



# [LAB-06] 3. 데이터 시각화 핵심 포인트



## #01. 준비작업



### 1. 라이브러리 참조

```
from hossam import load_data
from matplotlib import pyplot as plt
from matplotlib import font_manager as fm
import seaborn as sb
```



### 2. 데이터 불러오기

```
origin = load_data('traffic_acc')
origin
```

```
[94m[data] [0m https://data.hossam.kr/data/lab04/traffic_acc.xlsx
[94m[desc] [0m 2005년 1월부터 2018년 12월까지 월별 교통사고의 발생건수, 부상자
수, 사망자수 데이터(인덱스/메타데이터 없음, 출처: 공공데이터포털)
[91m[!] Cannot read metadata [0m
```

	년도	월	발생건수	사망자수	부상자수
0	2005	1	15494	504	25413
1	2005	2	13244	431	21635
2	2005	3	16580	477	25550
3	2005	4	17817	507	28131
4	2005	5	19085	571	29808
...	...	...	...	...	...
163	2018	8	18335	357	27749
164	2018	9	18371	348	27751
165	2018	10	19738	373	28836
166	2018	11	19029	298	28000
167	2018	12	18010	323	26463

168 rows × 5 columns

```
df = origin.drop('월', axis=1).groupby('년도').sum()
df
```

	발생건수	사망자수	부상자수
년도			
2005	214171	6376	342233
2006	213745	6327	340229
2007	211662	6166	335906
2008	215822	5870	338962
2009	231990	5838	361875
2010	226878	5505	352458
2011	221711	5229	341391
2012	223656	5392	344565
2013	215354	5092	328711
2014	223552	4762	337497
2015	232035	4621	350400
2016	220917	4292	331720
2017	216335	4185	322829
2018	217148	3781	323037



### 3. 그래프 초기화

```
my_dpi = 200
fpath = "./NotoSansKR-Regular.ttf"
fm.fontManager.addfont(fpath)
fprop = fm.FontProperties(fname=fpath)
fname = fprop.get_name()
    출
plt.rcParams['font.family'] = fname
plt.rcParams['font.size'] = 6
plt.rcParams['axes.unicode_minus'] = False # 그래프에 마이너스 깨짐 방지
# 이미지 선명도(100~300)
# 한글을 지원하는 폰트 파일의 경로
# 폰트의 글꼴을 시스템에 등록함
# 폰트의 속성을 읽어옴
# 읽어온 속성에서 폰트의 이름만 추출
# 그래프에 한글 폰트 적용
# 기본 폰트 크기
# 그래프에 마이너스 깨짐 방지
```



## #02. 시각화 기본 코드

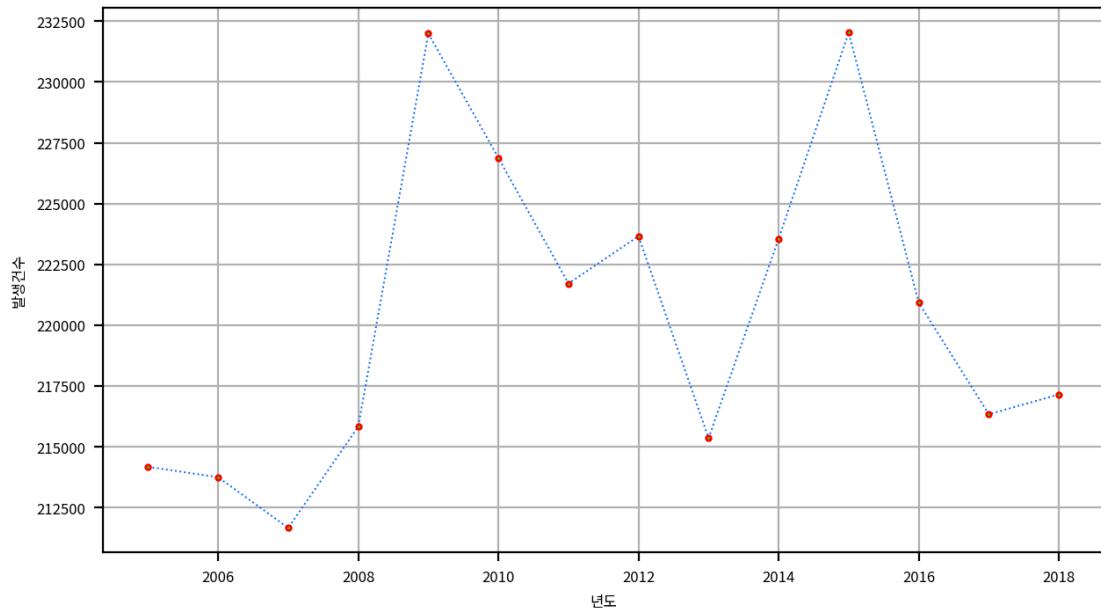
```
# 1) 그래프 초기화
width_px  = 1280                      # 그래프 가로 크기
height_px = 720                        # 그래프 세로 크기
rows      = 1                           # 그래프 행 수
cols      = 1                           # 그래프 열 수
figsize   = (width_px / my_dpi, height_px / my_dpi)
fig, ax  = plt.subplots(rows, cols, figsize=figsize, dpi=my_dpi)

# 2-1) BoxPlot 그리기
# sb.boxplot(data=df['발생건수'], orient="v")
# sb.boxplot(data=df['부상자수'], orient="v")
# sb.boxplot(data=df[['발생건수', '부상자수']], orient="v")

# 2-2) LinePlot
sb.lineplot(
    #df['발생건수'],
    #x=df.index, y=df['발생건수'],
    data=df, x=df.index, y='발생건수',
    color="#0066ff", linewidth=0.7, linestyle=':',
    marker="o", markersize=2,
    markerfacecolor="#00ff00",
    markeredgecolor="#ff0000", markeredgewidth=1)

# 3) 그래프 꾸미기
ax.grid(True)                         # 배경 격자 표시/숨김

# 4) 출력
plt.tight_layout()                    # 여백 제거
plt.show()                            # 그래프 화면 출력
plt.close()                           # 그래프 작업 종료
```



png

## #03. 다중 그래프

```

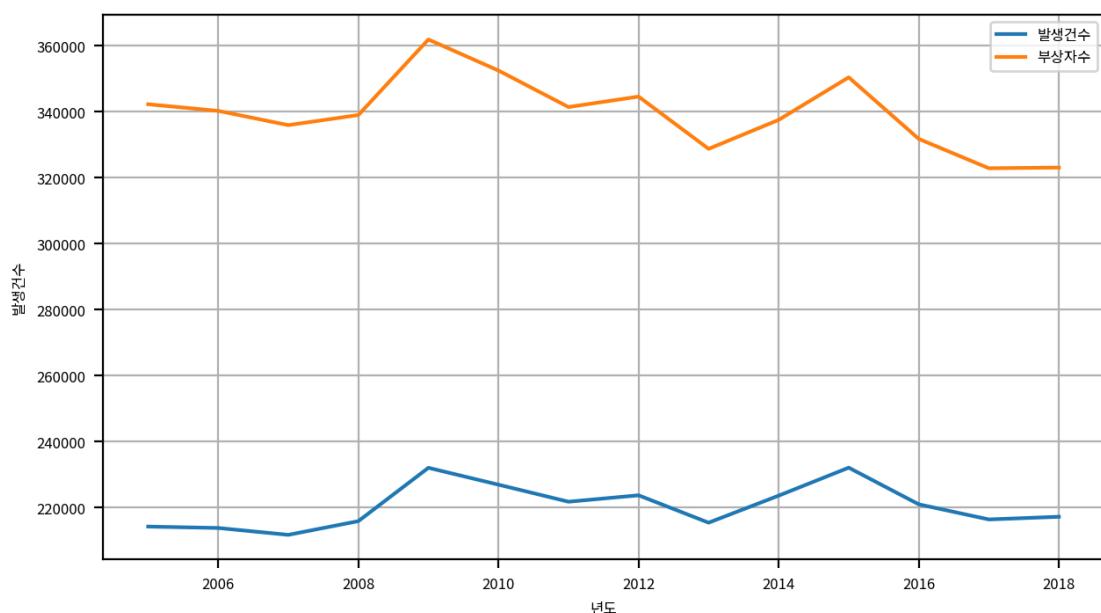
# 1) 그래프 초기화
width_px    = 1280          # 그래프 가로 크기
height_px   = 720           # 그래프 세로 크기
rows         = 1              # 그래프 행 수
cols         = 1              # 그래프 열 수
figsize      = (width_px / my_dpi, height_px / my_dpi)
fig, ax = plt.subplots(rows, cols, figsize=figsize, dpi=my_dpi)

# 2) 선 그래프
sb.lineplot(data=df, x=df.index, y='발생건수', label='발생건수')
sb.lineplot(data=df, x=df.index, y='부상자수', label='부상자수')

# 3) 그래프 꾸미기
ax.grid(True)                # 배경 격자 표시/숨김

# 4) 출력
plt.tight_layout()            # 여백 제거
plt.show()                    # 그래프 화면 출력
plt.close()                   # 그래프 작업 종료

```



png

## #04. hue 파라미터 이해하기

### 1. 데이터 전처리 (melt)

melt를 적용할 경우 index 해제

```
df1 = df.reset_index()
df1
```

	년도	발생건수	사망자수	부상자수
0	2005	214171	6376	342233
1	2006	213745	6327	340229
2	2007	211662	6166	335906
3	2008	215822	5870	338962
4	2009	231990	5838	361875
5	2010	226878	5505	352458
6	2011	221711	5229	341391
7	2012	223656	5392	344565
8	2013	215354	5092	328711
9	2014	223552	4762	337497

10	2015	232035	4621	350400
11	2016	220917	4292	331720
12	2017	216335	4185	322829
13	2018	217148	3781	323037

```
df2 = df1.melt(id_vars='년도', value_vars=['발생건수', '사망자수', '부상자수'],
                 var_name='구분')
df2
```

	년도	구분	value
0	2005	발생건수	214171
1	2006	발생건수	213745
2	2007	발생건수	211662
3	2008	발생건수	215822
4	2009	발생건수	231990
5	2010	발생건수	226878
6	2011	발생건수	221711
7	2012	발생건수	223656
8	2013	발생건수	215354
9	2014	발생건수	223552
10	2015	발생건수	232035
11	2016	발생건수	220917
12	2017	발생건수	216335
13	2018	발생건수	217148
14	2005	사망자수	6376
15	2006	사망자수	6327
16	2007	사망자수	6166
17	2008	사망자수	5870
18	2009	사망자수	5838
19	2010	사망자수	5505
20	2011	사망자수	5229
21	2012	사망자수	5392
22	2013	사망자수	5092

23	2014	사망자수	4762
24	2015	사망자수	4621
25	2016	사망자수	4292
26	2017	사망자수	4185
27	2018	사망자수	3781
28	2005	부상자수	342233
29	2006	부상자수	340229
30	2007	부상자수	335906
31	2008	부상자수	338962
32	2009	부상자수	361875
33	2010	부상자수	352458
34	2011	부상자수	341391
35	2012	부상자수	344565
36	2013	부상자수	328711
37	2014	부상자수	337497
38	2015	부상자수	350400
39	2016	부상자수	331720
40	2017	부상자수	322829
41	2018	부상자수	323037



## 2. hue 파라미터를 사용한 시각화

```

# 1) 그래프 초기화
width_px    = 1280                         # 그래프 가로 크기
height_px   = 720                           # 그래프 세로 크기
rows         = 1                             # 그래프 행 수
cols         = 1                             # 그래프 열 수
figsize      = (width_px / my_dpi, height_px / my_dpi)
fig, ax      = plt.subplots(rows, cols, figsize=figsize, dpi=my_dpi)

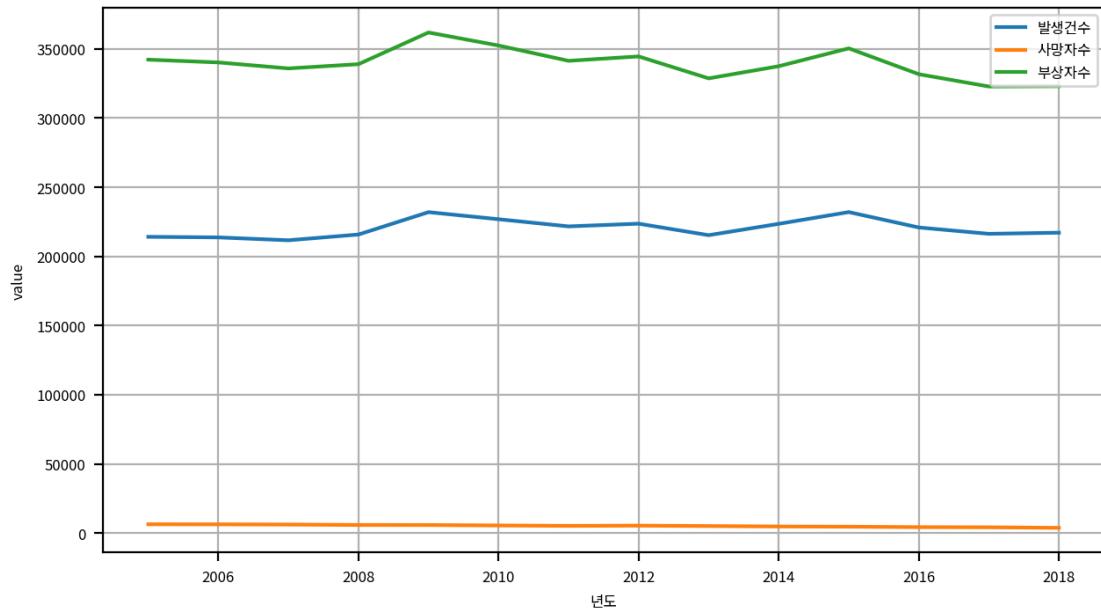
# 2) LinePlot 그리기
sb.lineplot(data=df2, x="년도", y="value", hue="구분")

# 3) 그래프 꾸미기
ax.grid(True)                                # 배경 격자 표시/숨김
ax.legend(loc="upper right")

# 4) 출력

```

```
plt.tight_layout()      # 여백 제거
plt.savefig("plot.png", dpi=my_dpi * 2)
plt.show()              # 그래프 화면 출력
plt.close()             # 그래프 작업 종료
```



png