[p1]_ _ _ _

September 14, 2021

1 [Project 1]

1.1

- 19
- , ,

1.2

1. : Dataframe 1.1.
2. : 2.1. column
3. : feature engineering 3.1. 3.2. 3.3. 8 3.4. 3.5. 8 3.6. 3.7.

1.3

- https://www.data.go.kr/tcs/dss/selectFileDataDetailView.do?publicDataPk=15063273

1.4

2020 19 . 6 .

`19` . , 19 .

1.5 1.

`import` `pandas` .

1.5.1 1.1.

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

```
[2]: # pd.read_csv     dataframe       .
     corona_all=pd.read_csv("./data/    19      .csv")
```

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

```
[3]:                                                                     \
     0  5748  10.21.  25530 NaN   NaN      NaN            NaN  NaN  NaN
     1  5747  10.21.  25528 NaN   NaN      NaN           NaN  NaN  NaN
     2  5746  10.21.  25525 NaN   NaN      NaN            NaN  NaN  NaN
     3  5745  10.21.  25517 NaN   NaN      NaN            NaN  NaN  NaN
     4  5744  10.21.  25504 NaN   NaN      NaN            NaN  NaN  NaN



     0  2020-10-22 10:58  2020-10-22 10:58    Y
     1  2020-10-22 10:58  2020-10-22 10:58    Y
     2  2020-10-22 10:58  2020-10-22 10:58    Y
     3  2020-10-22 10:58  2020-10-22 10:58    Y
     4  2020-10-22 10:58  2020-10-22 10:58    Y
```

```
[4]: # dataframe          .
     corona_all.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5748 entries, 0 to 5747
Data columns (total 14 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0          5748 non-null   int64
 1          5748 non-null   object
 2          5748 non-null   int64
 3           0 non-null      float64
 4          0 non-null      float64
 5           5748 non-null   object
 6          459 non-null     object
 7          5748 non-null   object
 8          0 non-null      float64
 9           5357 non-null   object
 10         5520 non-null   object
 11         5748 non-null   object
 12         5748 non-null   object
 13         5748 non-null   object
dtypes: float64(3), int64(2), object(9)
memory usage: 628.8+ KB
```

1.6 2.

(missing data), (outlier) .

1.6.1 2.1. column

`corona_all.info()` , , .

`dataframe.drop()` , , column dataframe `corona_del_col` .

[5]:

```
# drop          ,    ,      coulmn         .
corona_del_col = corona_all.drop(columns = ['  ','   ','   '])
```

[6]:

```
#      dataframe        .
corona_del_col.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5748 entries, 0 to 5747
Data columns (total 11 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0           5748 non-null   int64
 1          5748 non-null   object
 2         5748 non-null   int64
 3           5748 non-null   object
 4          459 non-null    object
 5          5748 non-null   object
 6           5357 non-null   object
 7         5520 non-null   object
 8          5748 non-null   object
 9          5748 non-null   object
 10        5748 non-null   object
dtypes: int64(2), object(9)
memory usage: 494.1+ KB
```

1.7 3.

`corona_del_col` column .

1.7.1 3.1.

. .

,   .

[7]:
```
corona_del_col['  ']
```

[7]:
```
0       10.21.
1       10.21.
2       10.21.
3       10.21.
4       10.21.
         …
5743     1.31.
5744     1.30.
5745     1.30.
5746     1.30.
5747     1.24.
Name:   , Length: 5748, dtype: object
```

month, day   month, day column int64   .

[8]:
```
# dataframe      ,            list     .
month = []
day = []

for data in corona_del_col['  ']:
    # split         ,       list     .
    month.append(data.split('.')[0])
    day.append(data.split('.')[1])
```

[9]:
```
# corona_del_col  `month`, `day` column        list              .
corona_del_col['month'] = month
corona_del_col['day'] = day

corona_del_col['day'].astype('int64')
corona_del_col['month'].astype('int64')
```

[9]:
```
0       10
1       10
2       10
3       10
4       10
        ..
5743     1
5744     1
5745     1
5746     1
5747     1
Name: month, Length: 5748, dtype: int64
```

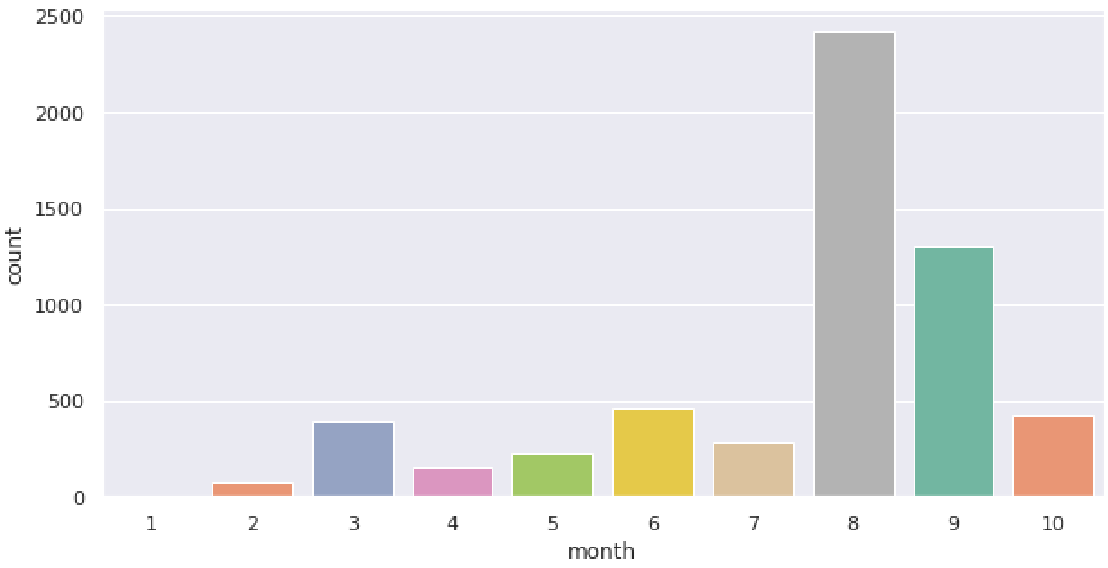
1.7.2 3.2.

month .

[10]:
```
#     x             order list     .
order = []
for i in range(1,11):
    order.append(str(i))

order
```

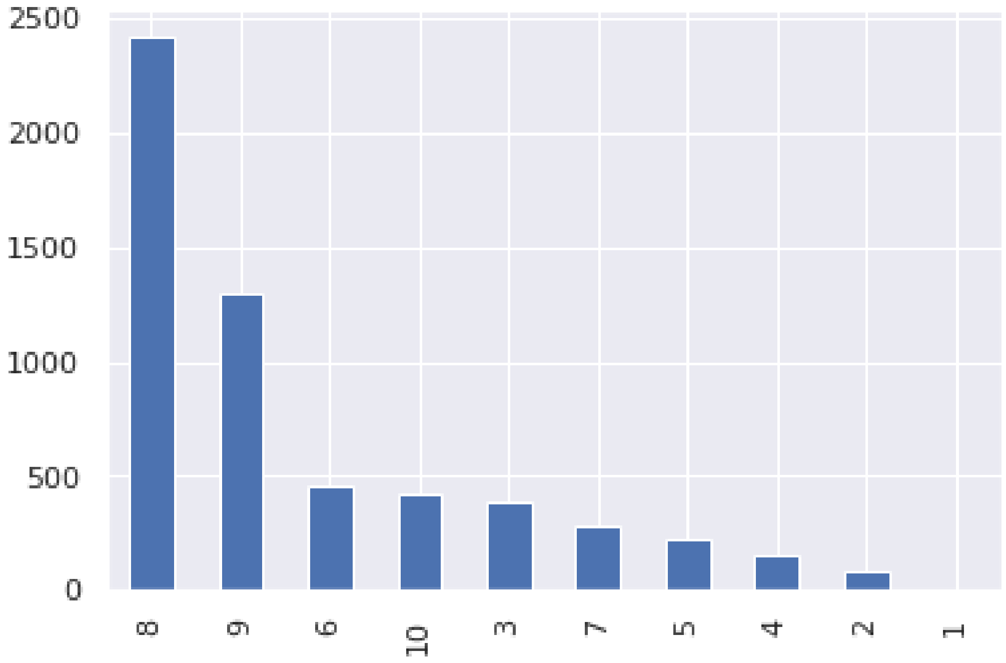
[10]:
```
['1', '2', '3', '4', '5', '6', '7', '8', '9', '10']
```

[11]:
```
#             .
plt.figure(figsize=(10,5))

# seaborn  countplot          .
sns.set(style="darkgrid")
ax = sns.countplot(x="month", data=corona_del_col, palette="Set2", order =␣
 ↪order)
```



[12]:
```
# series  plot             .
corona_del_col['month'].value_counts().plot(kind='bar')
```

[12]:
```
<AxesSubplot:>
```



[13]:

```
# value_counts()                    .
corona_del_col['month'].value_counts()
```

[13]:

```
8     2416
9     1304
6      460
10     425
3      391
7      281
5      228
4      156
2       80
1        7
Name: month, dtype: int64
```

1.7.3 3.3. 8

8 .

8 .

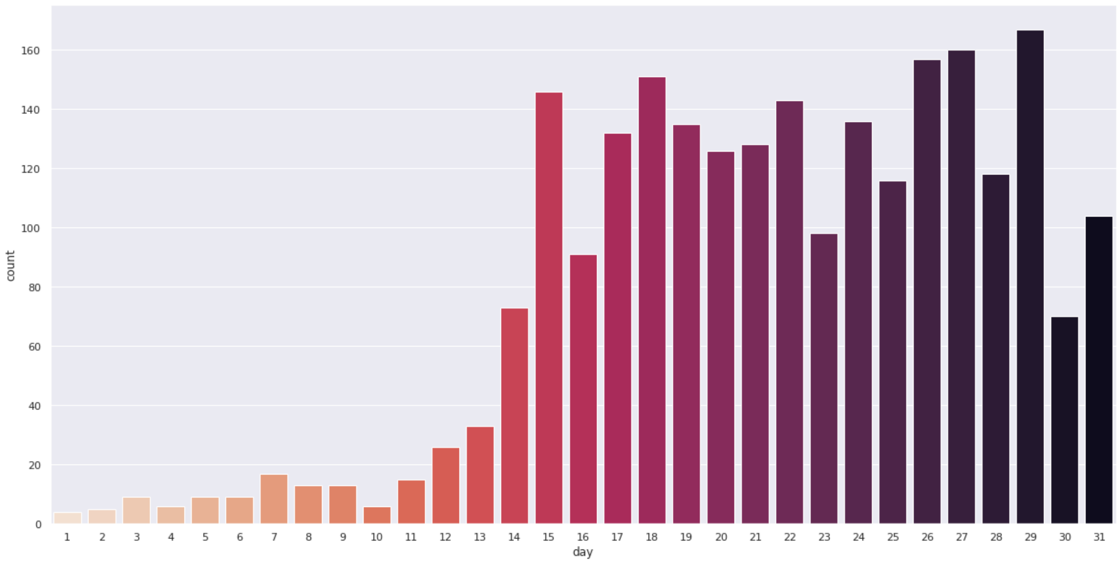
[14]:

```
#     x            order list    .
order2 = []
for i in range(1,32):

    order2.append(str(i))
```

```
order2
```

```
[14]: ['1',
       '2',
       '3',
       '4',
       '5',
       '6',
       '7',
       '8',
       '9',
       '10',
       '11',
       '12',
       '13',
       '14',
       '15',
       '16',
       '17',
       '18',
       '19',
       '20',
       '21',
       '22',
       '23',
       '24',
       '25',
       '26',
       '27',
       '28',
       '29',
       '30',
       '31']
```

```
[15]: # seaborn  countplot          .
      plt.figure(figsize=(20,10))
      sns.set(style="darkgrid")
      ax = sns.countplot(x="day", data=corona_del_col[corona_del_col['month'] ==␣
       ↪'8'], palette="rocket_r", order = order2)
```



1. 8 . (8 /31)

```
[16]: corona_del_col[corona_del_col['month'] == '8']['day'].count()/31
      corona_del_col[corona_del_col['month'] == '8']['day'].value_counts().mean()
```

```
[16]: 77.93548387096774
```

```
[17]: # 8 quiz_1 .
      # float .
      quiz_1 = corona_del_col[corona_del_col['month'] == '8']['day'].value_counts().
       ↪mean()
      quiz_1
```

```
[17]: 77.93548387096774
```

1.7.4 3.4.

oo .

```
[18]: corona_del_col[' ']
```

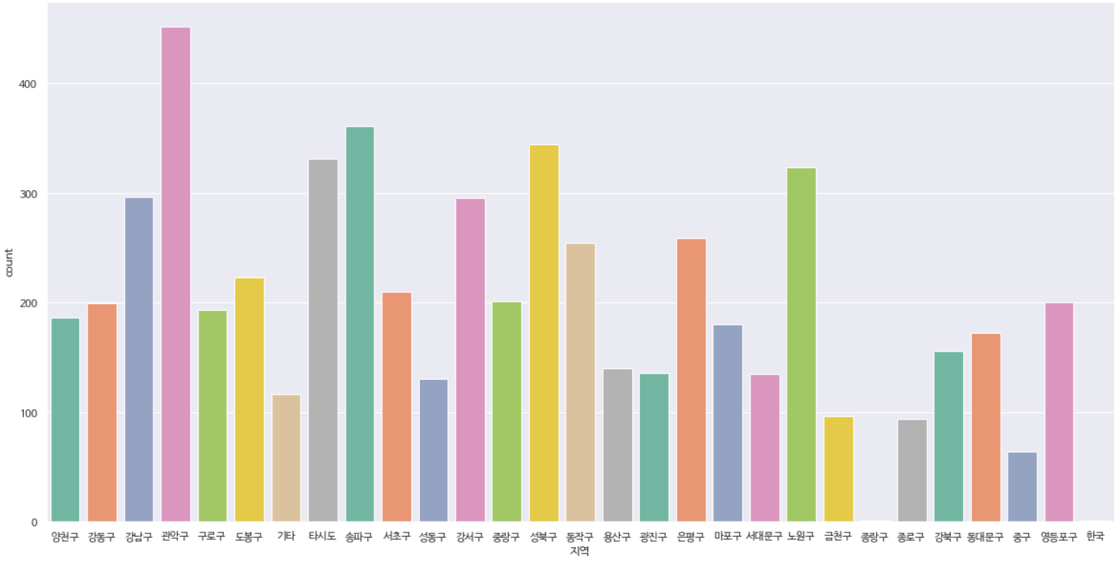
```
[18]: 0
      1
      2
      3
      4
           …
      5743
      5744
```

```
5745
5746
5747
Name:  , Length: 5748, dtype: object
```

.

[19]:
```python
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)
```

[20]:
```python
plt.figure(figsize=(20,10))
#                .
sns.set(font="NanumBarunGothic",
        rc={"axes.unicode_minus":False},
        style='darkgrid')
ax = sns.countplot(x="  ", data=corona_del_col, palette="Set2")
```

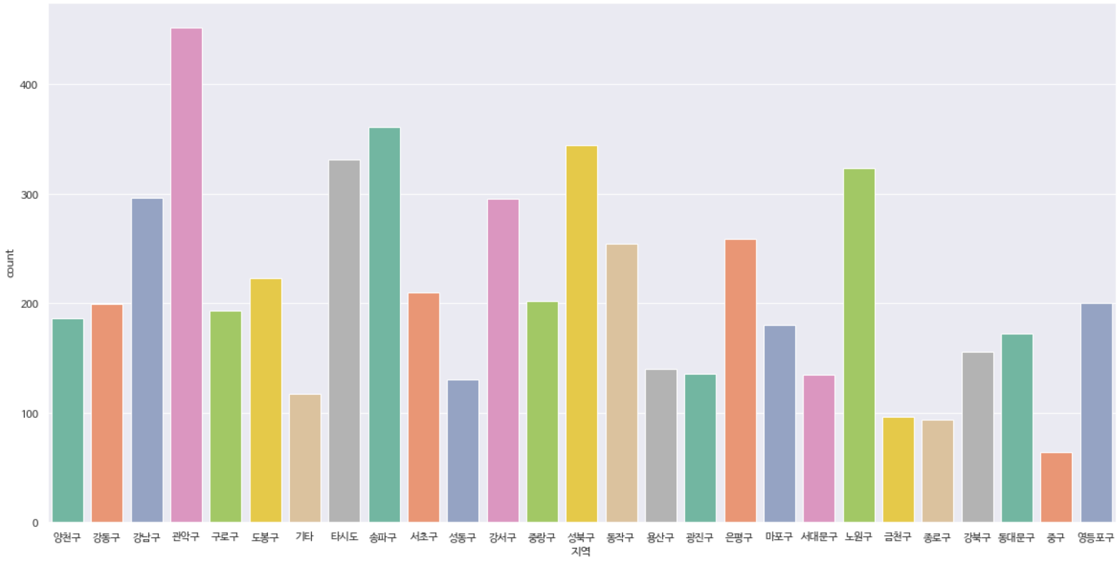


.

-> , -> .

[21]:
```python
# replace                  .
#              Dataframe     .
corona_out_region = corona_del_col.replace({'  ':'  ', ' ':' '})
```

[22]:
```
#                    .
plt.figure(figsize=(20,10))
sns.set(font="NanumBarunGothic",
        rc={"axes.unicode_minus":False},
        style='darkgrid')
ax = sns.countplot(x="  ", data=corona_out_region, palette="Set2")
```



1.7.5 3.5. 8

8   .

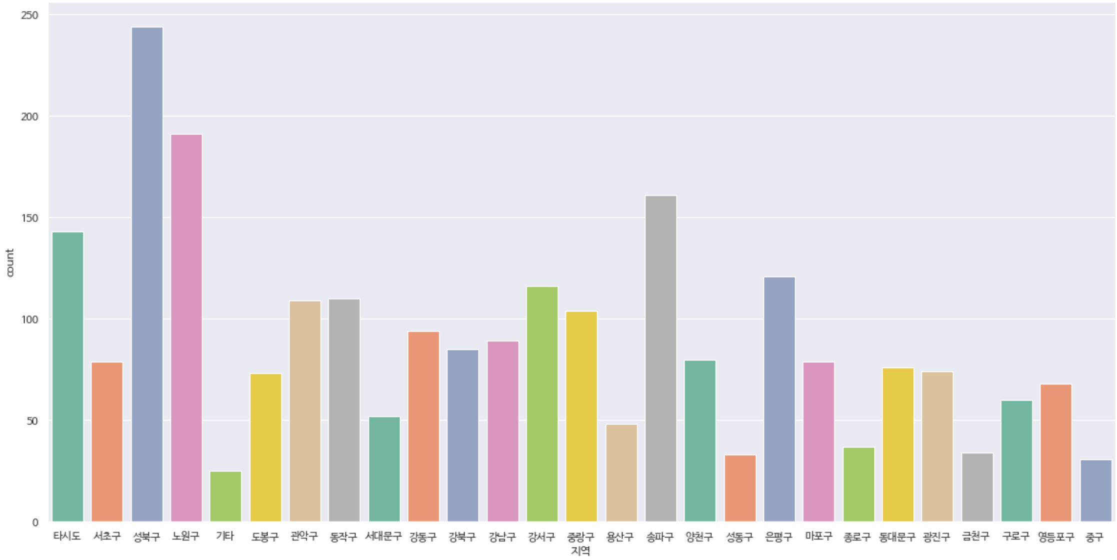
[23]:
```
#                                      .
corona_out_region[corona_del_col['month'] == '8']
```

[23]:
```
                                                                       \
1271  4477  8.31.  20132     NaN
1272  4476  8.25.  17968     NaN            NaN
1273  4475  8.26.  18821     NaN
1274  4474  8.27.  18818     NaN
1504  4244  8.30.  20116     NaN
…     …     …      …   …   …             …  …            …
4140  1608   8.2.  14382
4141  1607   8.1.  14364     NaN
4142  1606   8.1.  14362     NaN
4144  1604   8.1.  14366     NaN
4145  1603   8.1.  14365     NaN

                                        month day
1271  2020-10-22 10:58  2020-10-23 9:00    Y     8  31
```

```
1272  2020-10-22 10:58  2020-10-23 9:00    Y     8  25
1273  2020-10-22 10:58  2020-10-23 9:00    Y     8  26
1274  2020-10-22 10:58  2020-10-23 9:00    Y     8  27
1504  2020-10-22 10:58  2020-10-23 9:00    Y     8  30
…                …               …   …    …   ..
4140  2020-10-22 10:58  2020-10-23 9:00    Y     8   2
4141  2020-10-22 10:58  2020-10-23 9:00    Y     8   1
4142  2020-10-22 10:58  2020-10-23 9:00    Y     8   1
4144  2020-10-22 10:58  2020-10-23 9:00    Y     8   1
4145  2020-10-22 10:58  2020-10-23 9:00    Y     8   1

[2416 rows x 13 columns]
```

[24]:

```
#        .
plt.figure(figsize=(20,10))
sns.set(font="NanumBarunGothic",
        rc={"axes.unicode_minus":False},
        style='darkgrid')
ax = sns.countplot(x=" ", data=corona_out_region[corona_del_col['month'] ==␣
 ↪'8'], palette="Set2")
```



1.7.6 3.6.

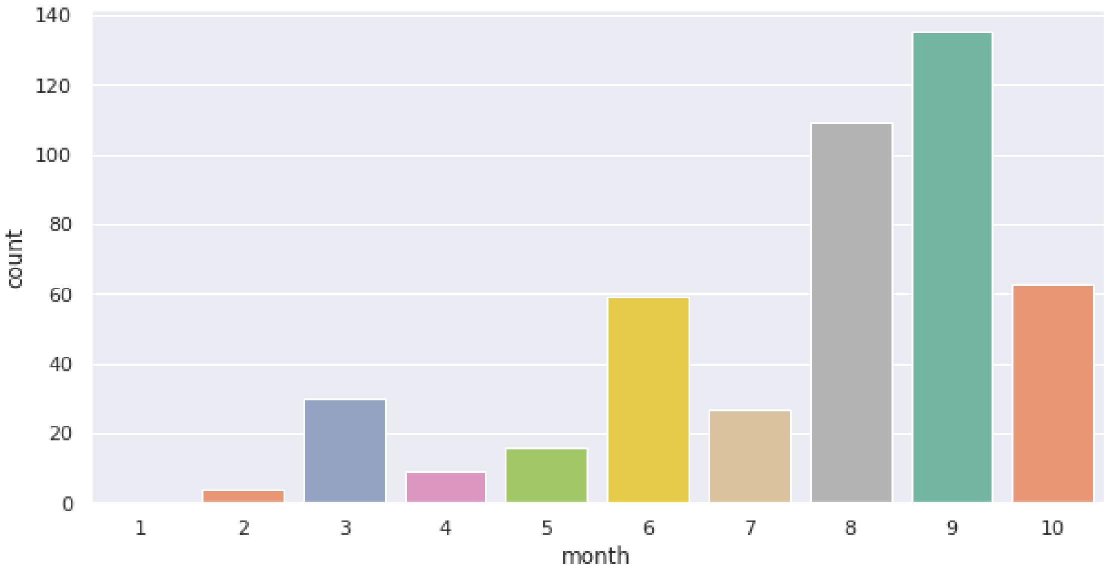
.

[25]:

```
#   column     series            .
corona_out_region['month'][corona_out_region['  '] == '  ']
```

```
[25]: 3       10
      4       10
      6       10
      7       10
      8       10
              ..
      5630     3
      5661     2
      5674     2
      5695     2
      5711     2
      Name: month, Length: 452, dtype: object
```

```
[26]: #       .
plt.figure(figsize=(10,5))
sns.set(style="darkgrid")
ax = sns.countplot(x="month", data=corona_out_region[corona_out_region['  '] ==␣
 ↪'  '], palette="Set2", order = order)
```



1.7.7 3.7.

folium .

```
[27]: #           folium  import   .
import folium

# Map              .
map_osm = folium.Map(location=[37.529622, 126.984307], zoom_start=11)
```

```
map_osm
```

[27]: `<folium.folium.Map at 0x7f5cbbc0da30>`

.

.

: https://data.seoul.go.kr/dataList/OA-11677/S/1/datasetView.do

[28]:
```
# CRS     .
CRS=pd.read_csv("./data/           ( _ WGS1984).csv")
```

[29]:
```
# Dataframe      .
CRS
```

[29]:
```
               _              _    ESRI_PK
0    1  11320            Dobong-gu       0  37.665861  127.031767
1    2  11380         Eunpyeong-gu       1  37.617612  126.922700
2    3  11230        Dongdaemun-gu      2  37.583801  127.050700
3    4  11590           Dongjak-gu       3  37.496504  126.944307
4    5  11545         Geumcheon-gu       4  37.460097  126.900155
5    6  11530              Guro-gu       5  37.495486  126.858121
6    7  11110            Jongno-gu       6  37.599100  126.986149
7    8  11305           Gangbuk-gu       7  37.646995  127.014716
8    9  11260          Jungnang-gu       8  37.595379  127.093967
9   10  11680           Gangnam-gu       9  37.495985  127.066409
10  11  11500           Gangseo-gu      10  37.565762  126.822656
11  12  11140              Jung-gu      11  37.557945  126.994190
12  13  11740          Gangdong-gu      12  37.549208  127.146482
13  14  11215          Gwangjin-gu      13  37.548144  127.085753
14  15  11440              Mapo-gu      14  37.562291  126.908780
15  16  11650            Seocho-gu      16  37.476953  127.037810
16  17  11290          Seongbuk-gu      17  37.606991  127.023218
17  18  11350             Nowon-gu      18  37.655264  127.077120
18  19  11710            Songpa-gu      19  37.504853  127.114482
19  20  11410         Seodaemun-gu     21  37.582037  126.935666
20  21  11470         Yangcheon-gu      22  37.527062  126.856153
21  22  11560       Yeongdeungpo-gu    23  37.520641  126.913924
22  23  11620            Gwanak-gu      15  37.465399  126.943807
23  24  11200          Seongdong-gu      20  37.550675  127.040962
24  25  11170            Yongsan-gu      24  37.531101  126.981074
```

.

[30]:
```
CRS[CRS['   _  '] == '  ']
```

[30]:
```
                _      _    ESRI_PK
11   12   11140         Jung-gu        11   37.557945   126.99419
```

for .

[31]:
```
# corona_out_region     'oo '    `  `,  ` `            .
#         ,               corona_seoul    .
corona_seoul = corona_out_region.drop(corona_out_region[corona_out_region['  ']␣
↪== '  '].index)
corona_seoul = corona_seoul.drop(corona_out_region[corona_out_region['  '] ==␣
↪'  '].index)

#                 .
map_osm = folium.Map(location=[37.557945, 126.99419], zoom_start=11)

#      set       25          .
for region in set(corona_seoul['  ']):

    #           count     .
    count = len(corona_seoul[corona_seoul['  '] == region])
    #          CRS     .
    CRS_region = CRS[CRS['   _  '] == region]

    # CircleMarker                 .
    marker = folium.CircleMarker([CRS_region['  '], CRS_region['  ']], #
                                 radius=count/10 + 10,                 #
                                 color='#3186cc',            #
                                 fill_color='#3186cc',       #
                                 popup=' '.join((region, str(count), ' '))) #␣
↪

    #             .
    marker.add_to(map_osm)

map_osm
```

[31]: `<folium.folium.Map at 0x7f5cbbf20e80>`

2. 6 .

[32]:
```
top = corona_out_region[corona_del_col['month'] == '6']['  '].value_counts()
top.index[0]
```

[32]: `'  '`

[33]:
```
# 6                 quiz_2      .
#        .
quiz_2 = top.index[0]
```

```
quiz_2
```

[33]:

```
' '
```

---

[34]:

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

1.8

1 2 , quiz_1 ~ 2 csv .

.

[35]:

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

[36]:

```
answer=pd.read_csv('submission.csv')
answer.loc[0]['quiz_2']
```

[36]:

```
' '
```

[ ]:

```
#
await upload()
```

[ ]:

```
#
await check_score()
```

, .