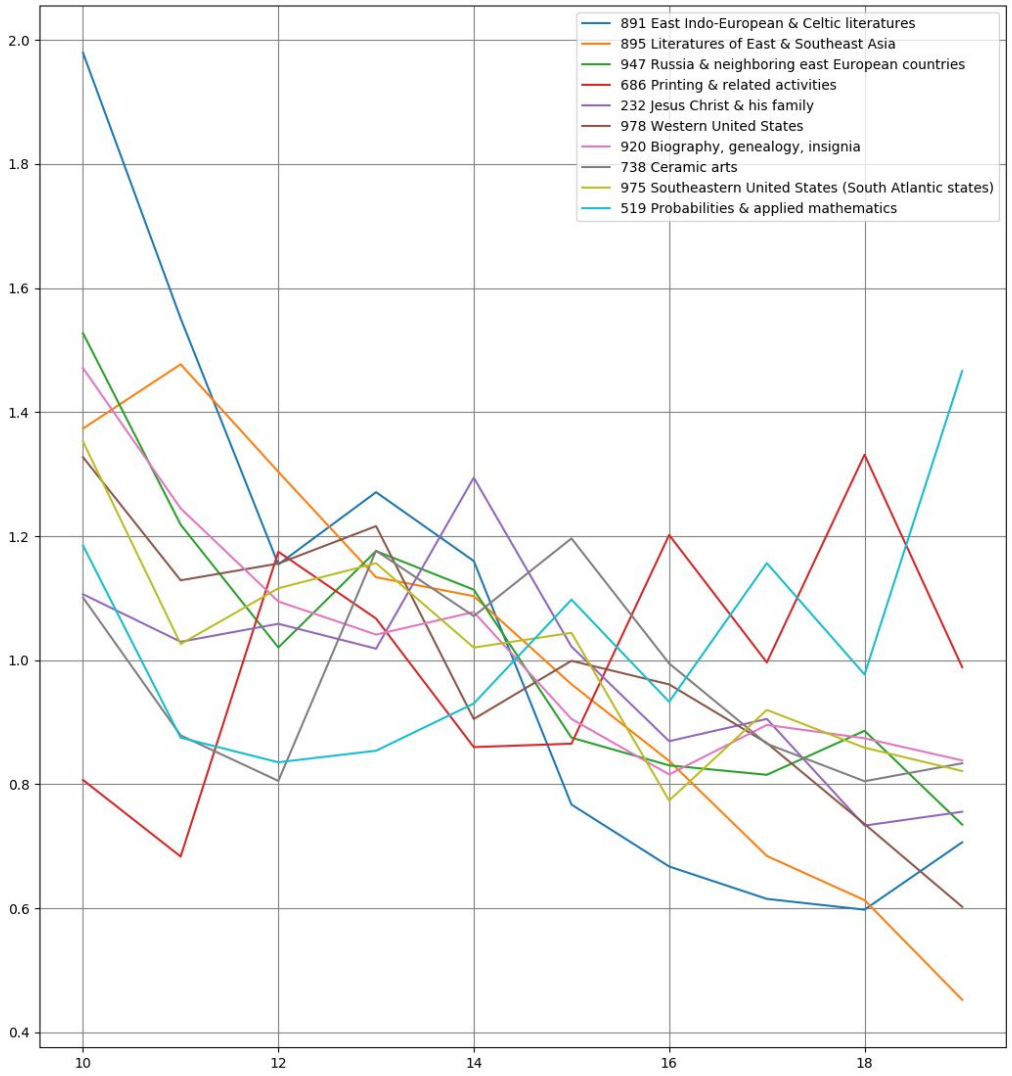


Much more readable, and we conclude here that for the majority of book classes, people borrow them in the afternoon (12pm - 5pm) during normal working hours. Wait…? Shouldn't they be working?

Anyways…

So who are those that tend to have a different activity pattern?

To identify them, I calculated the distribution of the book outlets (all classes) over time in the day as the baseline, and compare each class's distribution against the baseline. The top ten outliers are shown in the graph below. Note that the y axis is normalized against the baseline.



From the graph we see that most of the "weird" patterns fall into humanity related fields. Moreover, these humanity books have a much higher chance to be borrowed earlier in the day, while the only two math/engineering related classes (519 and 686) in the top ten have a higher chance to be borrowed later in the day.

Appendix / Code:

```
import mysql.connector
import pandas
import ipdb
import sshtunnel
import matplotlib.pyplot as plt

_host = 'tango.mat.ucsb.edu'
_ssh_port = 22
_username = 'mat259'
_password = ******
_remote_bind_address = _host
_remote_mysql_port = 3306
_local_bind_address = '127.0.0.1'
_local_mysql_port = 10087
_db_user = 'mat259'
_db_password = *******
_db_name = 'spl_2016'

dewey_mapping = {}
with open("mapping", "r") as handle:
    for line in handle:
        line = line.strip()
        number = line.split(" ")[0]
        dewey_mapping[number] = line

query = """
select deweyClass, count(*), hour(cout) from spl_2016.outraw
where cout Between '2015-01-01' AND '2016-01-01'
and deweyClass != ""
and itemtype = "acbk"
and hour(cout) < 20
and hour(cout) > 9
group by deweyClass, hour(cout)
"""

divs = []

with sshtunnel.SSHTunnelForwarder(
    (_host, _ssh_port),
    ssh_username=_username,
```

```
    ssh_password=_password,
    remote_bind_address=(_remote_bind_address, _remote_mysql_port),
    local_bind_address=(_local_bind_address, _local_mysql_port)
) as tunnel:
    conn = mysql.connector.connect(
        user=_db_user,
        password=_db_password,
        host=_local_bind_address,
        database=_db_name,
        port=_local_mysql_port)

    cursor = conn.cursor()
    cursor.execute(query)
    result = cursor.fetchall()

    tb = pandas.DataFrame(result, columns=["dewey", "num", "hour"])
    tb["dewey"] = tb["dewey"].apply(lambda x: x[:3])

    dewey_sum = tb.groupby(["dewey"])["num"].sum()
    dewey_hour_sum = tb.groupby(["dewey", "hour"])["num"].sum().astype("float")
    hour_sum = tb.groupby(["hour"])["num"].sum().astype("float")
    hour_sum /= hour_sum.sum()

    ipdb.set_trace()

    for dewey, total_num in dewey_sum.items():
        total = dewey_sum[dewey]
        dewey_hour_sum[dewey] /= total

        div = 0
        for h, p in hour_sum.items():
            if h >= 20 or h < 10: continue
            if h in dewey_hour_sum[dewey]:
                div += (p-dewey_hour_sum[dewey][h])**2
                dewey_hour_sum[dewey][h] /= p
            else:
                div += p**2
        # print(div)
        if total > 1000:
            divs.append((div, dewey))

    divs.sort(reverse=True)
    for i in range(10):
```

```
    _, dewey = divs[i]
    plt.plot(dewey_hour_sum[dewey], label=dewey_mapping[dewey])

plt.grid(color='grey', linestyle='-')
plt.legend()
plt.show()
```