데이터 시각화

07주차. 실제 데이터 실습하기(2)

서울사이버대학교

학습목표

1. 데이터 불러오기 및 탐색
2. 시각화 도구 불러오기
3. 확진일
4. 모든 날짜를 행에 만들어 주기
5. 누적 확진자 수 구하기
6. 확진월과 요일 구하기
7. 거주비별 확진자
8. 접촉력
9. 가장 많은 전파가 일어난 번호
10. 퇴원현황

데이터 불러오기 및 탐색

데이터 불러오기 및 탐색

출처: http://www.seoul.go.kr/coronaV/coronaStatus.do

1. 데이터 불러오기 및 탐색

- pandas, numpy 라이브러리 로드하기

```
import pandas as pd
Import numpy as np
```

- file_name 변수에 read_html 로 저장한 파일명을 지정해 줍니다.

```
file_name = f"seoul_covid19_9_07_.csv"
file_name
```

출처: http://www.seoul.go.kr/coronaV/coronaStatus.do

1. 데이터 불러오기 및 탐색

- Excel file 읽을 때 encoding 유의

```
df = pd.read_csv(file_name, encoding ="cp949")
df.shape
```

- "확진일", "환자번호"를 기준으로 역순으로 정렬합니다.

```
df = df.sort_values(by=["확진일","환자번호"], ascending=False)
```

```
df.head()
```

	연번	환자	확진일	거주지	여행력	접촉력	퇴원현황
0	5242	23696	9.28.	도봉구	-	도봉구 다나병원	NaN
1	5241	23697	9.28.	도봉구	-	도봉구 다나병원	NaN
2	5240	23698	9.28.	관악구	-	확진자 조사 중	NaN
3	5239	23685	9.28.	영등포구	-	확진자 조사 중	NaN
4	5238	23675	9.28.	성북구	-	기타 확진자 접촉	NaN

시각화 도구 불러오기

출처: http://www.seoul.go.kr/coronaV/coronaStatus.do

2. 시각화 도구 불러오기

pd.Series([1, 3, 5, 7, 9]).plot()

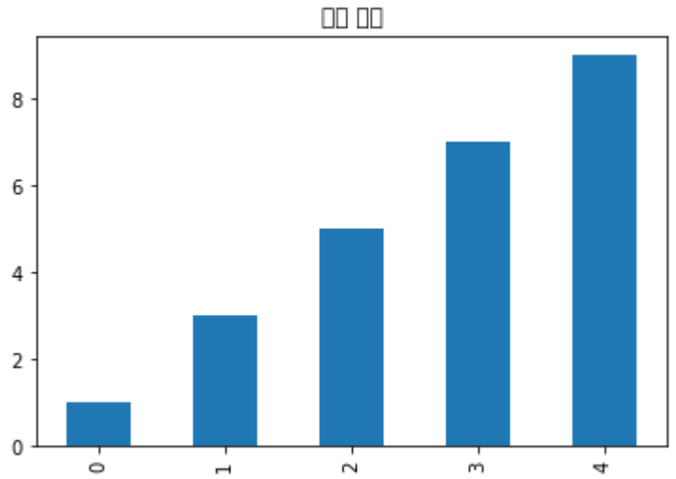
- 한글 폰트 에러 발생시 대처

```
# matplotlib.pyplot 을 통해 한글폰트를 설정합니다.
# plt.style.use 로 "fivethirtyeight" 스타일을 사용해 봅니다.

import matplotlib.pyplot as plt

plt.rc("font", family="Malgun Gothic") # window 일 때
#plt.rc("font", family="AppleGothic") # mac os 일 때
plt.rc("axes", unicode_minus=False) # minus 표시
plt.style.use("fivethirtyeight")
# plt.style.use("ggplot")

pd.Series([1, 3, 5, -7, 9]).plot.bar(title="한글 제목")
```



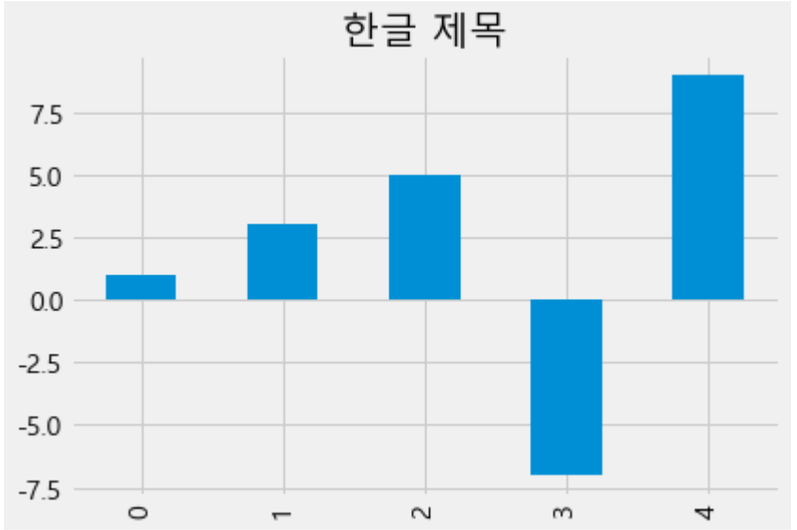
출처: http://www.seoul.go.kr/coronaV/coronaStatus.do

2. 시각화 도구 불러오기

- 시각화 선명하게 설정하기

retina 디스플레이가 지원되는 환경에서
시각화의 폰트가 좀 더 선명해 보입니다.

```
from IPython.display import set_matplotlib_formats
set_matplotlib_formats("retina") # retina display
```



확진일

출처: http://www.seoul.go.kr/coronaV/coronaStatus.do

3. 확진일

- 데이터 구조 파악

df["확진일"].head()

- 확진일 빈도수 파악

df["확진일"].value_counts()

- 데이터 타입 변경 (문자형태→ 날짜형태)

판다스의 to_datetime 을 사용해서 날짜 타입으로 변경할 수 있습니다.

연도가 없기 때문에 2020년을 날짜에 추가하고 "-" 문자로 날짜를 연결해 줍니다.

df["확진일자"] = pd.to_datetime("2020-"+ df["확진일"].str.replace("/", "-"))

df["확진일자"].head()

```
: df[["확진일", "확진일자"]].head()
```

:

	확진일	확진일자
0	9.28.	2020-09-28
1	9.28.	2020-09-28
2	9.28.	2020-09-28
3	9.28.	2020-09-28
4	9.28.	2020-09-28

출처: http://www.seoul.go.kr/coronaV/coronaStatus.do

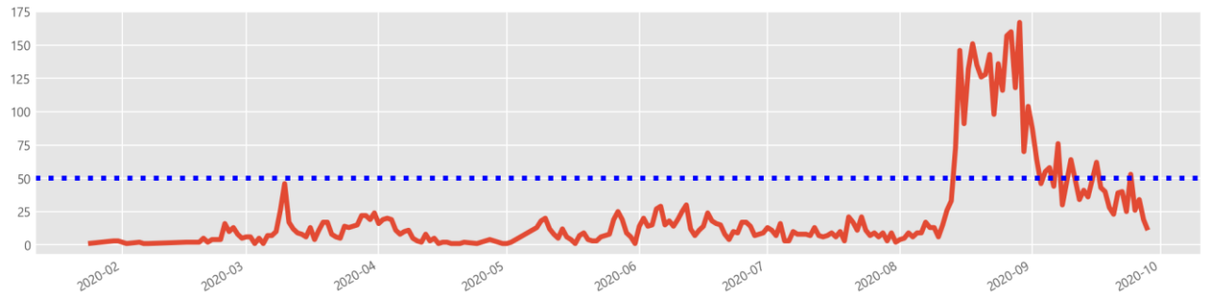
3. 확진일

- 선그래프 그리기 (참조: Visualization — pandas documentation)

```
df["확진일자"].value_counts().plot()

# 확진일자로 선그래프를 그립니다.
df["확진일자"].value_counts().sort_index().plot(figsize=(15, 4))
plt.axhline(50, color="red", linestyle=":")

# 뒤에서 문자 5개 가지고 오기
df["확진일자"].astype(str).map(lambda x : x[-5:]).head()
```



출처: http://www.seoul.go.kr/coronaV/coronaStatus.do

3. 확진일

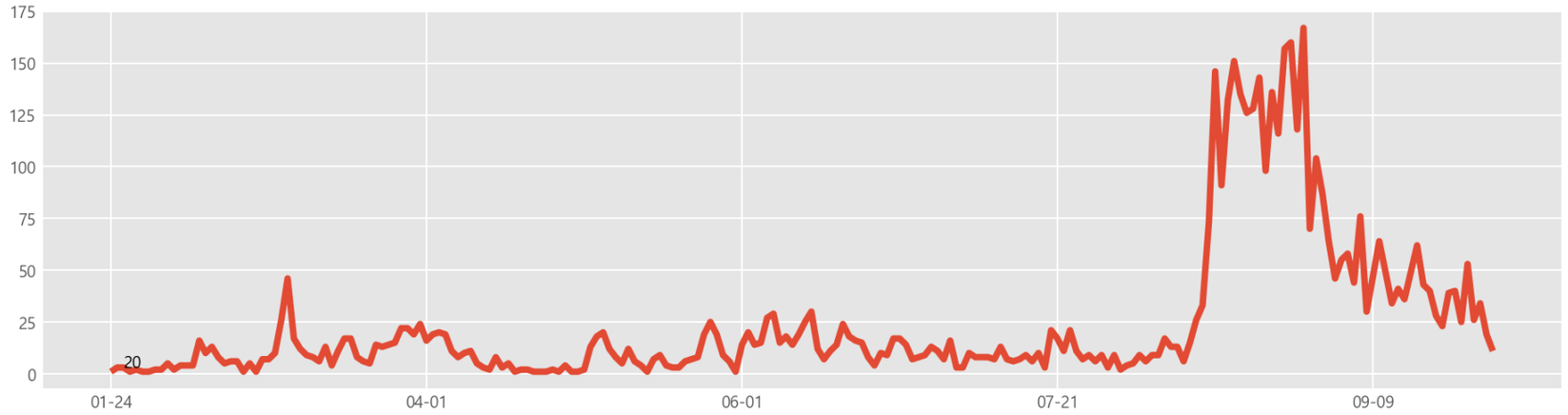
- 슬라이싱을 사용해 "월일" 컬럼을 만듭니다.

```
df["월일"] = df["확진일자"].astype(str).map(lambda x : x[-5:])
day_count = df["월일"].value_counts().sort_index()
day_count
```

```
01-24      1
01-30      3
01-31      3
02-02      1
02-05      2
          ..
09-24     53
09-25     26
09-26     34
09-27     19
09-28     11
Name: 월일, Length: 220, dtype: int64
```

- 선 그래프에 text 삽입

```
g = day_count.plot(figsize=(15,4))
g.text(x=2, y=3, s=20)
```



출처: http://www.seoul.go.kr/coronaV/coronaStatus.do

3. 확진일

- 환자발생이 50명 이상(사회적 거리두기 2단계)인 것만 추출

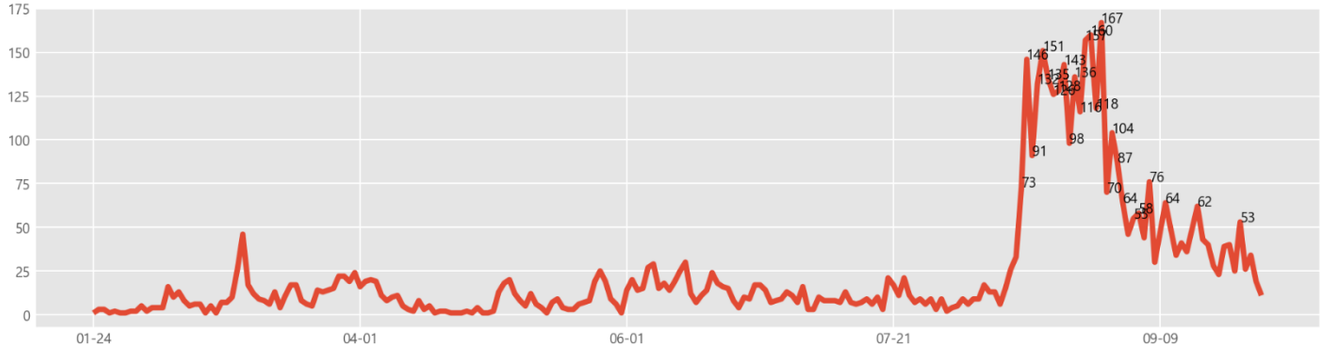
- for 반복문 활용

```
for i in range(len(day_count)):
    case_count = day_count.iloc[i]
    if case_count > 50:
        print(i, case_count)
```

```
174 73
175 146
176 91
177 132
178 151
179 135
180 126
181 128
182 143
183 98
184 135
185 115
186 157
187 159
188 118
189 167
190 70
```

- text (g 변수) 그래프 삽입

```
g = day_count.plot(figsize=(15, 4))
for i in range(len(day_count)):
    case_count = day_count.iloc[i]
    if case_count > 50:
        g.text(x=i, y=case_count, s=case_count)
```



출처: http://www.seoul.go.kr/coronaV/coronaStatus.do

3. 확진일

- 확진자가 가장 많이 나온 날

- 요약통계량 describe()

```
day_count.describe()
```

- 확진자가 가장 많이 나온 날

```
day_count[day_count == day_count.max()]
```

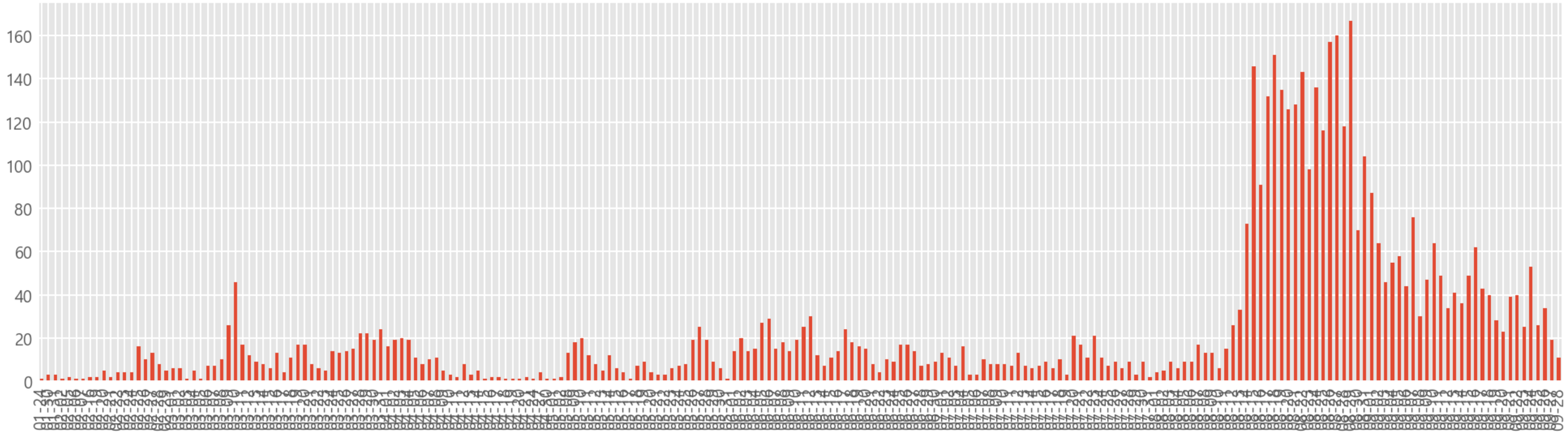
```
# 확진자가 가장 많았던 날의 발생이력
df[df["월일"] == "08-29"].head()
```

	연번	환자	확진일	거주지	여행력	접촉력	퇴원현황	확진일자	월	주	월일
516	3913	20100	8.29.	관악구	-	타시도 확진자 접촉	NaN	2020-08-29	8	35	08-29
551	3878	19716	8.29.	동작구	-	성북구 사랑제일교회 관련	NaN	2020-08-29	8	35	08-29
578	3851	19830	8.29.	구로구	-	영등포구 큰권능교회	NaN	2020-08-29	8	35	08-29
579	3850	19849	8.29.	노원구	-	노원구 빛가온교회 관련	NaN	2020-08-29	8	35	08-29
580	3849	19898	8.29.	금천구	-	타시도 확진자 접촉	NaN	2020-08-29	8	35	08-29

출처: http://www.seoul.go.kr/coronaV/coronaStatus.do

4. 막대그래프 그리기

```
day_count.plot.bar(figsize=(15, 4))
```



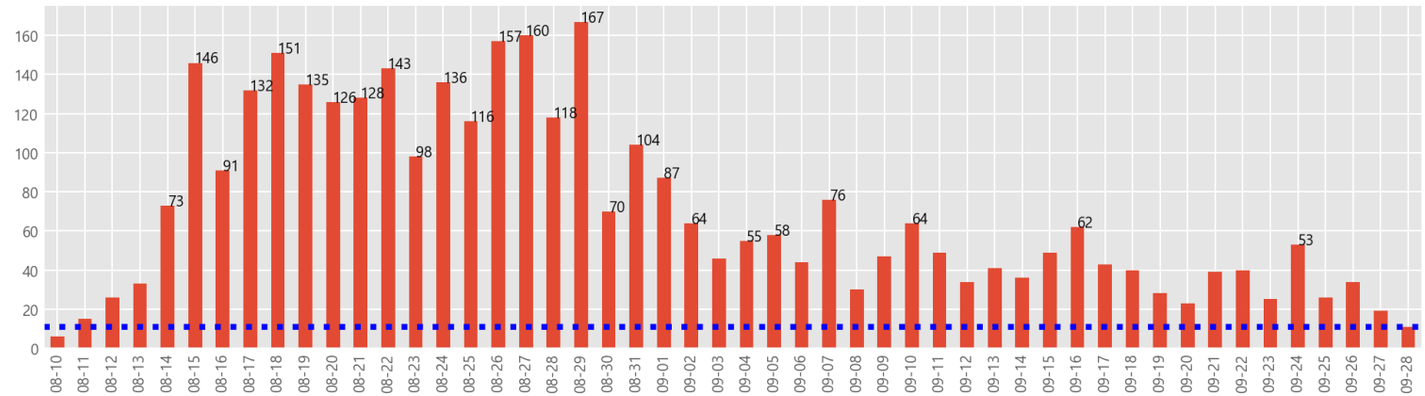
출처: http://www.seoul.go.kr/coronaV/coronaStatus.do

5. 슬라이싱으로 나누어 그리기

```
day_count[-50:].plot.bar(figsize=(15,4))
```

```
g = day_count[-50:].plot.bar(figsize=(15, 4))
g.axhline(day_count.median(), linestyle=":", color="red")
```

```
for i in range(50):
    case_count = day_count[-50:].iloc[i]
    if case_count >= 50:
        g.text(x=i-0.5, y=case_count, s=case_count)
# x=i-0.5 : text를 가운데로 옮기기 위해 위치 조정
```



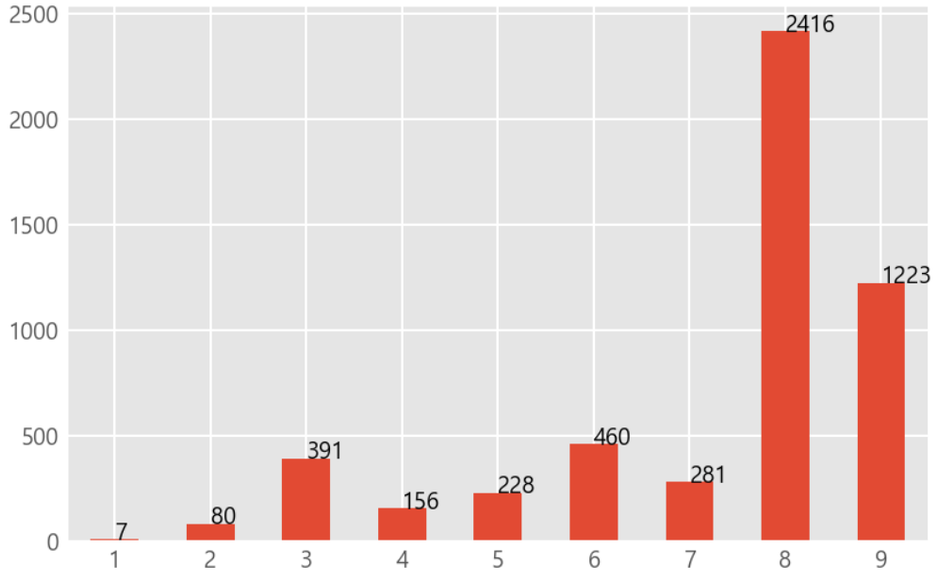
출처: http://www.seoul.go.kr/coronaV/coronaStatus.do

6. 월별 확진자 수 그리기

월별 확진자수에 대한 빈도수를 구해서 시각화

```
month_case = df["월"].value_counts().sort_index() # 월별로 빈도수 표현
g = month_case.plot.bar(rot=0) # rot=0 : rotation 을 0으로 변경

for i in range(len(month_case)):
    g.text(x=i-0.2, y=month_case.iloc[i]+10, s=month_case.iloc[i])
```

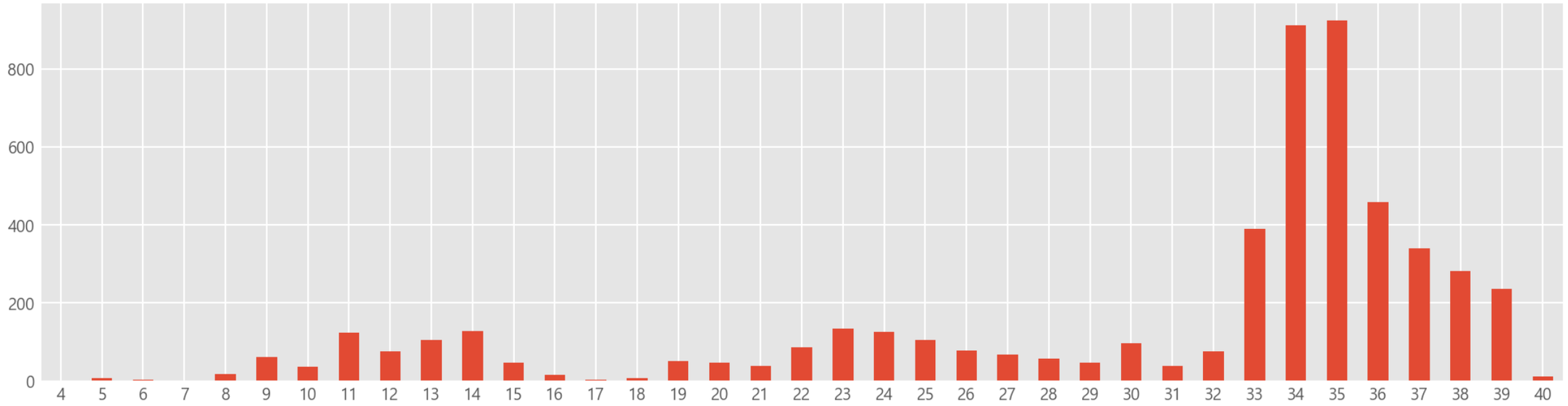


출처: http://www.seoul.go.kr/coronaV/coronaStatus.do

7. 주 단위 확진자수 그리기

주 별 확진자수에 대한 빈도수를 구해서 시각화

```
weekly_case = df["주"].value_counts().sort_index() # index값으로 정렬
weekly_case.plot.bar(figsize=(15, 4), rot=0)
```



출처: http://www.seoul.go.kr/coronaV/coronaStatus.do

8. 월-주 함께 그리기

groupby 서브모듈을 사용

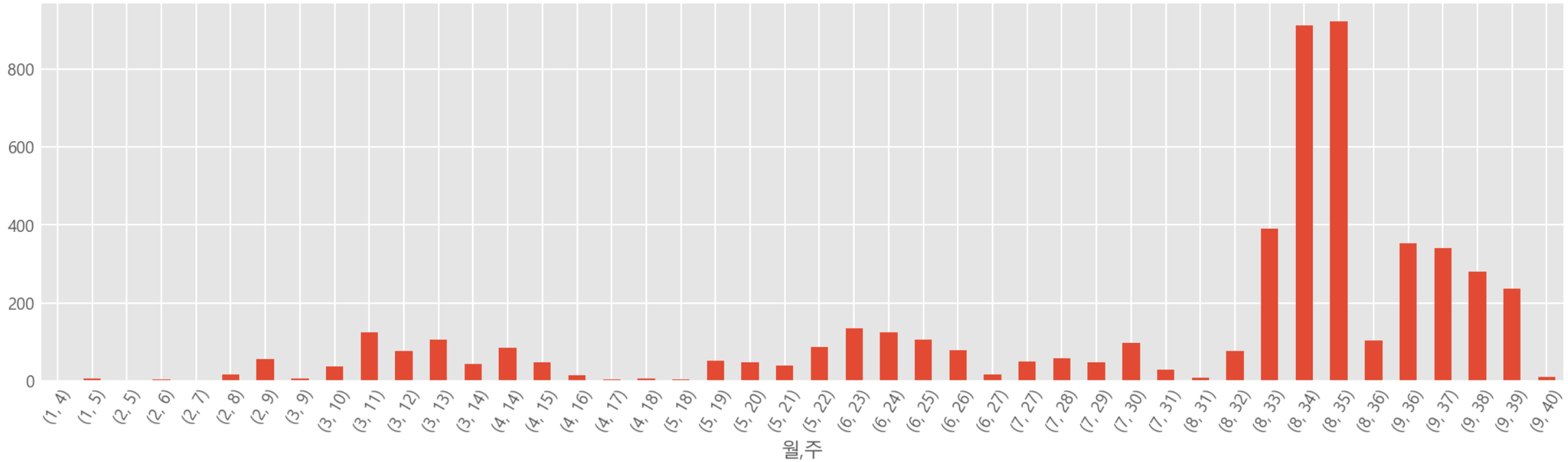
df.groupby(["월", "주"]).count()

월	주	연번	환자	확진일	거주지	여행력	접촉력	퇴원현황	확진일자	월일
1	4	1	1	1	1	1	1	1	1	1
	5	6	6	6	6	6	6	6	6	6
2	5	1	1	1	1	1	1	1	1	1
	6	4	4	4	4	4	4	4	4	4
	7	2	2	2	2	2	2	2	2	2
	8	17	17	17	17	17	17	17	17	17
	9	56	56	56	56	56	56	56	56	56
3	9	6	6	6	6	6	6	6	6	6
	10	37	37	37	37	37	37	37	37	37
	11	124	124	124	124	124	124	124	124	124
	12	76	76	76	76	76	76	76	76	76
	13	105	105	105	105	105	105	105	105	105
	14	43	43	43	43	43	43	43	43	43
	14	85	85	85	85	85	85	85	85	85
	15	47	47	47	47	47	47	47	47	47

출처: http://www.seoul.go.kr/coronaV/coronaStatus.do

8. 월-주 함께 그리기

```
# groupby 를 통해 "월", "주" 로 그룹화 하여 빈도수 계산
month_weekly_case = df.groupby(["월", "주"])["확진일"].count()
month_weekly_case.plot.bar(figsize=(15, 4), rot=60)
```



모든 날짜를 행에 만들어 주기

출처: http://www.seoul.go.kr/coronaV/coronaStatus.do

1. 확진자 없는 날 데이터 생성

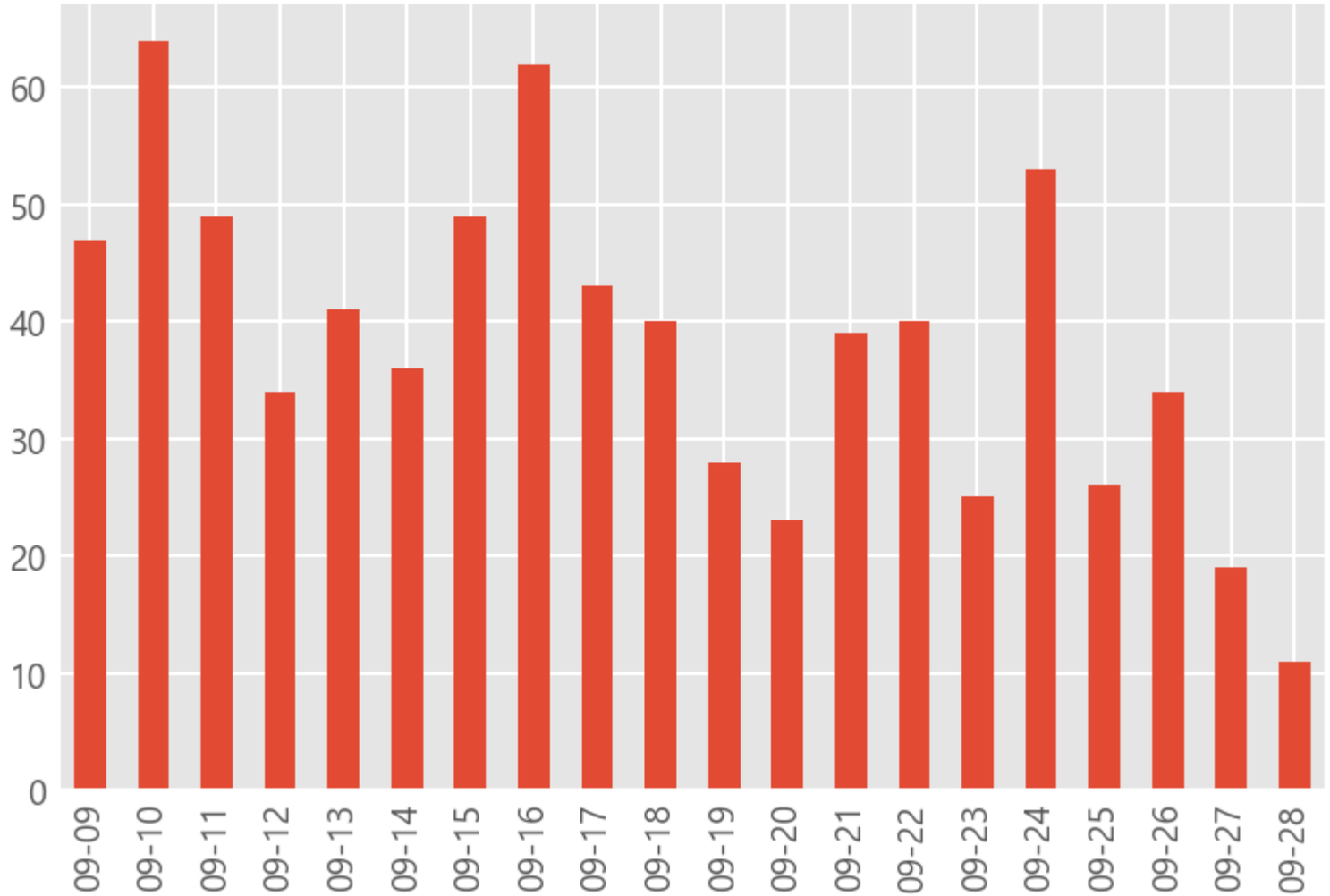
```
# 최근 20일의 일자별 환자수 확인
day_count[-20:].plot.bar()
```

```
 # 확진일자만 가지고 오고 싶을 때
first_day = df.iloc[-1, 7]
```

```
Timestamp('2020-01-24 00:00:00')
```

```
last_day = df.iloc[0, 7]
```

```
Timestamp('2020-09-07 00:00:00')
```



출처: http://www.seoul.go.kr/coronaV/coronaStatus.do

1. 확진자 없는 날 데이터 생성

pd.date_range 를 통해 시작하는 날짜부터 끝나는 날짜까지의
DatetimeIndex 를 만들고 days 라는 변수에 저장

```
days = pd.date_range(first_day, last_day)
days
```

```
DatetimeIndex(['2020-01-24', '2020-01-25', '2020-01-26', '2020-01-27',
               '2020-01-28', '2020-01-29', '2020-01-30', '2020-01-31',
               '2020-02-01', '2020-02-02',
               ...
               '2020-09-19', '2020-09-20', '2020-09-21', '2020-09-22',
               '2020-09-23', '2020-09-24', '2020-09-25', '2020-09-26',
               '2020-09-27', '2020-09-28'],
              dtype='datetime64[ns]', length=249, freq='D')
```

data frame 형태로 변환

```
pd.DataFrame({"확진일자":days})
```

출처: http://www.seoul.go.kr/coronaV/coronaStatus.do

1. 확진자 없는 날 데이터 생성

확진일자별로 빈도수 구하기

```
daily_case = df["확진일자"].value_counts()
daily_case.head()
```

```
2020-08-29    167
2020-08-27    159
2020-08-26    157
2020-08-18    151
2020-08-15    146
Name: 확진일자, dtype: int64
```

확진일자별로 빈도수 구한 내용을 데이터프레임으로 변환하기

```
df_daily_case = daily_case.to_frame()
df_daily_case.head()
```

	확진일자
2020-08-29	167
2020-08-27	159
2020-08-26	157
2020-08-18	151
2020-08-15	146

출처: http://www.seoul.go.kr/coronaV/coronaStatus.do

2. Merge 를 통해 전체 확진일자 만들기

```
# key 값을 "확진일자"로 왼쪽 테이블을 사용하고
   key 값을 daily_case.index (index 값을 기준)으로 병합(merge)

# 모든"확진일자" 별 확진수를 알고 싶기 때문에 how="left" 사용

all_day = df_days.merge(df_daily_case,
                        left_on="확진일자",
                        right_on=df_daily_case.index, how="left")
all_day.head(10)
```

	확진일자	확진수
0	2020-01-24	1.0
1	2020-01-25	NaN
2	2020-01-26	NaN
3	2020-01-27	NaN
4	2020-01-28	NaN
5	2020-01-29	NaN
6	2020-01-30	3.0
7	2020-01-31	3.0
8	2020-02-01	NaN
9	2020-02-02	1.0

누적 확진자 수 구하기

출처: http://www.seoul.go.kr/coronaV/coronaStatus.do

1. 결측치 0으로 채우기

확진수를 fillna를 통해 결측치를 0으로 채워주고 누적해서 더해줍니다.

all_day["확진수"].fillna(0)

```
0       1.0
1       0.0
2       0.0
3       0.0
4       0.0
       ...
244    53.0
245    26.0
246    34.0
247    19.0
248    11.0
Name: 확진수, Length: 249, dtype: float64
```

누적합

all_day["누적확진"] = all_day["확진수"].fillna(0).cumsum()

all_day

	확진일자	확진수	누적확진
0	2020-01-24	1	1
1	2020-01-25	0	1
2	2020-01-26	0	1
3	2020-01-27	0	1
4	2020-01-28	0	1
...	...	...	...
244	2020-09-24	53	5152
245	2020-09-25	26	5178
246	2020-09-26	34	5212
247	2020-09-27	19	5231
248	2020-09-28	11	5242

249 rows × 3 columns

출처: http://www.seoul.go.kr/coronaV/coronaStatus.do

2. 데이터 타입 변경

```
#  연도를 제외하고 월-일로 "일자" 컬럼 만들기
```

```
all_day["확진일자"].astype(str).map(lambda x : x).head()
# map(lamda x: x) : 익명함수 x 에 x 를 넣으면 자기 자신의 값 출력
```

```
0    2020-01-24
1    2020-01-25
2    2020-01-26
3    2020-01-27
4    2020-01-28
Name: 확진일자, dtype: object
```

```
all_day["일자"] = all_day["확진일자"].astype(str).map(lambda x : x[-5:])
all_day.head()
```

```
0    01-24
1    01-25
2    01-26
3    01-27
4    01-28
Name: 확진일자, dtype: object
```

출처: http://www.seoul.go.kr/coronaV/coronaStatus.do

2. 데이터 타입 변경

```
# "확진수", "누적확진" 컬럼을 갖는 데이터프레임을 만들기
cum_day = all_day[["일자", "확진수", "누적확진"]]
cum_day = cum_day.set_index("일자")
cum_day.head()
```

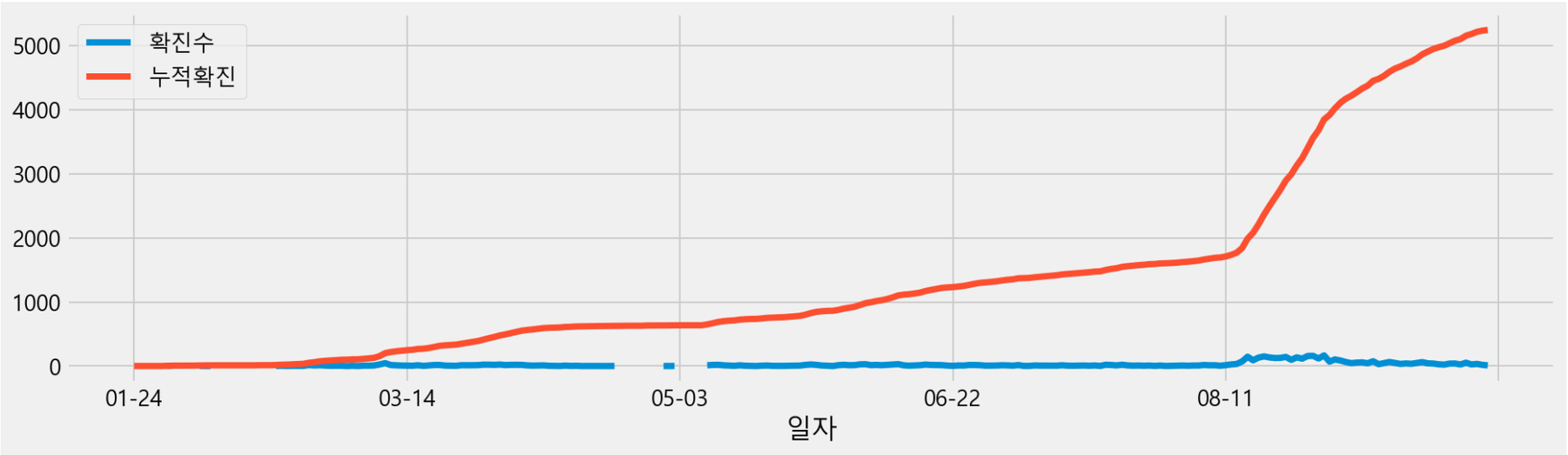
일자	확진수	누적확진
01-24	1.0	1.0
01-25	NaN	1.0
01-26	NaN	1.0
01-27	NaN	1.0
01-28	NaN	1.0

출처: http://www.seoul.go.kr/coronaV/coronaStatus.do

3. 데이터프레임으로 그리기

데이터프레임으로 확진수와 누적확진을 선그래프로 그립니다.

cum_day.plot(figsize=(15, 4))

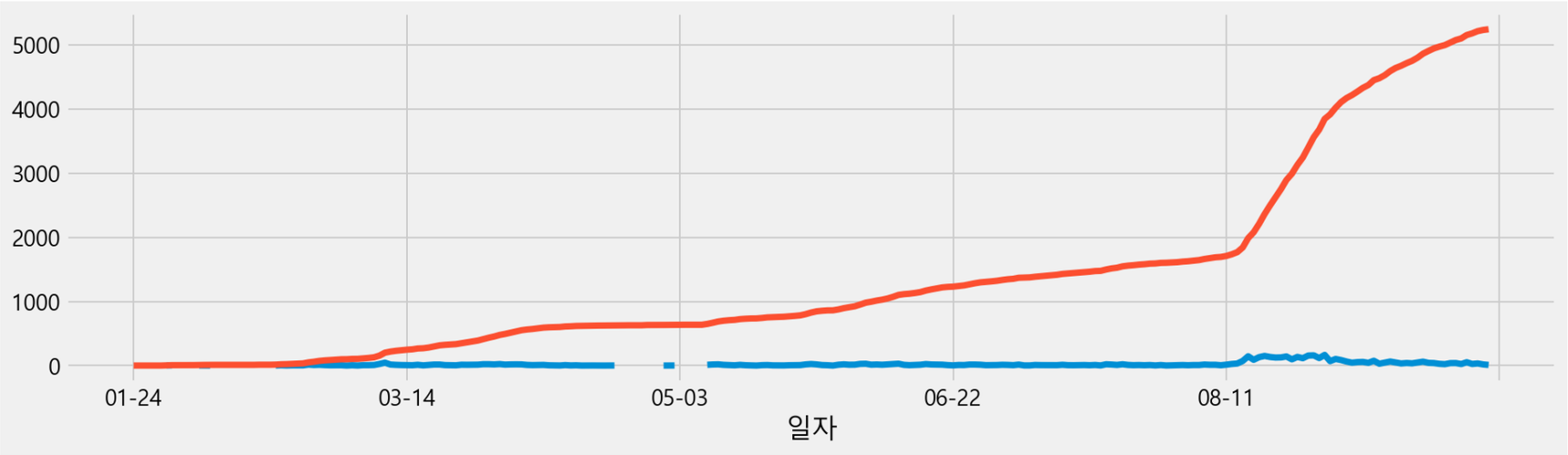


출처: http://www.seoul.go.kr/coronaV/coronaStatus.do

4. 시리즈로 그리기

시리즈로 2개의 그래프 그리기

```
cum_day["확진수"].plot()
cum_day["누적확진"].plot(figsize=(15, 4))
```

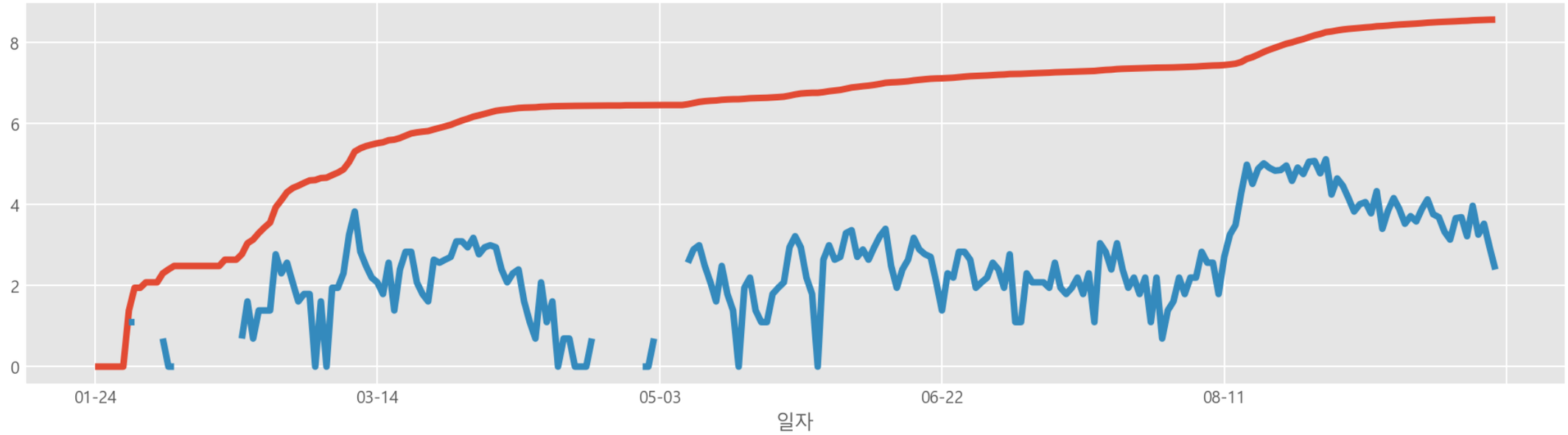


출처: http://www.seoul.go.kr/coronaV/coronaStatus.do

5. 로그스케일

* 차이가 너무 커서 그래프가 자세히 보이지 않을때 로그스케일로 표현하면 차이가 큰 값의 스케일을 조정

```
# 넘파이의 로그 모듈 활용(np.log)
np.log(cum_day["누적확진"]).plot(figsize=(15, 4))
np.log(cum_day["확진수"]).plot()
```



확진월과 요일 구하기

출처: http://www.seoul.go.kr/coronaV/coronaStatus.do

1. 요일별 확진 수

cum_day
확진요일 때 '월'요일이 0이 됨

```
all_day["확진월"] = all_day["확진일자"].dt.month
all_day["확진요일"] = all_day["확진일자"].dt.dayofweek
all_day.head()
```

	확진일자	확진수	누적확진	일자	확진월	확진요일
0	2020-01-24	1	1	01-24	1	4
1	2020-01-25	0	1	01-25	1	5
2	2020-01-26	0	1	01-26	1	6
3	2020-01-27	0	1	01-27	1	0
4	2020-01-28	0	1	01-28	1	1

출처: http://www.seoul.go.kr/coronaV/coronaStatus.do

1. 요일별 확진 수

#확진월, 확진요일 두 개의 시리즈를 부를 때,
  groupby 로 묶고, 확진수의 합을 구함

```
all_day_week = all_day.groupby(["확진월", "확진요일"])["확진수"].sum()
all_day_week
```

#행과 열을 바꾸어 주기 위해 unstack() 모듈 사용
#데이터 타입을 정수(integer)로 변경

```
all_day_week = all_day.groupby(["확진월", "확진요일"])["확진수"].sum()
all_day_week = all_day_week.unstack().astype(int)
all_day_week
```

```
확진월  확진요일
1     0        0.0
      1        0.0
      2        0.0
      3        3.0
      4        4.0
              ...
9     2      198.0
      3      206.0
      4      170.0
      5      154.0
      6      127.0
Name: 확진수, Length: 63, dtype: float64
```

확진요일	0	1	2	3	4	5	6
확진월							
1	0	0	0	3	4	0	0
2	4	16	14	19	11	9	7
3	69	89	46	44	48	45	50
4	16	17	28	27	26	22	20
5	27	36	34	34	32	36	29
6	55	67	66	75	90	70	37
7	37	42	42	58	34	40	28
8	387	288	327	328	336	473	277
9	162	206	198	206	170	154	127

출처: http://www.seoul.go.kr/coronaV/coronaStatus.do

1. 요일별 확진 수

숫자로 표현된 요일을 문자로 바꿔 주기 위해
  split 을 통해 문자를 리스트로 변경합니다.

```
dayofweek = "월 화 수 목 금 토 일"
dayofweek = dayofweek.split()
dayofweek
```

```
['월', '화', '수', '목', '금', '토', '일']
```

컬럼의 이름을 한글 요일명으로 변경해 줍니다.

```
all_day_week.columns = dayofweek
all_day_week
```

확진월	월	화	수	목	금	토	일
1	0	0	0	3	4	0	0
2	4	16	14	19	11	9	7
3	69	89	46	44	48	45	50
4	16	17	28	27	26	22	20
5	27	36	34	34	32	36	29
6	55	67	66	75	90	70	37
7	37	42	42	58	34	40	28
8	387	288	327	328	336	473	277
9	162	206	198	206	170	154	127

출처: http://www.seoul.go.kr/coronaV/coronaStatus.do

1. 요일별 확진 수

style.background_gradient 로 색상을 표현합니다.

all_day_week.style.background_gradient(cmap="Blues")

확진월	월	화	수	목	금	토	일
1	0	0	0	3	4	0	0
2	4	16	14	19	11	9	7
3	69	89	46	44	48	45	50
4	16	17	28	27	26	22	20
5	27	36	34	34	32	36	29
6	55	67	66	75	90	70	37
7	37	42	42	58	34	40	28
8	387	288	327	328	336	473	277
9	162	206	198	206	170	154	127

거주지별 확진자

출처: http://www.seoul.go.kr/coronaV/coronaStatus.do

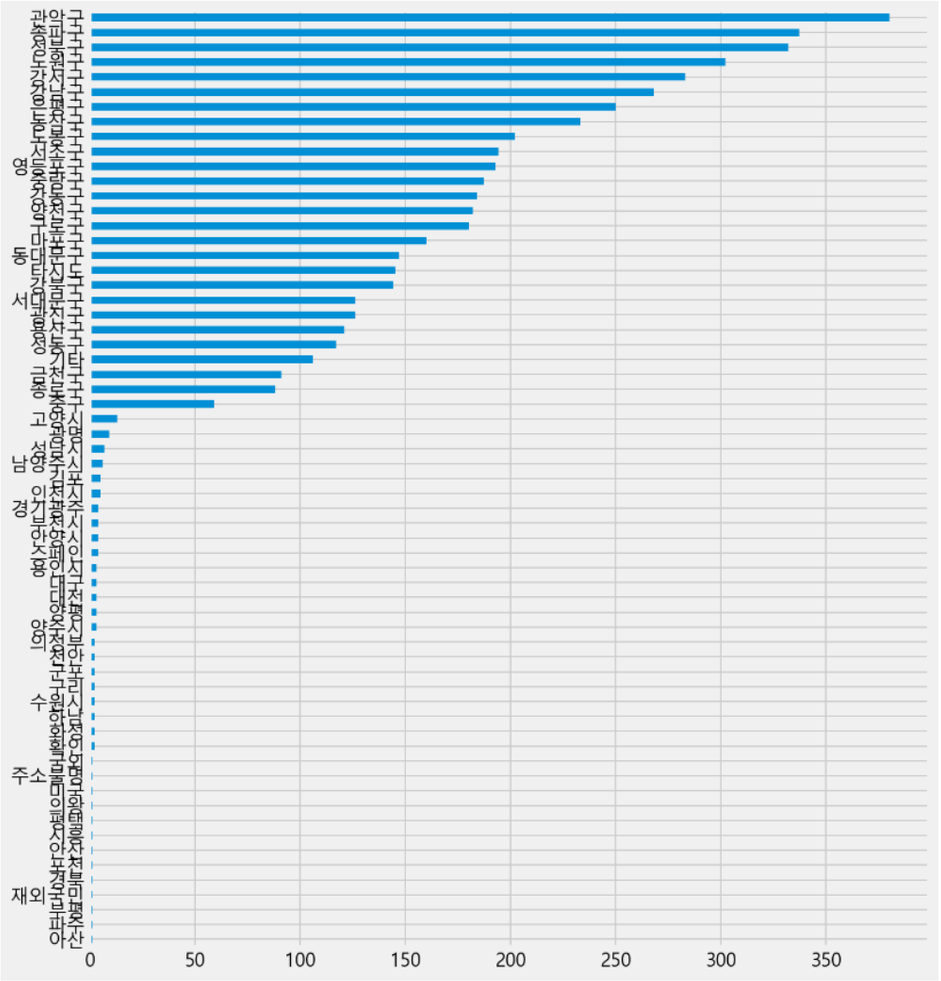
1. 거주지별 확진자

거주지(구별) 확진자의 빈도수를 구하고 시각화 합니다.

```
gu_count = df["거주지"].value_counts()
gu_count.head()
```

구별 확진자의 수를 시각화 합니다.

```
gu_count.sort_values().plot.barh(figsize=(10, 12))
```



출처: http://www.seoul.go.kr/coronaV/coronaStatus.do

1. 거주지별 확진자

서울시 거주자와 서울시 비거주자와 구분 작업
서울시는 25개의 구가 있음
서울에서 확진판정을 받은 데이터이기 때문에 거주지가 서울이 아닐 수도 있습니다.

```
gu_count[:25]
```

서울시 비거주자 데이터 제거 (drop() 사용)

```
gu_count[:27].drop(labels =['타시도','기타'])
```

```
gu = gu_count[:27].drop(labels =['타시도','기타']).index
gu
```

```
관악구      380
송파구      337
성북구      332
노원구      302
강서구      283
강남구      268
은평구      250
동작구      233
도봉구      202
서초구      194
영등포구     193
중랑구      187
강동구      184
양천구      182
구로구      180
마포구      160
동대문구     147
타시도      145
강북구      144
서대문구     126
광진구      126
용산구      121
성동구      117
기타       106
금천구       91
Name: 거주지, dtype: int64
```

출처: http://www.seoul.go.kr/coronaV/coronaStatus.do

1. 거주지별 확진자

거주지가 서울이 아닌 지역을 따로 추출합니다.

set(gu_count.index) - set(gu)

reset_index 활용, 데이터프레임으로 변환
컬럼명 변경

```
df_gu = gu_count.reset_index()
df_gu.columns = ["구", "확진수"]
df_gu.head()
```

	구	확진수
0	관악구	380
1	송파구	337
2	성북구	332
3	노원구	302
4	강서구	283

출처: http://www.seoul.go.kr/coronaV/coronaStatus.do

1. 거주지별 확진자

서울 거주 확진자만 보고 싶을 때

df_gu[df_gu["구"].isin(gu)]

서울 외 지역 확진자만 보고 싶다면

df_gu[~df_gu["구"].isin(gu)]

	구	확진수
0	관악구	380
1	송파구	337
2	성북구	332
3	노원구	302
4	강서구	283
5	강남구	268
6	은평구	250
7	동작구	233
8	도봉구	202
9	서초구	194

	구	확진수
17	타시도	145
23	기타	106
27	고양시	13
28	광명	9
29	성남시	7
30	남양주시	6
31	김포	5
32	인천시	5
33	스페인	4
34	안양시	4
35	부천시	4
36	경기광주	4
37	양평	3
38	양주시	3
39	대구	3
40	대전	3

출처: http://www.seoul.go.kr/coronaV/coronaStatus.do

1. 거주지별 확진자

지역 컬럼 생성

df.loc[df["거주지"].isin(gu), "지역"] = df["거주지"]

Df

df["지역"] = df["지역"].fillna("타지역")

df["지역"]

df["지역"].unique()

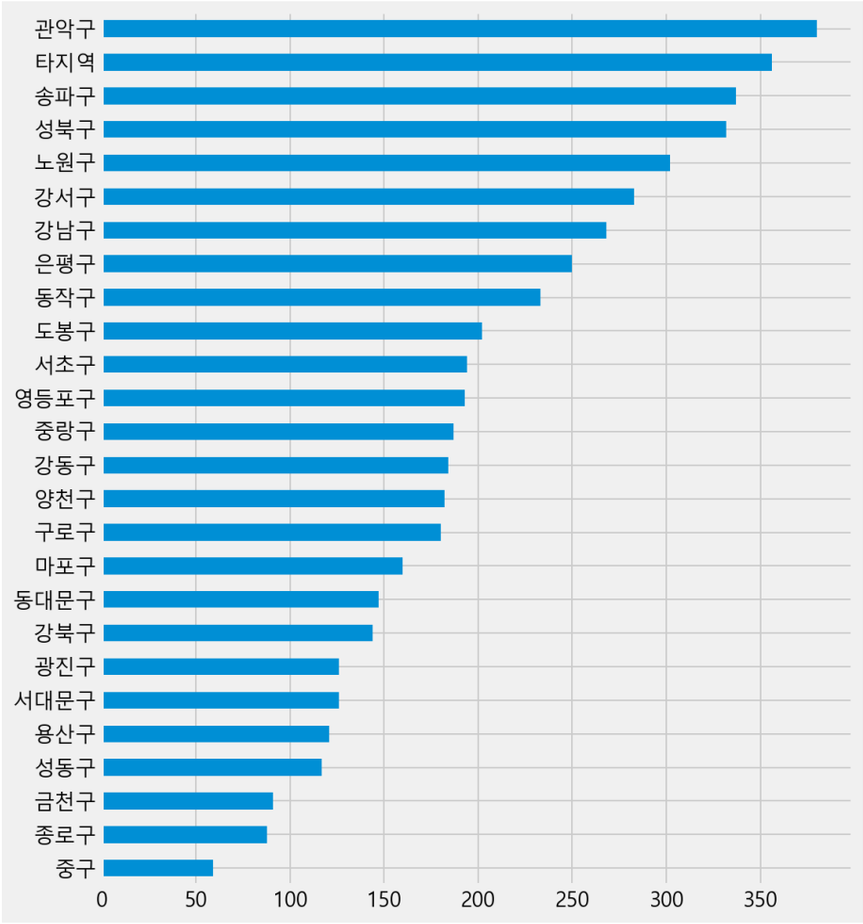
출처: http://www.seoul.go.kr/coronaV/coronaStatus.do

1. 거주지별 확진자

```
# 서울시 구와 타지역 반영해서 지역 컬럼으로 정함
gu_etc = df["지역"]

# "지역" 컬럼으로 확진자 빈도수를 구합니다.
gu_etc_count = df["지역"].value_counts()
gu_etc_count

# 빈도수를 막대그래프로 그리기(method chaining  활용)
gu_etc_count.sort_values().plot.barh(figsize=(8, 10))
```



접촉력

출처: http://www.seoul.go.kr/coronaV/coronaStatus.do

1. 접촉력 빈도수 시각화

접촉력 빈도수를 구합니다.

```
df["접촉력"].value_counts().head(20)
```

```
확인 중                872
기타 확진자 접촉         724
성북구 사랑제일교회 관련    641
해외 접촉 추정           390
타시도 확진자 접촉        214
이태원 클럽 관련          139
8.15서울도심집회          126
리치웨이 관련             119
구로구 콜센터 관련          60
노원구 빛가온교회 관련        46
양천구 운동시설 관련         43
성북구 체대입시 관련         43
요양시설 관련              43
확인중                   42
용인시 우리제일교회 관련      41
구로구 교회 관련           41
서대문구 세브란스병원         39
극단 산 관련              37
강남구 K보건산업           37
수도권 개척교회 관련         37
Name: 접촉력, dtype: int64
```

접촉력의 unique 값만 구합니다.

```
df["접촉력"].unique()
```

출처: http://www.seoul.go.kr/coronaV/coronaStatus.do

1. 접촉력 빈도수 시각화

이름이 유사한 변수 통합
- "확인" 이 들어가는 접촉력만 찾습니다.
df[df["접촉력"].str.contains("확인")]

- df.loc[df["접촉력"].str.contains("확인"), "접촉력"].unique()
  * loc[조건, column]

- '확인 중', '확인중' => "확인 중" 으로 변경합니다.
df.loc[df["접촉력"].str.contains("확인"), "접촉력"] = "확인 중"

- "확인" 이 들어가는 접촉력만 찾습니다.
df.loc[df["접촉력"].str.contains("확인"), "접촉력"].unique()

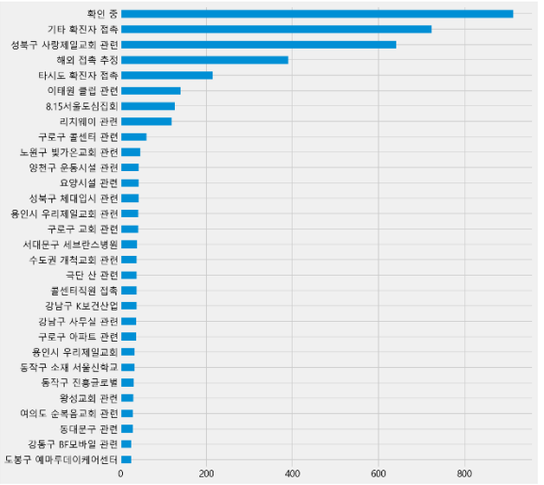
출처: http://www.seoul.go.kr/coronaV/coronaStatus.do

1. 접촉력 빈도수 시각화

```
# 상위에 랭크된 접촉 원인 파악
contact_count_top = contact_count.sort_values().tail(30)
contact_count_top.plot.barh(figsize=(10, 12))

# 상위 20개만 구합니다.
top_contact = contact_count_top.tail(20)

# 위에서 구한 top_contact 에 해당되는 데이터만 isin 으로 가져옵니다.
top_group = df[df["접촉력"].isin(top_contact.index)]
top_group.groupby(["접촉력", "월"])["연번"].count().unstack().fillna(0).astype(int)
```



월	2	3	4	5	6	7	8	9
접촉력								
8.15서울도심집회	0	0	0	0	0	0	113	13
강남구 K보건산업	0	0	0	0	0	0	0	37
구로구 교회 관련	0	35	6	0	0	0	0	0
구로구 콜센터 관련	0	60	0	0	0	0	0	0
극단 산 관련	0	0	0	0	0	0	36	1
기타 확진자 접촉	0	0	0	0	0	0	417	307
노원구 빛가온교회 관련	0	0	0	0	0	0	35	11
리치웨이 관련	0	0	0	0	119	0	0	0
서대문구 세브란스병원	0	0	0	0	0	0	0	39
성북구 사랑제일교회 관련	0	0	0	0	0	0	628	13
성북구 체대입시 관련	0	0	0	0	0	0	39	4

출처: http://www.seoul.go.kr/coronaV/coronaStatus.do

2. 이태원 클럽, 사랑제일교회 관련

6월에 이태원 클럽관련 확진자 찾기

Df[df[" 접촉력 "].str.contains(" 이태원 ") & (df[" 월 "] == 6)]

	연번	환자	확진일	거주지	여행력	접촉력	퇴원현황	확진일자	월	주	월일	지역
4262	980	11785	6.06.	성동구	-	이태원 클럽 관련	퇴원	2020-06-06	6	23	06-06	성동구
4277	965	11742	6.06.	은평구	-	이태원 클럽 관련	퇴원	2020-06-06	6	23	06-06	은평구
4286	956	11751	6.06.	성동구	-	이태원 클럽 관련	퇴원	2020-06-06	6	23	06-06	성동구
4293	949	11709	6.05.	은평구	-	이태원 클럽 관련	퇴원	2020-06-05	6	23	06-05	은평구
4308	934	11687	6.05.	성동구	-	이태원 클럽 관련	퇴원	2020-06-05	6	23	06-05	성동구
4366	876	11535	6.01.	강동구	-	이태원 클럽 관련	퇴원	2020-06-01	6	23	06-01	강동구

사랑제일교회도 9월 확진자 찾기

df[df["접촉력"].str.contains("사랑제일교회") & (df["월"] == 9)]

	연번	환자	확진일	거주지	여행력	접촉력	퇴원현황	확진일자	월	주	월일	지역
994	4248	20962	9.04.	성북구	-	성북구 사랑제일교회 관련	퇴원	2020-09-04	9	36	09-04	성북구
1048	4194	20833	9.03.	은평구	-	성북구 사랑제일교회 관련	NaN	2020-09-03	9	36	09-03	은평구
1055	4187	20885	9.03.	도봉구	-	성북구 사랑제일교회 관련	퇴원	2020-09-03	9	36	09-03	도봉구
1069	4173	20723	9.02.	강동구	-	성북구 사랑제일교회 관련	퇴원	2020-09-02	9	36	09-02	강동구
1080	4162	20724	9.02.	은평구	-	성북구 사랑제일교회 관련	NaN	2020-09-02	9	36	09-02	은평구
1115	4127	20596	9.02.	노원구	-	성북구 사랑제일교회 관련	퇴원	2020-09-02	9	36	09-02	노원구
1134	4108	20562	9.01.	은평구	-	성북구 사랑제일교회 관련	퇴원	2020-09-01	9	36	09-01	은평구
1142	4100	20577	9.02.	중구	-	성북구 사랑제일교회 관련	퇴원	2020-09-02	9	36	09-02	중구
1143	4099	20582	9.02.	중구	-	성북구 사랑제일교회 관련	퇴원	2020-09-02	9	36	09-02	중구

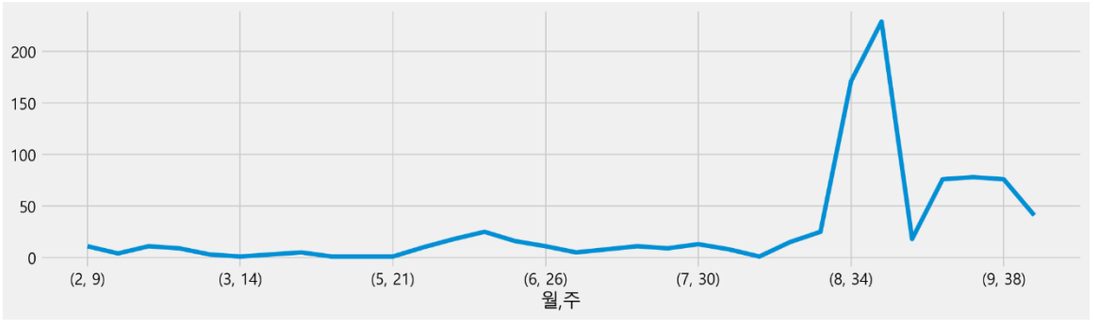
출처: http://www.seoul.go.kr/coronaV/coronaStatus.do

3. 감염경로 불명

```
# "접촉력" 이 "확인 중"인 데이터만 구합니다.
df_unknown = df[df["접촉력"] == "확인 중"]
```

	연번	환자	확진일	거주지	여행력	접촉력	퇴원현황	확진일자	월	주	월일	지역
32	5210	23607	9.26.	관악구	-	확인 중	NaN	2020-09-26	9	39	09-26	관악구
34	5208	23589	9.26.	강동구	-	확인 중	NaN	2020-09-26	9	39	09-26	강동구
50	5192	23574	9.26.	송파구	-	확인 중	NaN	2020-09-26	9	39	09-26	송파구
52	5190	23553	9.26.	타시도	-	확인 중	NaN	2020-09-26	9	39	09-26	타지역
57	5185	23562	9.26.	도봉구	-	확인 중	NaN	2020-09-26	9	39	09-26	도봉구

```
# 감염경로 불명이 어느정도인지 봅니다.
unknown_weekly_case = df_unknown.groupby(["월", "주"])["연번"].count()
unknown_weekly_case.plot(figsize=(15, 4))
```



출처: http://www.seoul.go.kr/coronaV/coronaStatus.do

3. 감염경로 불명

```
# 전체 확진수를 value_counts 로 구하고 데이터프레임 형태로 만듭니다.
all_weekly_case = df["주"].value_counts().to_frame()
all_weekly_case.columns = ["전체확진수"]
all_weekly_case.head()
```

	전체확진수
35	924
34	913
36	458
33	390
37	341

```
# 접촉원인을 모르는 데이터 확인
unknown_weekly_case = df_unknown["주"].value_counts().to_frame()
unknown_weekly_case.columns = ["불명확진수"]
unknown_weekly_case.head()
```

	불명확진수
35	229
34	171
36	94
37	78
38	76

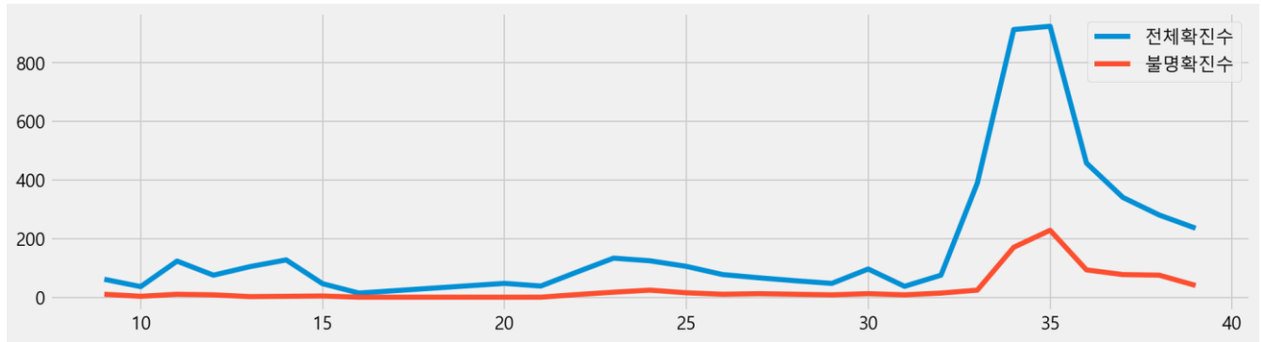
출처: http://www.seoul.go.kr/coronaV/coronaStatus.do

3. 감염경로 불명

```
# # all_weekly_case 와 unknown_weekly_case 를 비교해 봅니다. index 기준 merge 하기
unknown_case = all_weekly_case.merge(unknown_weekly_case, left_index=True, right_index=True)
unknown_case = unknown_case.sort_index()  # '주' 별로 정렬하기 위함
unknown_case.head()
```

	전체확진수	불명확진수
9	62	11
10	37	4
11	124	11
12	76	9
13	105	3

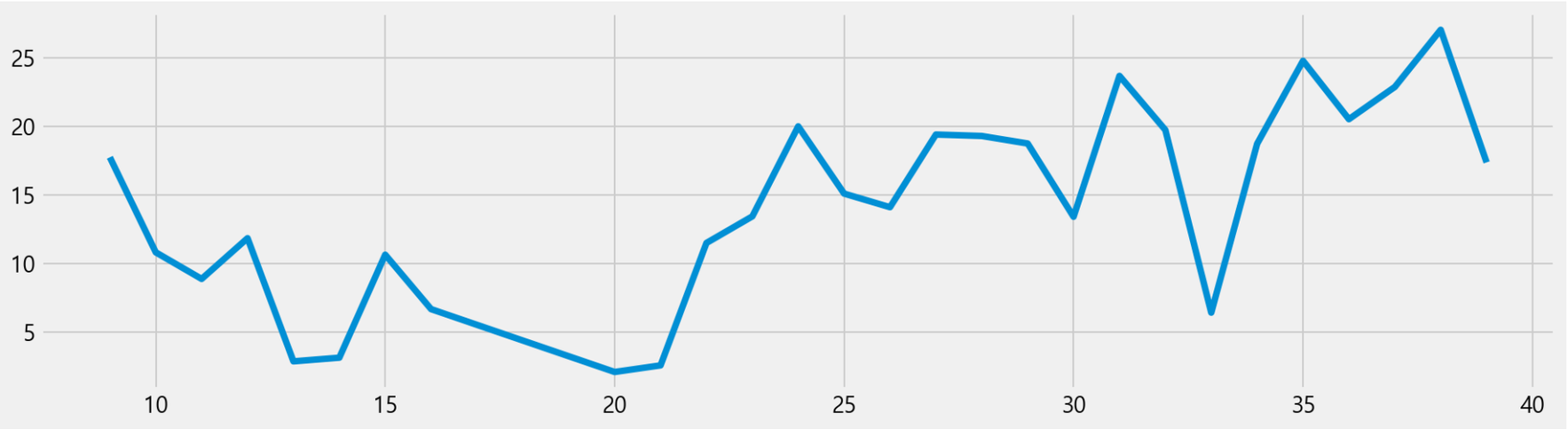
```
# 위에서 구한 결과를 시각화 합니다.
unknown_case.plot(figsize=(15, 4))
```



출처: http://www.seoul.go.kr/coronaV/coronaStatus.do

3. 감염경로 불명

```
# 감염경로 "확인 중"의 주별 비율
unknown_case["확인중비율"] = (unknown_case["불명확진수"] / unknown_case["전체확진수"]) * 100
unknown_case["확인중비율"].plot(figsize=(15, 4))
```



가장 많은 전파가 일어난 번호

출처: http://www.seoul.go.kr/coronaV/coronaStatus.do

1. 가장 많은 전파가 일어날 번호

•정규 표현식 - 위키백과, 우리 모두의 백과사전

•파이썬 공식문서 정규표현식 참고하기 :
  - https://docs.python.org/3.8/library/re.html#re.sub

•문자열 바꾸기 : re.sub("규칙", "패턴", "데이터")
  - https://docs.python.org/3.8/library/re.html#text-munging

•정규표현식 문자열 패턴
  - https://docs.python.org/3.8/howto/regex.html#matching-characters

•[ ] : 일치시킬 문자 세트의 패턴

•[가나다] : 가 or 나 or 다 중에 하나를 포함하고 있는지

•[가-힣] : 한글 가부터 힣까의 문자 중 하나를 포함하고 있는지

•[0-9] : 0~9까지의 숫자 중 하나를 포함하고 있는지

•[^0-9] : 숫자를 포함하고 있지 않음

•[^가-힣] : 한글이 포함되어 있지 않음

•[가-힣+] : 한글이 하나 이상 포함되는지

출처: http://www.seoul.go.kr/coronaV/coronaStatus.do

1. 가장 많은 전파가 일어날 번호

```
# 정규표현식 라이브러리를 부르기
import re

# 함수를 통해 숫자외의 문자를 제거하는 get_number 함수를 만듭니다.

def get_number(text):
    return re.sub("[^0-9]", "", text)

get_number("#7265 접촉(추정)")
```

'7265'

출처: http://www.seoul.go.kr/coronaV/coronaStatus.do

1. 가장 많은 전파가 일어날 번호

함수를 map을 통해 접촉번호를 구하기

```
df["접촉번호"] = df["접촉력"].map(get_number)
df["접촉번호"].value_counts()
```

```
contact = df["접촉번호"].value_counts().reset_index()
contact.head()
```

	index	접촉번호
0		5027
1	815	126
2	6	4
3	9734	4
4	8486	3

출처: http://www.seoul.go.kr/coronaV/coronaStatus.do

1. 가장 많은 전파가 일어날 번호

상위 10개의 접촉번호를 구해서 top_contact_no 변수에 할당하고 재사용

top_contact_no = df_contact["index"]

contact의 환자번호와 df의 접촉번호를 merge 하기

df[df["접촉번호"].isin(top_contact_no)]

	연번	환자	확진일	거주지	여행력	접촉력	퇴원현황	확진일자	월	주	월일	지역	접촉번호
568	4674	22192	9.13.	마포구	-	8.15서울도심집회	NaN	2020-09-13	9	37	09-13	마포구	815
726	4516	21669	9.09.	노원구	-	8.15서울도심집회	NaN	2020-09-09	9	37	09-09	노원구	815
807	4435	21494	9.07.	강서구	-	8.15서울도심집회	NaN	2020-09-07	9	37	09-07	강서구	815
865	4377	21314	9.07.	구로구	-	8.15서울도심집회	퇴원	2020-09-07	9	37	09-07	구로구	815
900	4342	21210	9.06.	노원구	-	8.15서울도심집회	NaN	2020-09-06	9	36	09-06	노원구	815
...	...	...	...	...	...	...	...	...	...	...	...	...	...
5232	10	21	2.5.	성북구	-	#6 접촉	퇴원	2020-02-05	2	6	02-05	성북구	6
5235	7	11	1.31.	종로구	-	#6 접촉	퇴원	2020-01-31	1	5	01-31	종로구	6
5236	6	10	1.31.	종로구	-	#6 접촉	퇴원	2020-01-31	1	5	01-31	종로구	6
5237	5	9	1.31.	성북구	-	#5 접촉	퇴원	2020-01-31	1	5	01-31	성북구	5
5239	3	6	1.30.	종로구	-	#3 접촉	퇴원	2020-01-30	1	5	01-30	종로구	3

215 rows × 13 columns

퇴원현황

출처: http://www.seoul.go.kr/coronaV/coronaStatus.do

1. 퇴원현황

#퇴원현황 확인

df["퇴원현황"].value_counts()

```
퇴원     4399
사망       55
Name: 퇴원현황, dtype: int64
```

퇴원/사망 비율 구하기

df["퇴원현황"].value_counts(normalize = True)

```
퇴원    0.987652
사망    0.012348
Name: 퇴원현황, dtype: float64
```

#"퇴원"이라는 글을 포함한 문자 추출

df["퇴원"] = df["퇴원현황"].str.contains("퇴원")

df["퇴원"] `

```
0        NaN
1        NaN
2        NaN
3        NaN
4        NaN
        ...
5237    True
5238    True
5239    True
5240    True
5241    True
Name: 퇴원, Length: 5242, dtype: object
```

출처: http://www.seoul.go.kr/coronaV/coronaStatus.do

1. 퇴원현황

퇴원 못한 환자 리스트 추출

df[(df["퇴원"] == False)]

	연번	환자	확진일	거주지	여행력	접촉력	퇴원현황	확진일자	월	주	월일	지역	접촉번호	퇴원	사망
783	4459	21525	9.08.	성북구	-	확인 중	사망	2020-09-08	9	37	09-08	성북구		False	True
991	4251	21075	9.04.	기타	-	송파구 소재병원	사망	2020-09-04	9	36	09-04	타지역		False	True
1034	4208	20939	9.04.	은평구	-	확인 중	사망	2020-09-04	9	36	09-04	은평구		False	True
1071	4171	20753	9.02.	성북구	-	성북구 요양시설	사망	2020-09-02	9	36	09-02	성북구		False	True
1156	4086	20532	9.01.	서대문구	-	서대문구 지인모임	사망	2020-09-01	9	36	09-01	서대문구		False	True
1303	3939	20073	8.30.	노원구	-	확인 중	사망	2020-08-30	8	35	08-30	노원구		False	True
1309	3933	20157	8.31.	성북구	-	성북구 요양시설	사망	2020-08-31	8	36	08-31	성북구		False	True
1340	3902	19974	8.30.	성동구	-	확인 중	사망	2020-08-30	8	35	08-30	성동구		False	True
1389	3853	19875	8.30.	기타	-	확인 중	사망	2020-08-30	8	35	08-30	타지역		False	True
1400	3842	19759	8.29.	강서구	-	확인 중	사망	2020-08-29	8	35	08-29	강서구		False	True
1446	3796	19825	8.30.	동작구	-	동작구 요양시설 관련	사망	2020-08-30	8	35	08-30	동작구		False	True
1612	3630	19593	8.27.	강서구	-	강동구 소재 병원	사망	2020-08-27	8	35	08-27	강서구		False	True
1697	3545	19167	8.28.	강남구	-	확인 중	사망	2020-08-28	8	35	08-28	강남구		False	True
1749	3493	18967	8.27.	관악구	-	강동구 소재 병원	사망	2020-08-27	8	35	08-27	관악구		False	True
1776	3466	18901	8.27.	은평구	-	동작구 진흥글로벌	사망	2020-08-27	8	35	08-27	은평구		False	True

형성평가

번호	문제	보기	정답	해설
1	다음 코드를 설명하시오. all_day["확진수"].fillna(0)	① 확진수를 fillna를 통해 전부 0으로 채워주고 누적해서 더해줍니다 ② 확진수를 fillna를 통해 결측치를 0으로 채워주고 누적해서 더해준다. ③ 확진수를 fillna를 통해 0으로 표현된 값을 제거하고 누적해서 더해줍니다 ④ 확진수를 fillna를 통해 0으로 나누어 주고 누적해서 더해줍니다	2	제시된 코드는 확진수를 fillna를 통해 결측치를 0으로 채워주고 누적해서 더해준다는 코드입니다.
2	"퇴원"이라는 글을 포함한 문자 추출하고자 사용한 코드로 맞는 것은?	① df["퇴원현황"].head("퇴원") ② df["퇴원현황"].value_counts() ③ df["퇴원현황"].str.contains("퇴원") ④ df["퇴원현황"].value_counts(normalize = True)	3	str.contains("문자")에 대한 설명입니다. string이 contain(담겨있다)는 코드로 문장에서 추출하고자 하는 문자를 추출할 때 사용됩니다.
3	정규표현식 라이브러리를 부를 때 사용하는 코드를 작성하시오		Import re	정규표현식을 부르는 코드인 re는 regular expression에 앞글자를 따서 만들어진 코드입니다.

학습정리

- CODE BOOK 인 관계로 목차처럼 페이지 수를 표기합니다.