

소쿠리 데이터 분석대회

서광류탈 조

201002383 김동수
201000869 최준우
201002488 장한별

환경 구축

Json 데이터를 DB로 구축하기 위하여 Mongo DB를 사용

The screenshot shows the MongoDB Shell interface. At the top, there are three tabs: `db.getCollection('sokuri')`, `db.getCollection('sokuri')`, and `* db.getCollection('sokuri')`. Below the tabs, the connection information is displayed: `New Connection`, `localhost:27017`, and `local`. The command entered in the shell is `db.getCollection('sokuri').find({})`. The execution time is shown as `0.012 sec.`. The results are displayed in a table with two columns: **Key** and **Value**. The table contains 12 rows of data, each representing a document in the `sokuri` collection. Each document has a key (e.g., `(1) changed_1_20151110_sleep.json`) and a value (`{ 3 fields }`).

Key	Value
(1) changed_1_20151110_sleep.json	{ 3 fields }
(2) changed_17_20151109_heart.json	{ 3 fields }
(3) changed_17_20151110_heart.json	{ 3 fields }
(4) changed_3_20151110_heart.json	{ 3 fields }
(5) changed_4_20151107_sleep.json	{ 3 fields }
(6) changed_4_20151108_sleep.json	{ 3 fields }
(7) changed_4_20151109_sleep.json	{ 3 fields }
(8) changed_6_20151104_sleep.json	{ 3 fields }
(9) changed_6_20151107_sleep.json	{ 3 fields }
(10) changed_6_20151108_sleep.json	{ 3 fields }
(11) changed_6_20151110_sleep.json	{ 3 fields }
(12) changed_7_20151109_sleep.json	{ 3 fields }

환경 구축

배치 파일을 작성하여 Json 파일을 MongoDB로 Import



data_imp

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

cd C:\data\bin

```
mongoimport -d local -c sokuri --file c:\Users\choijunwoo\Desktop\소쿠리\changed_1_20151101_heart.json
mongoimport -d local -c sokuri --file c:\Users\choijunwoo\Desktop\소쿠리\changed_1_20151101_sleep.json
mongoimport -d local -c sokuri --file c:\Users\choijunwoo\Desktop\소쿠리\changed_1_20151101_step.json
mongoimport -d local -c sokuri --file c:\Users\choijunwoo\Desktop\소쿠리\changed_1_20151102_heart.json
mongoimport -d local -c sokuri --file c:\Users\choijunwoo\Desktop\소쿠리\changed_1_20151102_sleep.json
mongoimport -d local -c sokuri --file c:\Users\choijunwoo\Desktop\소쿠리\changed_1_20151102_step.json
mongoimport -d local -c sokuri --file c:\Users\choijunwoo\Desktop\소쿠리\changed_1_20151103_heart.json
mongoimport -d local -c sokuri --file c:\Users\choijunwoo\Desktop\소쿠리\changed_1_20151103_sleep.json
mongoimport -d local -c sokuri --file c:\Users\choijunwoo\Desktop\소쿠리\changed_1_20151103_step.json
mongoimport -d local -c sokuri --file c:\Users\choijunwoo\Desktop\소쿠리\changed_1_20151104_heart.json
mongoimport -d local -c sokuri --file c:\Users\choijunwoo\Desktop\소쿠리\changed_1_20151104_sleep.json
mongoimport -d local -c sokuri --file c:\Users\choijunwoo\Desktop\소쿠리\changed_1_20151104_step.json
mongoimport -d local -c sokuri --file c:\Users\choijunwoo\Desktop\소쿠리\changed_1_20151105_heart.json
mongoimport -d local -c sokuri --file c:\Users\choijunwoo\Desktop\소쿠리\changed_1_20151105_sleep.json
mongoimport -d local -c sokuri --file c:\Users\choijunwoo\Desktop\소쿠리\changed_1_20151105_step.json
mongoimport -d local -c sokuri --file c:\Users\choijunwoo\Desktop\소쿠리\changed_1_20151106_heart.json
```

환경 구축

데이터 분석에 용이한 Ipython과 Python 라이브러리 이용

11월12일에 최종수정 된 Data Set 사용

문제 1

**참가자 중 오전에 가장 많이 운동을 한 랭킹 TOP와
오후에 운동을 많이 한 랭킹 TOP5를 구하시오.**

문제 1

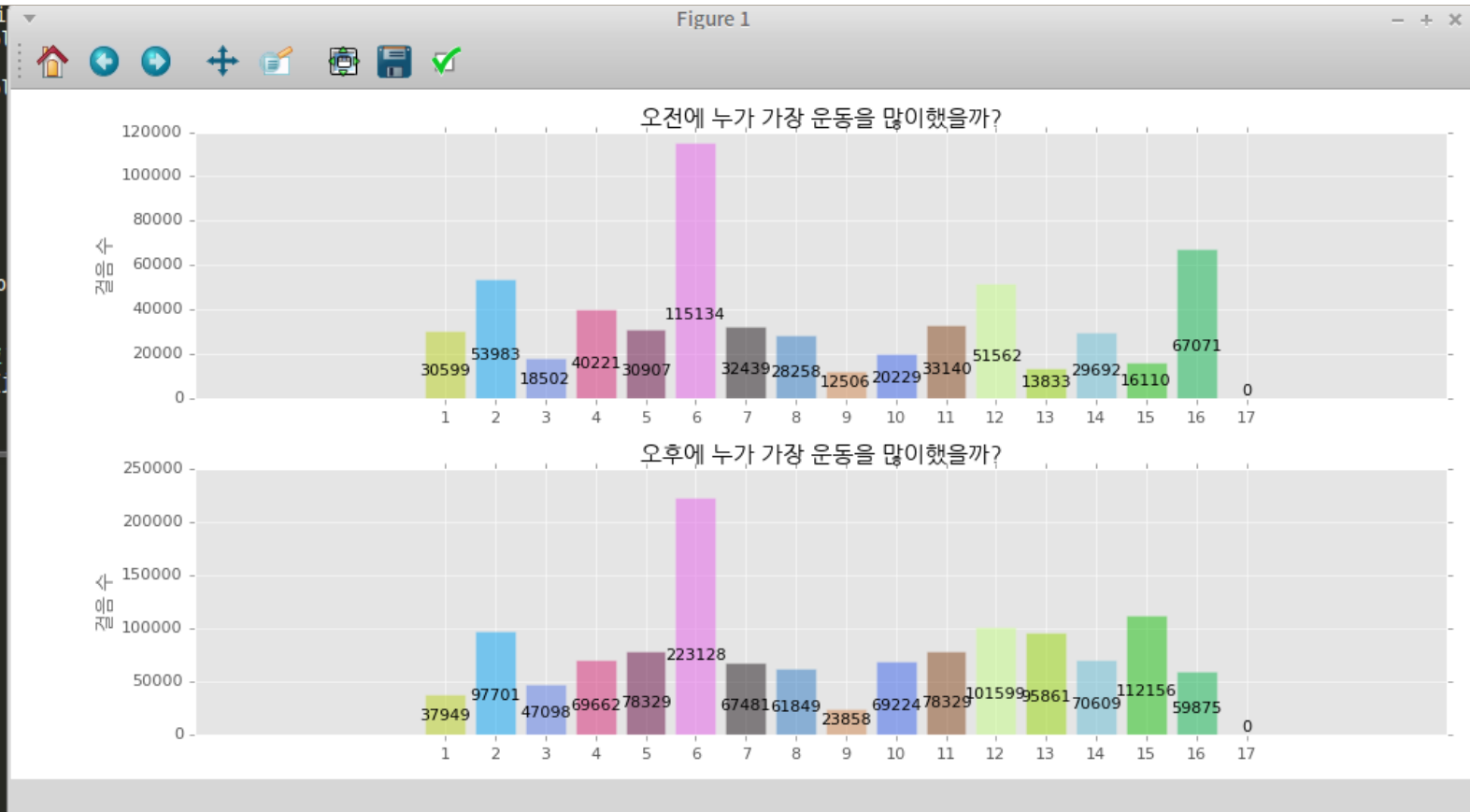
1. 모든 참가자의 STEP DATA LOAD
2. STEP DATA 의 “time” 오브젝트로 오전, 오후 분류
3. 참가자별 분류된 DATA들의 STEP VALUE COUNT
4. COUNT 값으로 랭킹 출력 , 데이터 시각화

문제 1

```
5 from operator import itemgetter
6 import matplotlib.pyplot as plt
7 import numpy as np
8 import matplotlib.pyplot as plt
9 import matplotlib
10 import random
11
12 def readStep(user):
13
14     step_list = []
15
16     file_list = glob.glob('data/*')
17     for f in file_list:
18         file = open(f)
19         js = json.loads(file.read())
20         step_list.append(js['step'])
21         file.close()
22     return step_list
```

오전에 많이 운동한 사람 랭킹 TOP 5
User Number : 6 115134
User Number : 16 67071
User Number : 2 53983
User Number : 12 51562
User Number : 4 40221

오후에 많이 운동한 사람 랭킹 TOP 5
User Number : 6 223128
User Number : 15 112156
User Number : 12 101599
User Number : 2 97701
User Number : 13 95861



정답 : 오전 랭킹 6,16,2,12,4

오후 랭킹 6,15,12,2,13

문제 2

참가자 중 가장 꿀잠을 잔 참가자를 찾아라!

(권장 수면 시간은 [8시간] 권장 기상 시간은 일출 시간[오전 7시]이다.)

참가자 중 오후 11시부터 오전 7시까지의 수면 시간만을

꿀잠이라고 정의 했을 때 일주일 간 가장 꿀잠을 잔 참가자를 찾아주세요.

문제 2

1. 모든 참가자들의 SLEEP DATA LOAD
2. 참가자별 “minuteData” 값 파싱
3. 23시부터 7시 까지의 SLEEP VALUE COUNT 값을
 꿀잠지수로 지정
4. 꿀잠지수 랭킹 출력 및 데이터 시각화

문제 2

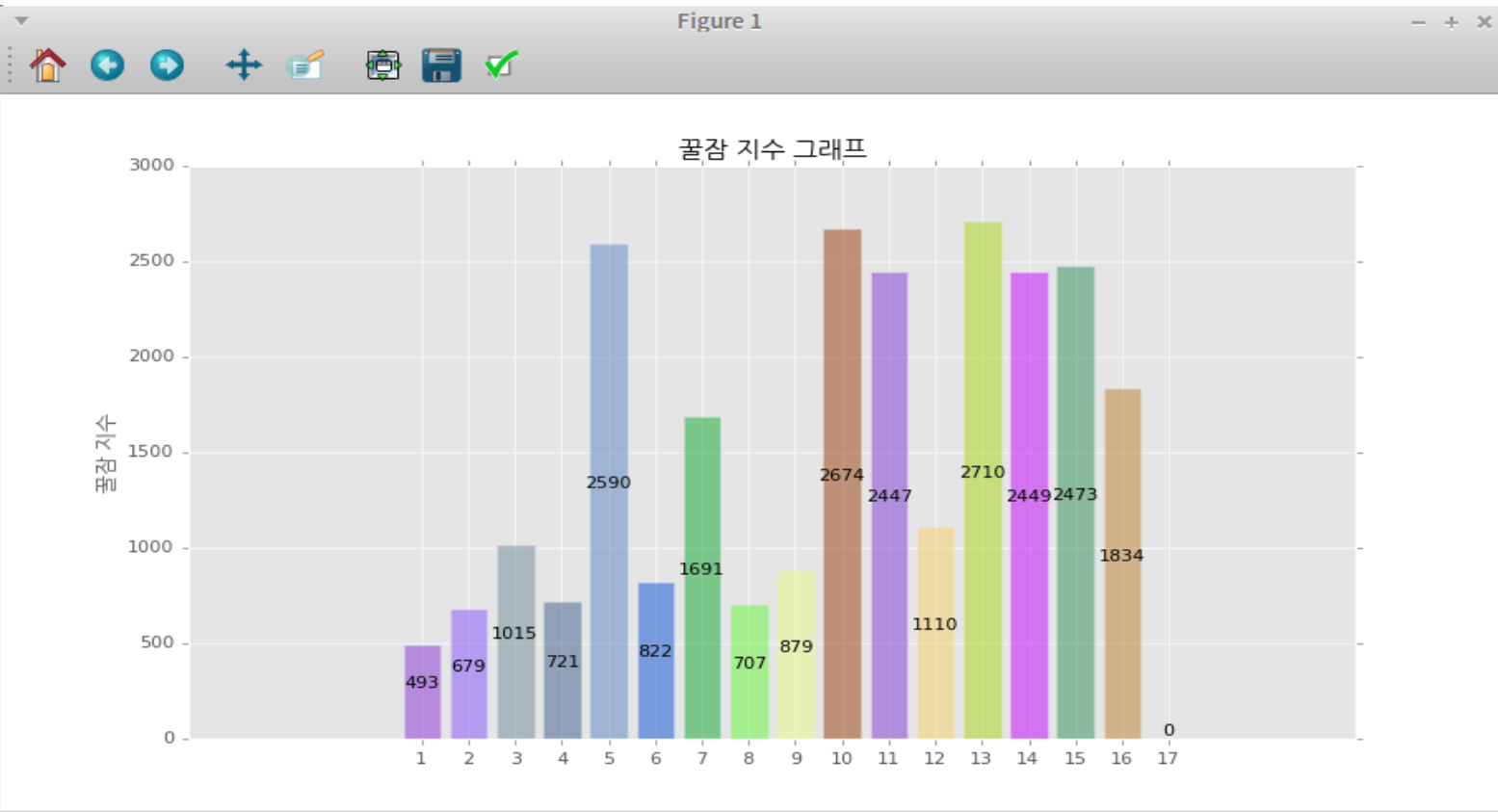
```
67 print("₩가장 꿀잠을  
68 print(str(final_list))  
69 matplotlib.style.use('seaborn')  
70  
71  
72 y_pos = np.arange(len(users))  
73 colors = []  
74 for i in range(0, len(users)):  
75     colors.append("#%06X" % (i * 255))  
76  
77 bar = plt.bar(y_pos, final_list, color=colors)  
78 plt.xticks(y_pos, users)  
79 plt.ylabel('꿀잠 지수')  
80  
81 plt.title('꿀잠 지수 그래프')  
82  
83 def autolabel(rects):  
84     for rect in rects:  
85         height = rect.get_height()  
86         autolabel(rect, height)
```

11월 1일 부터 일주일간 꿀잠 랭킹

User Number : 13	2710
User Number : 10	2674
User Number : 5	2590
User Number : 15	2473
User Number : 14	2449
User Number : 11	2447

가장 꿀잠을 주무신 분

User Number : 13



가장 꿀잠을 주무신 분 : 13번 참가자

문제 4

참가자들을 아침형 인간과 저녁형 인간으로 나누고 그 이유를 설명해주세요.

또한 사람의 생활 패턴이 아침형 인간과 저녁형 인간으로 나뉘어지지 않을 수도 있습니다.

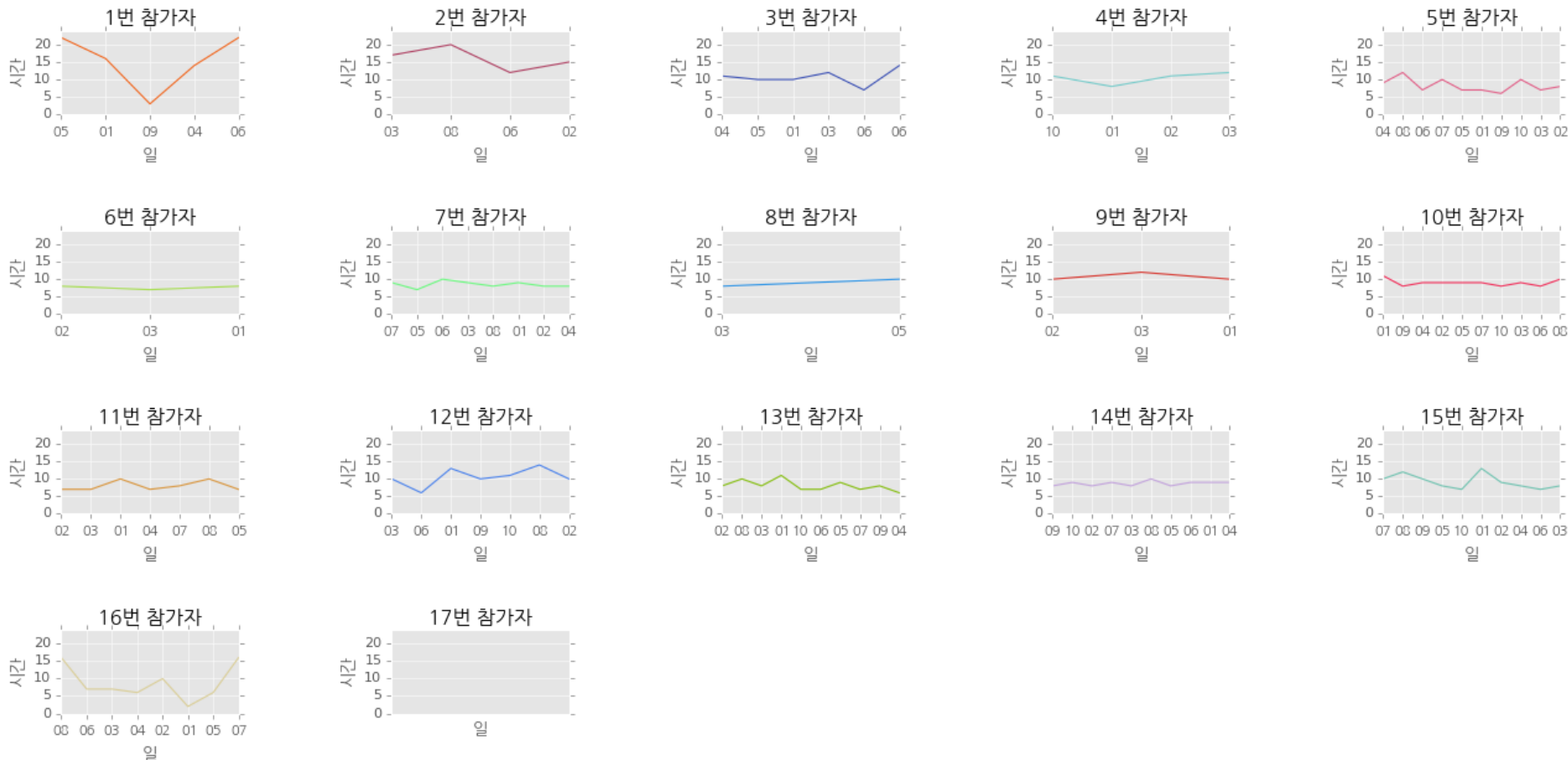
이러할 경우 이유를 설명해 주세요.

**(아침형 인간이란 아침에 기상하고 저녁에 잠을 자는 생활 패턴을 그리는 사람이라고 할 때
반대로 저녁형 인간이란 새벽에 잠이 들고 오후에 일어나서 생활을 하는 사람입니다.)**

문제 4

1. 모든 참가자들의 SLEEP DATA LOAD
2. 4시간 이상 잤을때를 낮잠,쪽잠을 제외한 주 수면 시간으로 설정
3. 사용자별 주 수면의 기상 시간 체크
4. 기상 시간 데이터 시각화

문제 4



그래프가 주로 아래에 위치한 참가자 : 아침형 인간 (3,4,5,6,7,8,9,10,11,13,14)

그래프가 주로 위에 위치한 참가자 : 저녁형 인간 (1,2)

그래프가 들쭉 날쭉한 참가자 : 분류하기 힘들 (16,12,15)

문제 4

수업을 듣는 학생이라 아침형 인간이 더 많았음.

FitBit을 착용하지 않고 잔 날이 많은 참가자가 많았음.

문제 5

컴퓨터 공학과 학생인 이현호와 그의 친구A는 체중감량을 위해 규칙적인 운동을 합니다.

이들은 같이 운동을 할 사람을 찾고 싶었지만,

슬프게도 둘 다 친구가 많지 않아 찾기가 쉽지 않았다고 합니다.

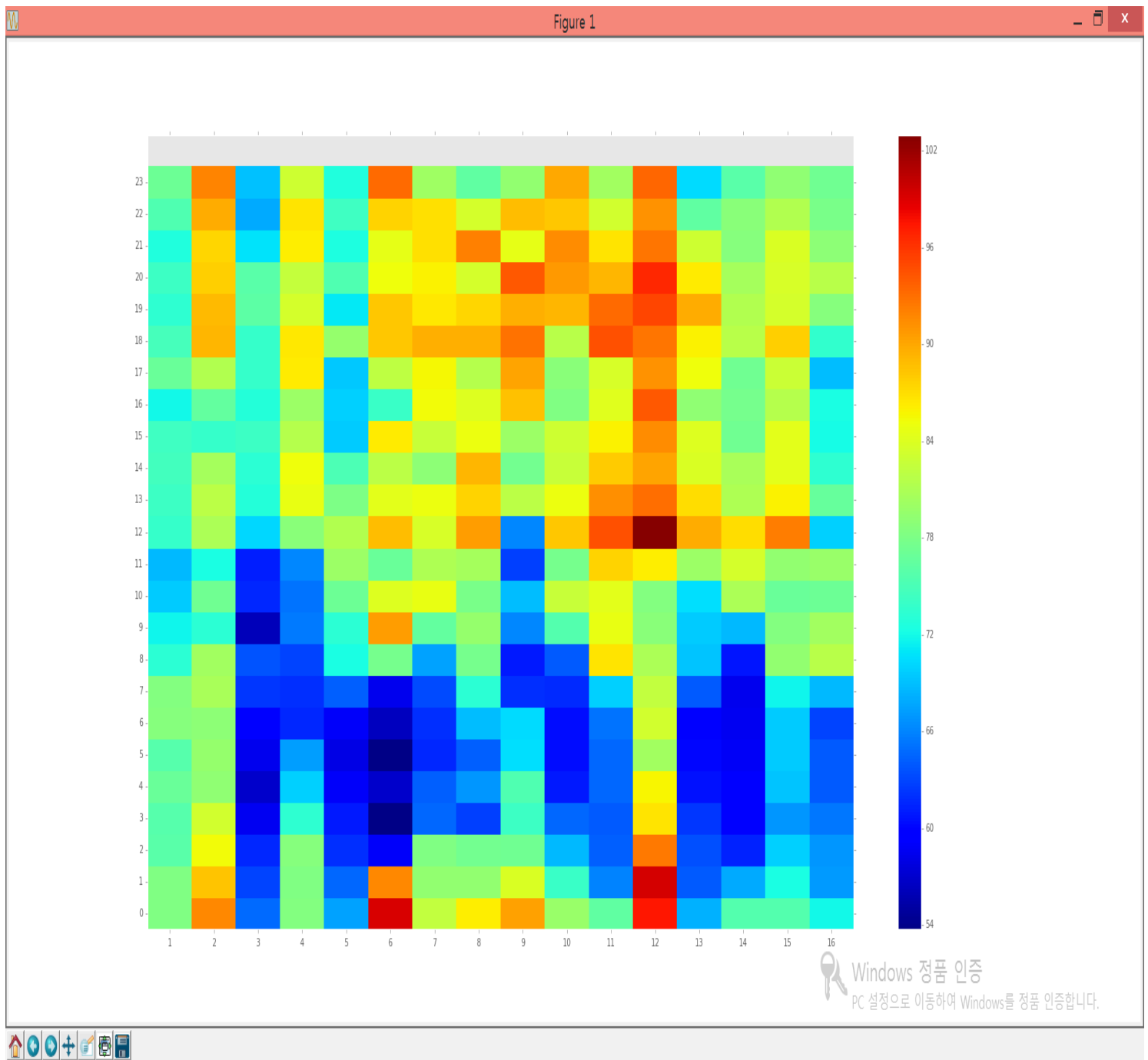
주어진 데이터를 분석하여 이현호의 친구A가 누구인지 찾아내고,

그들과 같이 운동을 할 수 있는 새 친구를 찾아보자!

문제 5

1. 모든 참가자들의 Heart DATA LOAD
2. 1시간 단위로 평균을 내어 User 별 심박수 Matrix 계산
3. 두 사용자 간 상관 계수를 이용하여 1번 사용자와 유사한 User 탐색
4. 사용자 별로 Heat Map 시각화

문제 5



문제 5

In [93]: `import numpy as np`

```
for i in range(1,16):  
    print(str(i) + " : " + str(np.corrcoef(heartBeat.iloc[0], heartBeat.iloc[i])[0][1]))
```

```
1 : 0.475421608299  
2 : -0.100011213816  
3 : 0.0796830573898  
4 : -0.626882261078  
5 : -0.224630094202  
6 : -0.400004475793  
7 : -0.330734825651  
8 : 0.215200094217  
9 : -0.360158635224  
10 : -0.594568709614  
11 : 0.20839815674  
12 : -0.433041933041  
13 : -0.501593110453  
14 : -0.41294676384  
15 : -0.642138024441
```

In [96]: `from sklearn.cluster import KMeans`

```
est=KMeans(7)  
#heartBeat=heartBeat_t.transpose()  
#print(heartBeat)  
est.fit(heartBeat)  
labels = est.labels_  
  
print(labels)
```

```
[5 5 2 0 2 4 6 1 0 6 1 3 1 2 1 2]
```

문제 5

도식화 한 Heat Map을 이용해 확인한 1번 사용자는 주로 저녁에 많이 생활하며 운동 또한 늦은 저녁에 하는 것을 확인 할 수 있었다. 이는 2번 사용자에서도 보이는 패턴으로 1번 사용자의 친구는 2번 사용자인 것을 알 수 있었다. 이러한 기법을 통해 많이 유사하지는 않지만 9번 사용자와 12번 사용자를 친구로 추천 할 수 있다고 판별 할 수 있었다.

문제 6

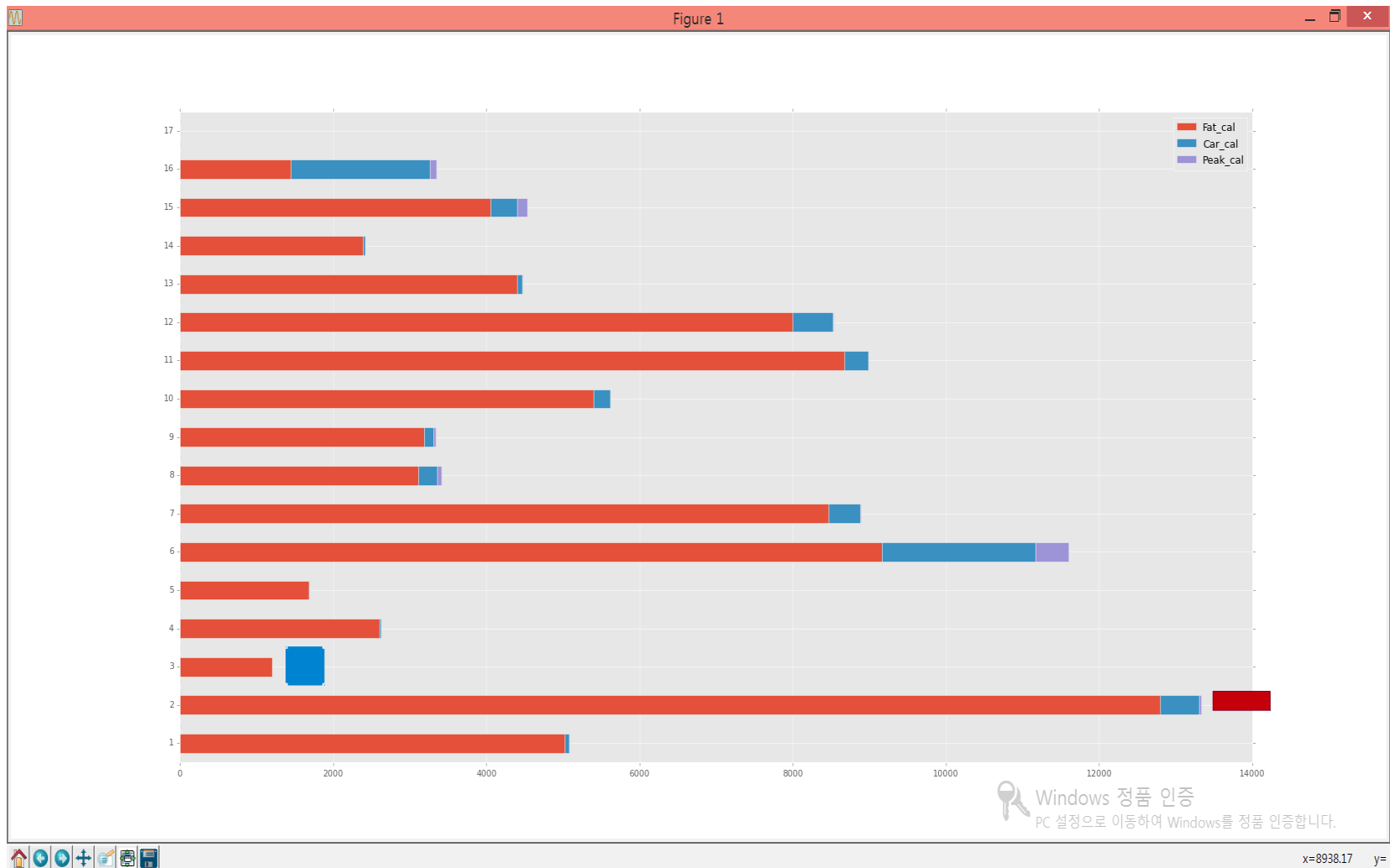
**참가자들의 생활 패턴을 보고 앞으로 체중이 가장 많이 증가 할 것 같은 참가자와 감소
할 것 같은 참가자를 예측해주세요.**

(참가자들의 종합적인 모든 데이터를 바탕으로 예측하시면 됩니다.)

문제 6

1. 모든 참가자들의 Heart DATA LOAD
2. 일일 단위로 평균을 내어 심박수 구간 별 칼로리 계산
3. 구간 별 칼로리 합계 도출
4. 사용자 별로 Stacked Bar Chart로 시각화

문제 6



문제 6

도식화 한 Bar Chart를 이용하여 2번 사용자가 누적 칼로리 소비가 가장 많아 체중이 감소할 가능성이 가장 높을 것으로 추정하였고 3번 사용자가 누적 칼로리 소비가 가장 적어 체중이 증가할 가능성이 많아 보였다.



Q & A