[p2]_ _ _ _ _ _

September 17, 2021

1 [Project 2]

1.1

- ,
- , ,

1.2

1. **:** DataFrame 1.1. 1.2.
2. **:** 2.1. 2021 6
3. **:** feature engineering 3.1. 3.2. 3.3. 3.4. 3.5.

1.3

- : http://data.seoul.go.kr/dataList/OA-12252/S/1/datasetView.do

---

1.4

. .

.

1.5 1.

`import pandas` .

1.5.1 1.1.

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

, metro_all .

[2]:
```
# pd.read_csv .
metro_all = pd.read_csv("./data/ _20210705.
↪csv", encoding = 'cp949')
```

[3]:
```
# 5 .
metro_all.head()
```

[3]:

	04 -05	04 -05	05 -06	05 -06	\
0	202106	1	715	14	13235	2131
1	202106	1	51	1	3218	1100
2	202106	1	654	17	9008	6400
3	202106	1	37	0	1881	4340
4	202106	1	343	3	8150	3192

	06 -07	06 -07	07 -08	…	23 -24	00 -01	\
0	8936	6979	14776	…	8211	16
1	3422	4802	5896	…	2589	4
2	12474	37203	37253	…	8024	30
3	2948	21443	6280	…	1485	3
4	8131	10929	17021	…	5451	10

	00 -01	01 -02	01 -02	02 -03	02 -03	\
0	1434	1	1	0	0
1	1348	0	0	0	0
2	637	0	1	0	0
3	92	0	0	0	0
4	449	0	0	0	0

	03 -04	03 -04	
0	0	0	20210703
1	0	0	20210703
2	0	0	20210703
3	0	0	20210703
4	0	0	20210703

[5 rows x 52 columns]

[4]:
```
# .
metro_all.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45338 entries, 0 to 45337
Data columns (total 52 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0                 45338 non-null  int64
 1                 45338 non-null  object
 2                 45338 non-null  object
 3   04 -05        45338 non-null  int64
 4   04 -05        45338 non-null  int64
 5   05 -06        45338 non-null  int64
 6   05 -06        45338 non-null  int64
 7   06 -07        45338 non-null  int64
 8   06 -07        45338 non-null  int64
 9   07 -08        45338 non-null  int64
 10  07 -08        45338 non-null  int64
 11  08 -09        45338 non-null  int64
 12  08 -09        45338 non-null  int64
 13  09 -10        45338 non-null  int64
 14  09 -10        45338 non-null  int64
 15  10 -11        45338 non-null  int64
 16  10 -11        45338 non-null  int64
 17  11 -12        45338 non-null  int64
 18  11 -12        45338 non-null  int64
 19  12 -13        45338 non-null  int64
 20  12 -13        45338 non-null  int64
 21  13 -14        45338 non-null  int64
 22  13 -14        45338 non-null  int64
 23  14 -15        45338 non-null  int64
 24  14 -15        45338 non-null  int64
 25  15 -16        45338 non-null  int64
 26  15 -16        45338 non-null  int64
 27  16 -17        45338 non-null  int64
 28  16 -17        45338 non-null  int64
 29  17 -18        45338 non-null  int64
 30  17 -18        45338 non-null  int64
 31  18 -19        45338 non-null  int64
 32  18 -19        45338 non-null  int64
 33  19 -20        45338 non-null  int64
 34  19 -20        45338 non-null  int64
 35  20 -21        45338 non-null  int64
 36  20 -21        45338 non-null  int64
 37  21 -22        45338 non-null  int64
 38  21 -22        45338 non-null  int64
 39  22 -23        45338 non-null  int64
 40  22 -23        45338 non-null  int64
 41  23 -24        45338 non-null  int64
 42  23 -24        45338 non-null  int64
```

```
 43  00 -01      45338 non-null  int64
 44  00 -01      45338 non-null  int64
 45  01 -02      45338 non-null  int64
 46  01 -02      45338 non-null  int64
 47  02 -03      45338 non-null  int64
 48  02 -03      45338 non-null  int64
 49  03 -04      45338 non-null  int64
 50  03 -04      45338 non-null  int64
 51               45338 non-null  int64
dtypes: int64(50), object(2)
memory usage: 18.0+ MB
```

1.5.2 1.2.

.

[5]:
```
# metro_all DataFrame
sorted(list(set(metro_all['  '])))
```

[5]:
```
[201501,
 201502,
 201503,
 201504,
 201505,
 201506,
 201507,
 201508,
 201509,
 201510,
 201511,
 201512,
 201601,
 201602,
 201603,
 201604,
 201605,
 201606,
 201607,
 201608,
 201609,
 201610,
 201611,
 201612,
 201701,
 201702,
 201703,
 201704,
 201705,
```

201706,
201707,
201708,
201709,
201710,
201711,
201712,
201801,
201802,
201803,
201804,
201805,
201806,
201807,
201808,
201809,
201810,
201811,
201812,
201901,
201902,
201903,
201904,
201905,
201906,
201907,
201908,
201909,
201910,
201911,
201912,
202001,
202002,
202003,
202004,
202005,
202006,
202007,
202008,
202009,
202010,
202011,
202012,
202101,
202102,
202103,
202104,

```
 202105,
 202106]
```

[6]:
```
# metro_all DataFrame
sorted(list(set(metro_all['  '])))
```

[6]:
```
['1 ',
 '2 ',
 '3 ',
 '4 ',
 '5 ',
 '6 ',
 '7 ',
 '8 ',
 '9 ',
 '9 2~3 ',
 '9 2 ',
 '  ',
 '  ',
 '  ',
 '  ',
 '  ',
 '  ',
 '   1 ',
 '  ',
 '  ',
 '  ',
 '  ',
 '   ',
 '  ',
 '  ',
 '  ']
```

[7]:
```
# DataFrame
sorted(list(set(metro_all['  '])))
```

[7]:
```
['4.19  ',
 ' ',
 '  ',
 '    ',
 ' ',
 '  ',
 ' ',
 '  ',
 ' ',
 ' ',
 ' ',
```

```
' ',
'  ',
' ',
'  ',
' ',
' ',
' (   )',
' ',
' ',
' ',
' ',
'  ',
' ',
'  ',
' ',
'  ',
'  ',
' ',
' (   )',
'  ',
' ',
' ',
' ',
' ( )',
' ',
'   ',
' ',
' ',
'  ',
' ',
' ',
' (   )',
'  ',
'   ',
' ',
' ',
' ',
' ( )',
' ',
'   ',
'  ',
'  ',
' (   )',
'  ',
' ( )',
```

```
' ',
' ( .  )',
' ',
'    ',
' ',
' ',
'  ',
' ',
' ',
' ',
' (  )',
' ',
'  ',
' ',
'   ',
' ',
' ( )',
' ',
'  ',
'  ',
'   ',
'  (    )',
' ',
' ',
' ',
'  ',
' ',
' ',
' ',
' ',
' ',
'  ',
'  ',
' ',
' ',
' ',
' ( )',
' ',
'    ',
'   ',
'   (   )',
' ',
' ',
' ',
' ',
'    ',
'   (  .  )',
```

```
' ',
' ',
' ',
' ',
' ',
' (  )',
' ',
' ',
' ',
' ',
'  ',
' ',
' ',
' ',
' ',
' ',
' ',
' ',
' ',
' (  )',
'  ',
' ',
' ',
' ',
' ',
' ',
' ',
' ',
' (  )',
' ',
' ',
' ',
' ',
' ',
' ',
' ',
' ',
' ',
' ',
' ',
' ',
' ',
' ',
' ',
'    ',
'    (DDP)',
```

```
'   ',
'  ',
'    ',
'   ',
'  ',
' ',
'  ',
' ',
' (  )',
' ',
'  ',
'   ',
' ',
'     ',
' ',
'   ',
' ',
'   ',
'   (   )',
' ',
' ',
' ',
' ',
' ',
' ',
' ',
'   ',
' ',
' ',
'  ',
' ',
' ',
' ',
' ',
'   ',
' ',
' ',
' ',
' ',
' ',
' ',
' ',
'   ',
'   (  )',
'  ',
' ',
' ',
' ',
' ',
' ',
```

```
' ',
' ',
' (    )',
'   ',
' ',
' ',
' ',
' ',
' ',
' ',
' ',
' ',
' ',
' ',
' ',
' ',
'  ',
' ',
'   ',
' ',
' ',
' ',
'  ',
' ',
' ',
' ',
' ',
' ',
'  ',
' ',
'  (   )',
' ',
' ',
' ',
'   ',
'    ',
' ',
'   ',
'    ',
'    ',
' ',
'  ',
' ',
' ',
' ',
' ',
' ',
' ',
'  ',
```

```
' ',
'   ',
' ',
' (  )',
'  ',
' ',
' ',
'   ',
' ',
' ',
' ',
' ',
' ',
' ',
'  ',
' ',
' (    )',
' ',
'  ',
'  ',
' (     )',
'  ',
' ',
' ',
' ( )',
' ',
'  ',
'  ',
'  ',
'  ',
'  ',
'   ',
'   (  )',
'  ',
'  ',
'  ',
' ',
' ',
' ',
' ',
' ',
'  ',
' ',
'  ',
'  ',
'  ',
'  ',
'   ',
' ',
'   ',
```

```
'   ( )',
' ',
' ',
' ',
'   ',
'  ',
' ',
' ',
'  ',
'  ',
' ',
' ',
' ',
' ',
' ',
' ',
' ',
'  ',
' ',
'  ',
'  ',
' ',
' ',
' ',
'  ',
' (  )',
' ',
'  ',
'  ( )',
'   ',
'   ( )',
' ',
' ',
' ',
'  ',
' ',
'  ',
' ',
' ',
'  ',
'   ',
'    ',
'  ',
'   ',
' ',
'  ',
'  ',
'  ',
```

```
' ',
'  ',
'  ',
' ',
'  ',
' ',
' ( )',
'   ',
'  ',
' ',
' (  )',
' ',
' ',
' ',
' ',
' ',
' ',
' (  )',
'  ',
' ',
' ',
' ',
'  ',
' (    )',
' ',
' ',
' ',
' ',
' (   )',
' ',
' ',
'  ',
'    ',
'  ',
' ',
' ',
' ',
' ',
' ',
' ',
' ',
' (  )',
' ',
' ',
'  ',
'   ',
' ',
'   ',
```

```
'   ( )',
' ',
' ',
'  ',
' ',
' ',
' ',
' ',
' ',
' ',
' ',
' ',
' ',
'  ',
'   ',
'   ',
' ',
' ',
' ',
' ',
' ',
' ',
' ',
' (   )',
' ',
' ',
' ',
' ',
' ',
' ',
' ',
' (   )',
'  ',
'   ',
'  ( )',
' ',
' ( )',
' ',
' ',
' ',
' (   )',
' ',
' (   )',
' ',
' ',
' ',
' ',
' ',
' ',
```

```
' ',
' ',
'  ',
' ',
' ',
' ',
' (  )',
' ',
'    ',
'   ( )',
' ',
'  3 ',
'  4 ',
'    ',
' ',
' ',
' ',
'  ',
' ',
' ',
' ',
' ',
' ',
' ',
' (    )',
'  ',
'  ',
' ',
'   1  ',
'   2  ',
'    ',
'   ',
'  ',
' ',
' ',
'  ',
' ',
' (  )',
'   ',
'   ',
' ',
'   ',
' ',
' ',
'  ',
' ',
'  ',
'    ',
```

```
' ',
' ',
'  ',
'  ',
' ',
' 3 ',
' 5 ',
'   ',
' ',
' ',
' ',
' ',
' ',
' ',
' ',
' ',
'    ',
' ',
' ',
' ',
' (  )',
' ',
' ',
' ',
' ',
' ',
' ',
' ',
' ',
' ',
'  ',
' ',
' ',
' ',
' ',
' (  )',
' ',
' ',
' ',
' ',
'    ',
'  (    )',
'  ( )',
'  ( )',
' ',
' ',
' ',
' ',
'   ',
'   ( )',
```

```
' ',
' ',
' ',
' ',
' (   )',
' ',
'   ',
' ',
' ',
' ',
' ',
' ',
'   ',
' ',
' ',
'   ',
' ',
' ',
'    ',
'  ( ? )',
'   ',
' ',
' ',
' ',
' ',
' ',
'    ',
'   ( )',
'   ',
' ',
' ',
' ',
' ',
' ',
' ',
' ',
' ',
'   ',
'   ',
' ',
' ',
' ',
' ',
'  ',
' (   )',
' ',
' ',
' ',
' ',
' ',
' ',
```

```
 ' ',
 ' (   )',
 '   ',
 ' (   )']
```

[8]:
```
# DataFrame
len(list(set(metro_all['   '])))
```

[8]: 579

1.6 2.

2015 1 2021 6 .

.

1.6.1 2.1. 2021 6

6 .

[9]:
```
# 2021  6
metro_recent = metro_all[metro_all['  ']==202106]
metro_recent
```

[9]:

			04 -05	04 -05	05 -06	05 -06	\
0	202106	1	715	14	13235	2131	
1	202106	1	51	1	3218	1100	
2	202106	1	654	17	9008	6400	
3	202106	1	37	0	1881	4340	
4	202106	1	343	3	8150	3192	
..	…	…	…	…	…	…	…
603	202106		47	1	350	7	
604	202106		160	1	6077	564	
605	202106		0	0	0	0	
606	202106		2	1	267	132	
607	202106		819	8	12044	3526	

	06 -07	06 -07	07 -08	…	23 -24	\
0	8936	6979	14776	…	8211	
1	3422	4802	5896	…	2589	
2	12474	37203	37253	…	8024	
3	2948	21443	6280	…	1485	
4	8131	10929	17021	…	5451	
..	…	…	…	…	…	…
603	653	225	882	…	219	
604	9670	2216	22839	…	4161	
605	1	0	377	…	0	

```
606           722           675          1546  …           251
607         20777         11468         55410  …         19484

     00 -01       00 -01       01 -02       01 -02       02 -03       \
0                16           1434            1            1            0
1                 4           1348            0            0            0
2                30            637            0            1            0
3                 3             92            0            0            0
4                10            449            0            0            0
..              …             …              …            …            …
603               1             61            0            0            0
604               9            273            0            0            0
605               0              0            0            0            0
606               0              0            0            0            0
607             116           2332            0            0            0

     02 -03       03 -04       03 -04
0                 0            0            0  20210703
1                 0            0            0  20210703
2                 0            0            0  20210703
3                 0            0            0  20210703
4                 0            0            0  20210703
..              …             …              …        …
603               0            0            0  20210703
604               0            0            0  20210703
605               0            0            0  20210703
606               0            0            0  20210703
607               0            0            0  20210703

[608 rows x 52 columns]
```

[10]:
```
#
metro_recent = metro_recent.drop(columns={'   '})
metro_recent
```

[10]:
```
                  04 -05       04 -05       05 -06       05 -06       \
0    202106  1                715           14          13235          2131
1    202106  1                 51            1           3218          1100
2    202106  1                654           17           9008          6400
3    202106  1                 37            0           1881          4340
4    202106  1                343            3           8150          3192
..      …  …  …             …             …            …            …
603  202106                    47            1            350             7
604  202106                   160            1           6077           564
605  202106                     0            0              0             0
606  202106                     2            1            267           132
607  202106                   819            8          12044          3526
```

```
     06 -07      06 -07      07 -08      …  23 -24      \
0             8936          6979          14776  …          2811
1             3422          4802           5896  …          1035
2            12474         37203          37253  …         11581
3             2948         21443           6280  …          4390
4             8131         10929          17021  …          1952
..             …             …              …  …             …
603            653           225            882  …             2
604           9670          2216          22839  …           640
605              1             0            377  …             0
606            722           675           1546  …           178
607          20777         11468          55410  …          5865

     23 -24      00 -01      00 -01      01 -02      01 -02      \
0             8211            16          1434               1             1
1             2589             4          1348               0             0
2             8024            30           637               0             1
3             1485             3            92               0             0
4             5451            10           449               0             0
..             …             …              …               …             …
603            219             1            61               0             0
604           4161             9           273               0             0
605              0             0             0               0             0
606            251             0             0               0             0
607          19484           116          2332               0             0

     02 -03      02 -03      03 -04      03 -04
0             0           0           0           0
1             0           0           0           0
2             0           0           0           0
3             0           0           0           0
4             0           0           0           0
..            …           …           …           …
603           0           0           0           0
604           0           0           0           0
605           0           0           0           0
606           0           0           0           0
607           0           0           0           0

[608 rows x 51 columns]
```

1.7 3.

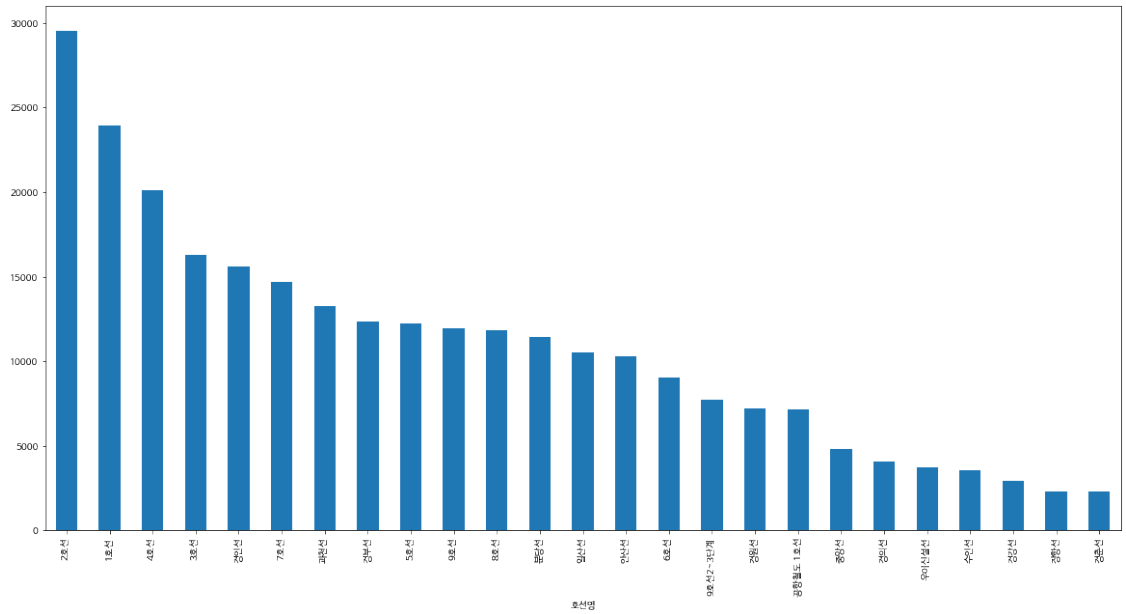
2021 6 `metro_recent` .

1.7.1 3.1.

metro_recent .

[11]:
```
import matplotlib.font_manager as fm

font_dirs = ['/usr/share/fonts/truetype/nanum', ]
font_files = fm.findSystemFonts(fontpaths=font_dirs)

for font_file in font_files:
    fm.fontManager.addfont(font_file)
```

[13]:
```
metro_line = metro_recent.groupby([' ']).mean().reset_index()
metro_line = metro_line.drop(columns=' ').set_index(' ')
metro_line = metro_line.mean(axis=1).sort_values(ascending=False)

plt.figure(figsize=(20,10))
plt.rc('font', family="NanumBarunGothic")
plt.rcParams['axes.unicode_minus'] = False

metro_line.plot(kind=('bar'))
plt.show()
```



1.7.2 3.2.

? 2 .

[23]:
```
line = '6 '
metro_st = metro_recent.groupby([' ',' ']).mean().reset_index()
metro_st_line2 = metro_st[metro_st[' ']==line]
metro_st_line2
```

[23]:

				04 -05	04 -05	05 -06	\
176	6		( )	202106	41	4	3769
177	6			202106	114	2	4643
178	6		( )	202106	7	1	2658
179	6			202106	1	0	5029
180	6		( )	202106	12	0	860
181	6		( )	202106	19	0	2006
182	6			202106	2	0	1971
183	6			202106	29	2	6121
184	6			202106	116	1	4530
185	6			202106	110	4	5529
186	6			202106	17	0	6867
187	6			202106	2	0	5393
188	6			202106	1	0	682
189	6			202106	5	0	3650
190	6		( )	202106	58	3	4808
191	6			202106	28	0	1500
192	6			202106	3	0	1458
193	6			202106	8	0	1209
194	6	(	)	202106	13	1	4695
195	6		( )	202106	6	0	9976
196	6			202106	958	7	8601
197	6			202106	0	0	0
198	6			202106	141	1	7854
199	6		( )	202106	2	0	1395
200	6			202106	5	0	1432
201	6			202106	2	0	1973
202	6			202106	0	0	0
203	6		( )	202106	14	0	6581
204	6		( )	202106	1	0	1113
205	6			202106	21	0	11358
206	6			202106	2	0	2060
207	6		( )	202106	9	3	7663
208	6			202106	2	0	2386
209	6			202106	2	0	1708
210	6			202106	10	1	4295
211	6			202106	2	0	541
212	6			202106	3	0	3550
213	6		( )	202106	15	0	5879
214	6			202106	29	0	2145

05 -06	06 -07	06 -07	07 -08	…	\

176	899	7132	3956	20845	…
177	1766	7647	10036	20724	…
178	1214	6802	4507	18523	…
179	215	11342	2049	29522	…
180	400	1846	4167	4758	…
181	868	5458	4491	15004	…
182	256	5083	982	13838	…
183	782	13916	3073	36163	…
184	687	4662	4348	8598	…
185	3207	13638	13304	44088	…
186	2218	13088	8084	39316	…
187	787	9804	4925	26835	…
188	325	1775	1319	4733	…
189	932	7160	3807	18599	…
190	1073	14992	2945	48580	…
191	468	3769	2773	10280	…
192	640	3306	4591	7738	…
193	968	3031	2923	8201	…
194	508	9008	1904	25215	…
195	604	20504	4906	54785	…
196	1869	19573	4513	48711	…
197	0	0	0	0	…
198	874	5365	6169	6579	…
199	1167	2989	5343	7996	…
200	241	1870	1900	5091	…
201	265	5822	1829	15852	…
202	0	0	0	0	…
203	882	12769	4034	39801	…
204	1141	2491	3081	8391	…
205	1161	25522	5369	73765	…
206	1050	4506	7071	8905	…
207	744	15590	3667	40317	…
208	303	5279	2059	14596	…
209	504	2999	2344	7070	…
210	845	7117	2756	18472	…
211	557	1171	6738	3023	…
212	1185	5830	6529	13609	…
213	943	21485	3262	60170	…
214	404	4930	3227	16099	…

	23 -24	23 -24	00 -01	00 -01	01 -02	\	
176		1623	6587	1	122		0
177		4463	5690	24	1615		0
178		1608	4107	1	102		0
179		790	4140	0	407		0
180		2228	1785	0	137		0
181		2287	4417	1	175		0

182	249	3170	0	252	0
183	1472	7513	0	121	0
184	1535	7267	0	821	0
185	1920	7187	30	1186	1
186	2568	8910	0	58	0
187	4391	7538	0	0	0
188	410	1443	0	77	0
189	1761	6435	0	133	0
190	778	7348	3	594	0
191	440	3024	2	35	0
192	1904	2686	0	119	0
193	4308	3304	0	39	0
194	1015	5314	0	102	0
195	1203	11213	14	1927	0
196	2166	11588	23	705	0
197	0	0	0	0	0
198	2960	9052	5	964	0
199	4807	5947	4	601	0
200	914	1266	3	78	0
201	371	3980	0	1	0
202	0	0	0	0	0
203	1921	8399	1	141	0
204	952	1459	0	79	0
205	1678	16630	1	773	0
206	4394	4547	2	473	0
207	1245	8802	1	943	0
208	399	2422	0	52	0
209	924	1774	2	177	0
210	1150	6270	4	725	0
211	2105	1364	0	746	0
212	5930	5629	2	45	0
213	1030	8939	0	603	0
214	1213	3361	1	332	0

	01 -02	02 -03	02 -03	03 -04	03 -04	
176	0	0	0	0	0	
177	0	0	0	0	0	
178	0	0	0	0	0	
179	0	0	0	0	0	
180	0	0	0	0	0	
181	0	0	0	0	0	
182	0	0	0	0	0	
183	0	0	0	0	0	
184	0	0	0	0	0	
185	2	0	0	0	0	
186	0	0	0	0	0	
187	0	0	0	0	0	

```
188              0              0              0              0              0
189              0              0              0              0              0
190              0              0              0              0              0
191              0              0              0              0              0
192              0              0              0              0              0
193              0              0              0              0              0
194              0              0              0              0              0
195              0              0              0              0              0
196              0              0              0              0              0
197              0              0              0              0              0
198              0              0              0              0              0
199              0              0              0              0              0
200              0              0              0              0              0
201              0              0              0              0              0
202              0              0              0              0              0
203              0              0              0              0              0
204              0              0              0              0              0
205              0              0              0              0              0
206              0              0              0              0              0
207              0              0              0              0              0
208              0              0              0              0              0
209              0              0              0              0              0
210              0              0              0              0              0
211              0              0              0              0              0
212              0              0              0              0              0
213              0              0              0              0              0
214              0              0              0              0              0

[39 rows x 51 columns]
```

[24]:
```
#
metro_get_on = pd.DataFrame()
metro_get_on['  '] = metro_st_line2['  ']
for i in range(int((len(metro_recent.columns)-3)/2)):
    metro_get_on[metro_st_line2.columns[3+2*i]] = metro_st_line2[metro_st_line2.
 ↪columns[3+2*i]]
metro_get_on = metro_get_on.set_index('  ')
metro_get_on
```

[24]:
```
                04 -05       05 -06       06 -07       07 -08       \

  ( )                41           3769          7132          20845
                    114           4643          7647          20724
  ( )                 7           2658          6802          18523
                      1           5029          11342         29522
  (  )               12           860           1846          4758
 (  )                19           2006          5458          15004
```

	2	1971	5083	13838
	29	6121	13916	36163
	116	4530	4662	8598
	110	5529	13638	44088
	17	6867	13088	39316
	2	5393	9804	26835
	1	682	1775	4733
	5	3650	7160	18599
( )	58	4808	14992	48580
	28	1500	3769	10280
	3	1458	3306	7738
	8	1209	3031	8201
( )	13	4695	9008	25215
( )	6	9976	20504	54785
	958	8601	19573	48711
	0	0	0	0
	141	7854	5365	6579
( )	2	1395	2989	7996
	5	1432	1870	5091
	2	1973	5822	15852
	0	0	0	0
( )	14	6581	12769	39801
( )	1	1113	2491	8391
	21	11358	25522	73765
	2	2060	4506	8905
( )	9	7663	15590	40317
	2	2386	5279	14596
	2	1708	2999	7070
	10	4295	7117	18472
	2	541	1171	3023
	3	3550	5830	13609
( )	15	5879	21485	60170
	29	2145	4930	16099

	08 -09	09 -10	10 -11	11 -12	\
( )	25062	13055	8823	8791	
	28121	18342	15213	16745	
( )	24132	14635	10990	11183	
	35106	17295	10961	9851	
( )	7852	5080	4065	4307	
( )	19996	11584	8429	8235	
	13383	6255	3900	3993	
	39620	19350	12622	10647	
	12551	10830	10055	12053	
	47064	23676	14196	12147	
	52128	29930	19320	17419	

	38323	22657	16105	15597
	7145	3947	2582	2533
	28111	14943	9102	8414
( )	39241	17094	12776	10591
	11862	6477	5609	5968
	11894	7616	5310	6351
	12729	7970	5909	6512
( )	28403	14400	8861	8067
( )	63193	30188	18211	15931
	53987	27236	18551	15623
	1	0	0	0
	9532	8589	7808	9587
( )	10571	9066	10485	12606
	7128	5193	4251	4281
	16781	8766	5761	5039
	1	0	3	1
( )	43486	22329	14954	13739
( )	10599	5626	5322	6251
	88640	42050	25226	22298
	13616	9764	7473	8513
( )	47834	24752	15823	12965
	16394	8401	4768	4043
	9968	5686	4291	3917
	20939	10689	7254	6500
	4648	3686	2897	3495
	20314	13845	10704	11781
( )	53016	23726	15312	13334
	24111	13277	8849	7909

	12 -13	13 -14	…	18 -19	19 -20	\
				…		
( )	9686	11063	…	18192	8782	
	18643	20191	…	86562	33672	
( )	10477	10349	…	31203	13198	
	10528	9548	…	8769	5109	
( )	4944	5882	…	17950	10427	
( )	10494	10298	…	24401	11983	
	3747	3646	…	2705	1674	
	11506	11812	…	12958	8942	
	14600	14369	…	23245	12142	
	12339	11462	…	50819	20044	
	17595	17431	…	36566	16429	
	18273	18907	…	36915	24012	
	2797	2593	…	5331	2854	
	9194	9109	…	16036	7925	
( )	10559	9770	…	9959	4960	
	7007	7594	…	9329	5993	

	7658	8549 …	22954	10210
	8342	9628 …	26454	17657
( )	7920	7424 …	9533	6150
( )	16571	15602 …	14508	8951
	16055	15616 …	13595	9978
	0	1 …	2	1
	10598	9391 …	25431	14109
( )	13238	12962 …	29410	15585
	4707	4968 …	12527	5726
	5311	5148 …	5114	2610
	3	1 …	2	1
( )	14019	13691 …	23465	11658
( )	6619	6931 …	12811	7006
	23218	22126 …	20774	12451
	10300	11614 …	28878	20833
( )	12891	12544 …	12100	7382
	4034	3819 …	4540	2579
	4032	4356 …	13493	8365
	7397	7490 …	8103	5363
	4814	5951 …	31446	21903
	14006	15809 …	45512	28090
( )	13376	12608 …	11787	6686
	8878	8665 …	21635	9655

	20 –21	21 –22	22 –23	23 –24	\
( )	6527	7069	6327	1623	
	24738	26720	24656	4463	
( )	8551	7932	6728	1608	
	3928	3845	3576	790	
( )	9383	12054	14436	2228	
( )	8765	8979	9576	2287	
	950	807	691	249	
	5333	5147	4958	1472	
	7855	7118	5753	1535	
	10258	8491	7227	1920	
	11804	12241	9491	2568	
	20407	21931	19240	4391	
	1806	1593	1652	410	
	6040	5799	4856	1761	
( )	3673	3420	2978	778	
	4386	4232	2815	440	
	7530	7948	8268	1904	
	14871	20636	25372	4308	
( )	3011	2594	2204	1015	
( )	6304	5812	4858	1203	
	7322	7314	6517	2166	

	0	0	0	0
	11398	12658	13410	2960
( )	13427	15926	16043	4807
	4588	4973	4164	914
	1900	1663	1491	371
	1	0	3	0
( )	8142	7438	5461	1921
( )	5084	4442	3739	952
	9681	8703	7069	1678
	19010	28096	29870	4394
( )	5115	4597	3871	1245
	2163	1882	1291	399
	4833	4363	3778	924
	3944	3857	3600	1150
	17247	24489	26973	2105
	24186	32333	32825	5930
( )	4530	3872	3246	1030
	5730	5637	4871	1213

	00 -01	01 -02	02 -03	03 -04
( )	1	0	0	0
	24	0	0	0
( )	1	0	0	0
	0	0	0	0
( )	0	0	0	0
( )	1	0	0	0
	0	0	0	0
	0	0	0	0
	0	0	0	0
	30	1	0	0
	0	0	0	0
	0	0	0	0
	0	0	0	0
	0	0	0	0
( )	3	0	0	0
	2	0	0	0
	0	0	0	0
	0	0	0	0
( )	0	0	0	0
( )	14	0	0	0
	23	0	0	0
	0	0	0	0
	5	0	0	0
( )	4	0	0	0
	3	0	0	0
	0	0	0	0

	0	0	0	0
( )	1	0	0	0
( )	0	0	0	0
	1	0	0	0
	2	0	0	0
( )	1	0	0	0
	0	0	0	0
	2	0	0	0
	4	0	0	0
	0	0	0	0
	2	0	0	0
( )	0	0	0	0
	1	0	0	0

[39 rows x 24 columns]

[25]:
```
#
metro_get_off = pd.DataFrame()
metro_get_off['  '] = metro_st_line2['  ']
for i in range(int((len(metro_recent.columns)-3)/2)):
    metro_get_off[metro_st_line2.columns[4+2*i]] =␣
 ↪metro_st_line2[metro_st_line2.columns[4+2*i]]
metro_get_off = metro_get_off.set_index('  ')
metro_get_off
```

[25]:

	04 -05	05 -06	06 -07	07 -08	\
( )	4	899	3956	10687	
	2	1766	10036	37229	
( )	1	1214	4507	16694	
	0	215	2049	7109	
( )	0	400	4167	9942	
( )	0	868	4491	14288	
	0	256	982	1676	
	2	782	3073	5800	
	1	687	4348	9436	
	4	3207	13304	26911	
	0	2218	8084	14535	
	0	787	4925	9019	
	0	325	1319	2662	
	0	932	3807	8480	
( )	3	1073	2945	5593	
	0	468	2773	4213	
	0	640	4591	13051	
	0	968	2923	6327	
( )	1	508	1904	3483	
( )	0	604	4906	7661	

	7	1869	4513	7831
	0	0	0	0
	1	874	6169	16263
( )	0	1167	5343	16530
	0	241	1900	5984
	0	265	1829	2761
	0	0	0	0
( )	0	882	4034	6833
( )	0	1141	3081	5507
	0	1161	5369	9638
	0	1050	7071	10643
( )	3	744	3667	5343
	0	303	2059	6774
	0	504	2344	6993
	1	845	2756	3967
	0	557	6738	19009
	0	1185	6529	17025
( )	0	943	3262	7335
	0	404	3227	6579

	08 –09	09 –10	10 –11	11 –12	\
( )	25975	11572	7380	7128	
	97068	48896	21788	16948	
( )	36554	16104	9547	8056	
	7399	4975	4592	5876	
( )	18165	8081	6692	6083	
( )	31359	15648	10007	8620	
	2994	3400	3384	2709	
	15772	9762	8330	7565	
	26699	19651	14208	14003	
	72484	29279	11349	9545	
	35605	17216	12645	11341	
	32551	29285	19617	21624	
	4817	2980	2307	2049	
	18999	9162	6155	5540	
( )	12316	6540	5868	5302	
	7378	6794	5454	5697	
	22477	9060	6300	5874	
	24687	19716	14104	12953	
( )	12566	6506	5057	4844	
( )	14732	9323	8549	9494	
	13140	9810	9102	10461	
	0	0	0	0	
	39255	21843	12443	9904	
( )	35268	31528	17977	12973	
	16933	8102	4933	4521	

	8468	4444	3435	4690
	0	0	0	0
( )	23053	15716	11024	10336
( )	9574	8695	9140	9170
	18499	11789	12023	13761
	25199	21120	13631	13916
( )	10423	7691	6845	7723
	5990	3131	2420	2345
	24775	10460	4688	3803
	7696	6292	5076	5792
	35317	27053	14779	15113
	43415	26847	15753	16856
( )	12636	8195	6647	6964
	22428	12463	7126	6740

	12 –13	13 –14	…	18 –19	19 –20	\
				…		
( )	7355	9334	…	20719	17437	
	17189	17322	…	37639	23703	
( )	8346	9397	…	21451	16211	
	6018	6958	…	19494	19110	
( )	7584	8263	…	14076	8746	
( )	10723	11241	…	21641	14225	
	2901	2854	…	10973	10643	
	9084	9033	…	30487	26520	
	12676	13983	…	15325	11743	
	10397	11182	…	40319	30687	
	13818	14492	…	40811	30349	
	24922	26761	…	48045	33596	
	2104	2493	…	6099	4811	
	6362	7045	…	20471	16263	
( )	8190	8995	…	30358	28863	
	5831	6295	…	14641	11013	
	6242	6373	…	12109	7964	
	14355	16468	…	28390	18382	
( )	5522	5782	…	18376	16366	
( )	11354	12586	…	48506	40639	
	12638	12879	…	49803	43313	
	0	0	…	0	0	
	8240	9492	…	14312	16631	
( )	14619	17438	…	17294	13215	
	4212	4809	…	8207	5607	
	4576	4977	…	16298	13786	
	0	0	…	0	0	
( )	11392	11871	…	38021	32128	
( )	9355	10352	…	12167	8894	
	16386	16942	…	73460	60244	

	15180	16853	…	31235	22000
( )	9077	10324	…	40843	32826
	2739	2974	…	9137	7381
	3694	4237	…	7174	5457
	5954	6881	…	22313	19264
	14705	19797	…	23065	11267
	17335	19908	…	38335	24600
( )	7490	8466	…	34865	31867
	7247	7671	…	18284	12868

	20 -21	21 -22	22 -23	23 -24	\
( )	10363	10558	11800	6587	
	13260	12679	15259	5690	
( )	9642	8640	11176	4107	
	9477	8417	8517	4140	
( )	4610	3623	4447	1785	
( )	8279	7643	9423	4417	
	5948	5087	5718	3170	
	15256	12356	15148	7513	
	7420	7048	8527	7267	
	17289	16214	18486	7187	
	18176	16722	20810	8910	
	16900	14653	15531	7538	
	2841	2943	3608	1443	
	10257	9858	12829	6435	
( )	16259	13291	15370	7348	
	5687	5071	5778	3024	
	5464	5195	7860	2686	
	8701	6148	7039	3304	
( )	9882	9169	10604	5314	
( )	22756	20385	22810	11213	
	23292	20306	22777	11588	
	0	0	0	0	
	8680	6274	7270	9052	
( )	8346	8685	10899	5947	
	2926	2698	3286	1266	
	6897	7356	7605	3980	
	0	0	0	0	
( )	16716	16561	17285	8399	
( )	4944	3826	3764	1459	
	36580	29467	33964	16630	
	11395	8503	9262	4547	
( )	18483	18467	19843	8802	
	4466	4573	5538	2422	
	3618	3258	3940	1774	
	9512	9530	10841	6270	

	3088	2692	2913	1364
	11474	10384	12199	5629
( )	17781	15275	16692	8939
	7663	7495	8302	3361

	00 -01	01 -02	02 -03	03 -04	
( )	122	0	0	0	
	1615	0	0	0	
( )	102	0	0	0	
	407	0	0	0	
( )	137	0	0	0	
( )	175	0	0	0	
	252	0	0	0	
	121	0	0	0	
	821	0	0	0	
	1186	2	0	0	
	58	0	0	0	
	0	0	0	0	
	77	0	0	0	
	133	0	0	0	
( )	594	0	0	0	
	35	0	0	0	
	119	0	0	0	
	39	0	0	0	
( )	102	0	0	0	
( )	1927	0	0	0	
	705	0	0	0	
	0	0	0	0	
	964	0	0	0	
( )	601	0	0	0	
	78	0	0	0	
	1	0	0	0	
	0	0	0	0	
( )	141	0	0	0	
( )	79	0	0	0	
	773	0	0	0	
	473	0	0	0	
( )	943	0	0	0	
	52	0	0	0	
	177	0	0	0	
	725	0	0	0	
	746	0	0	0	
	45	0	0	0	
( )	603	0	0	0	
	332	0	0	0	

[39 rows x 24 columns]

[26]:
```
#
df = pd.DataFrame(index = metro_st_line2['  '])
df['       '] = metro_get_on.mean(axis=1).astype(int)
df['       '] = metro_get_off.mean(axis=1).astype(int)
df
```

[26]:

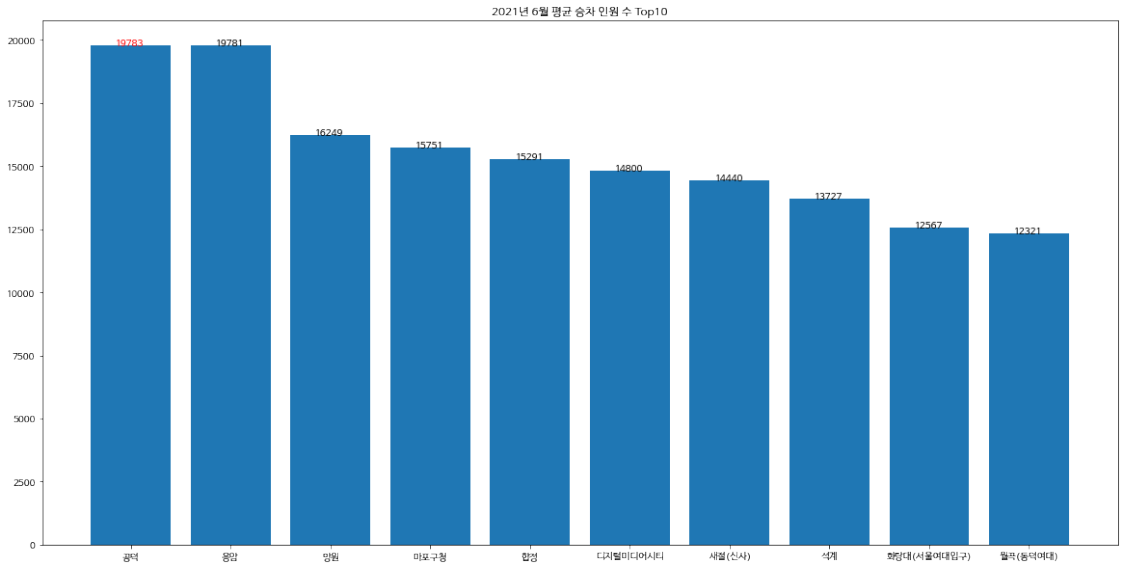
( )	8761	8350
	19783	19001
( )	9772	9385
	8504	6408
( )	5997	5918
( )	8862	9121
	3167	3376
	10229	9590
	9367	9427
	14800	15997
	15751	14197
	16249	17694
	2290	2328
	8016	7405
( )	9706	9110
	5158	5031
	6993	5966
	9413	10550
( )	6980	6289
( )	14440	13553
	13727	14035
	0	0
	8857	9253
( )	11233	11202
	4026	3992
	4265	4948
	0	0
( )	12321	12226
( )	5522	5794
	19781	19512
	11739	12151
( )	11287	11150
	3992	3289
	4208	4417
	6072	6718
	9250	10815
	15291	14702

( ) 12567 9989
7757 7009

1.7.3 3.3.

2 6 > > > > > .

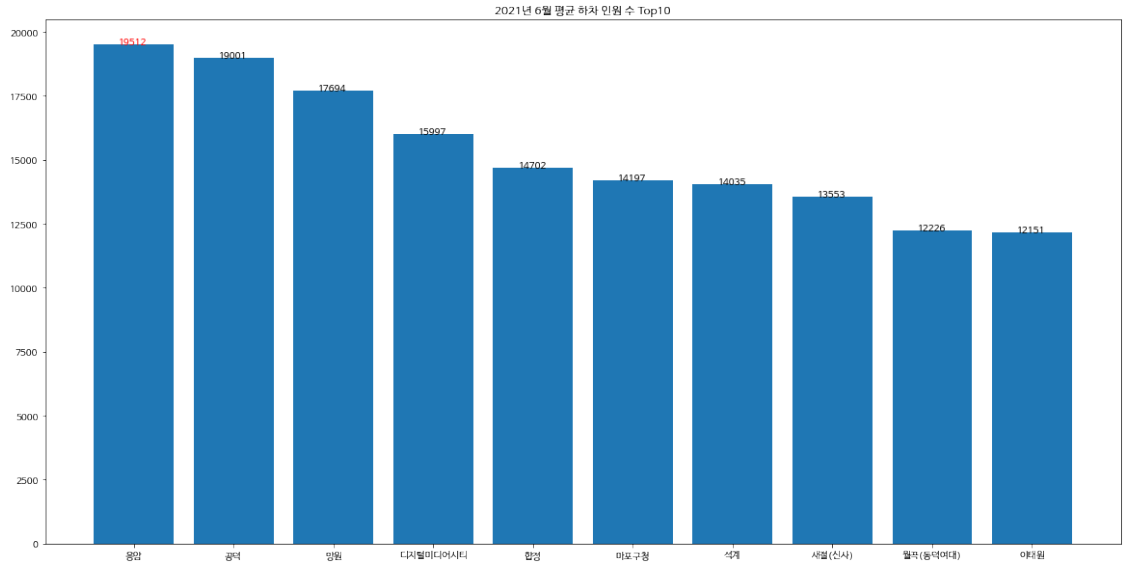
[27]:
```
#       Top10
top10_on = df.sort_values(by='       ', ascending=False).head(10)

plt.figure(figsize=(20,10))
plt.rc('font', family="NanumBarunGothic")
plt.rcParams['axes.unicode_minus'] = False

plt.bar(top10_on.index, top10_on['       '])
for x, y in enumerate(list(top10_on['       '])):
    if x == 0:
        plt.annotate(y, (x-0.15, y), color = 'red')
    else:
        plt.annotate(y, (x-0.15, y))

plt.title('2021  6         Top10')
plt.show()
```



> > > > > .

[28]:
```
#       Top10
top10_off = df.sort_values(by='       ', ascending=False).head(10)
```

```
plt.figure(figsize=(20,10))
plt.rc('font', family="NanumBarunGothic")
plt.rcParams['axes.unicode_minus'] = False

plt.bar(top10_off.index, top10_off['       '])
for x, y in enumerate(list(top10_off['       '])):
    if x == 0:
        plt.annotate(y, (x-0.15, y), color = 'red')
    else:
        plt.annotate(y, (x-0.15, y))

plt.title('2021  6         Top10')
plt.show()
```



1. 6 .

[31]:
```
# 3.2.        line
# 3.2.  3.3.                  .

top_on = df.sort_values(by='       ', ascending=False).head(1)
top_on.index[0]
```

[31]: ' '

[32]:
```
# 6                   quiz_1       .
# '~~ '       ~~        .
quiz_1 = top_on.index[0]
quiz_1
```

[32]: ' '

1.7.4 3.4.

.

API csv .

: https://developers.kakao.com/docs/latest/ko/local/dev-guide#search-by-keyword
https://developers.kakao.com/docs/latest/ko/local/dev-guide#address-coord

[33]:
```
#                 .
subway_location = pd.read_csv('./data/       .csv')
subway_location
```

[33]:

			x	y
0	4.19	72-182	37.649457	127.013506
1		197-1	37.747906	127.044358
2		184-23	37.492915	127.118215
3		468-4	37.482414	126.882240
4		14-61	37.561758	126.853997
..	…	…	…	…
574		50-5	37.713908	127.046619
575		64-1	37.557688	126.976720
576		64-1	37.557688	126.976720
577		80	37.539622	126.960984
578		4	37.508502	126.964009

[579 rows x 4 columns]

.

[34]:
```
#                                   .
def get_nums_and_location(line, metro_st):

    #           .
    metro_line_n = metro_st[metro_st['  ']==line]

    #           .
    metro_get_on = pd.DataFrame()
    metro_get_on['  '] = metro_line_n['  ']
    for i in range(int((len(metro_recent.columns)-3)/2)):
        metro_get_on[metro_line_n.columns[3+2*i]] = metro_line_n[metro_line_n.
↪columns[3+2*i]]
    metro_get_on = metro_get_on.set_index('  ')

    #           .
    metro_get_off = pd.DataFrame()
    metro_get_off['  '] = metro_line_n['  ']
```

```
    for i in range(int((len(metro_recent.columns)-3)/2)):
        metro_get_off[metro_line_n.columns[4+2*i]] = metro_line_n[metro_line_n.
↪columns[4+2*i]]
    metro_get_off = metro_get_off.set_index('   ')

    #                                .
    df = pd.DataFrame(index = metro_line_n['   '])
    df['       '] = metro_get_on.mean(axis=1).astype(int)
    df['       '] = metro_get_off.mean(axis=1).astype(int)

    #            .
    temp = []
    df = df.reset_index()
    for name in df['   ']:
        temp.append(name.split('(')[0]+' ')
    df['   '] = temp

    #                  .
    df = df.merge(subway_location, left_on='   ', right_on='   ')
    return df
```

[35]:
```
get_nums_and_location('6 ', metro_st)
```

[35]:

				x	\
0	8761	8350	29-18	37.589679	
1	8761	8350	29-18	37.589679	
2	19783	19001	423-29	37.544487	
3	9772	9385	145-17	37.547426	
4	9772	9385	145-17	37.547426	
5	8504	6408	1	37.611212	
6	5997	5918	4 4	37.534446	
7	8862	9121	128-1	37.547730	
8	8862	9121	128-1	37.547730	
9	3167	3376	13-33	37.618377	
10	10229	9590	349-8	37.610606	
11	9367	9427	117	37.573597	
12	14800	15997	223-25	37.577451	
13	15751	14197	592	37.563426	
14	16249	17694	378	37.556057	
15	2290	2328	366-454	37.548187	
16	8016	7405	1 127-1	37.585888	
17	9706	9110	643-1	37.616424	
18	5158	5031	13-10	37.610345	
19	6993	5966	1 228-1	37.534653	
20	9413	10550	309-10	37.547943	
21	6980	6289	26-1	37.607086	
22	6980	6289	26-1	37.607086	

23	14440	13553		337-5	37.591160
24	14440	13553		337-5	37.591160
25	13727	14035		36-4	37.615879
26	0	0		19	37.612057
27	8857	9253		99	37.565325
28	11233	11202	5	146-1	37.586092
29	11233	11202	5	146-1	37.586092
30	4026	3992		369-44	37.554449
31	4265	4948		153-31	37.605670
32	0	0		397	37.618933
33	12321	12226		35-1	37.601750
34	12321	12226		35-1	37.601750
35	5522	5794		420	37.569943
36	5522	5794		420	37.569943
37	19781	19512		22-15	37.598807
38	11739	12151		119-23	37.534551
39	11287	11150		199-8	37.583651
40	11287	11150		199-8	37.583651
41	3992	3289		20-8	37.579450
42	4208	4417		295-2	37.560349
43	6072	6718		616-4	37.619143
44	9250	10815		726-494	37.540807
45	15291	14702		393	37.550115
46	12567	9989		285-2	37.619896
47	12567	9989		285-2	37.619896
48	7757	7009		80	37.539622

	y
0	127.035926
1	127.035926
2	126.951195
3	126.932477
4	126.932477
5	126.917182
6	126.985525
7	126.942379
8	126.942379
9	126.932857
10	127.057147
11	127.017139
12	126.902154
13	126.903357
14	126.910034
15	127.007066
16	127.019705
17	127.093316
18	126.929907

```
19  126.973222
20  126.922937
21  127.049831
22  127.049831
23  126.913285
24  126.913285
25  127.065393
26  127.109058
27  127.016667
28  127.029372
29  127.029372
30  127.010991
31  126.923455
32  126.920853
33  127.041416
34  127.041416
35  126.899033
36  126.899033
37  126.914482
38  126.994729
39  126.909377
40  126.909377
41  127.015190
42  127.013871
43  127.075136
44  127.001841
45  126.914638
46  127.083287
47  127.083287
48  126.960984
```

1.7.5 3.5.

folium .

[36]:
```
import folium

#    ,        OpenStreetMap
map_osm = folium.Map(location = [37.529622, 126.984307], zoom_start=12)
map_osm
```

[36]:
```
<folium.folium.Map at 0x7fd7340a3580>
```

.

[37]:
```
#                          .
rail = '6 '
df = get_nums_and_location(rail, metro_st)
```

```
#                          .
latitude = subway_location[subway_location['  ']=='  ']['x ']
longitude = subway_location[subway_location['  ']=='  ']['y ']
map_osm = folium.Map(location = [latitude, longitude], zoom_start = 12)

#                  .
for i in df.index:
    marker = folium.CircleMarker([df['x '][i],df['y '][i]],
                        radius = (df['       '][i]+1)/3000, #     0
                        popup = [df['  '][i],df['       '][i]],
                        color = 'blue',
                        fill_color = 'blue')
    marker.add_to(map_osm)

map_osm
```

[37]: <folium.folium.Map at 0x7fd733af0400>

2. **x ( )** **.**

[ ]:
```
# get_nums_and_location()              .
#    2      df = get_nums_and_location('2 ', metro_st)          .
# df[df['  ']==' ']['x ']      '  ' ' ' '       'x '     .

df = get_nums_and_location('2 ', metro_st)
x = df[df['  '] == '  ']['x ']
x[0]
```

[ ]:
```
# float         .   : 37.123456
quiz_2 = x[0]
quix_2
```

1.8

1 2 , quiz_1 ~ 2 csv .

.

[ ]:
```
d = {'quiz_1': [quiz_1], 'quiz_2': [quiz_2]}
df_quiz = pd.DataFrame(data=d)
df_quiz.to_csv("submission.csv",index=False)
```

[ ]:
```
#
import sys
sys.path.append('vendor')
from elice_challenge import check_score, upload
```

[ ]:
```
#
await upload()
```

[ ]:
```
#
await check_score()
```