! !

3 - LS

,

---

## Table of content

- : **3**
- : , , ,

---

- ! , ,
- , , !

- !

---

---

1. .
2. .
3. .
4. .

---

---

-

---

-

```python
import pandas as pd
import numpy as np
import matplotlib as mpl

mpl.rcParams['font.family'] = 'Malgun Gothic'
camera = pd.read_csv('../team2/data/camera.csv')
accident = pd.read_csv('../team2/data/accident.csv', encoding='cp949')
```

---

-

```python
print(camera.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 27403 entries, 0 to 27402
Data columns (total 21 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0             27403 non-null  object
 1                  27403 non-null  object
 2                  27403 non-null  object
 3                27403 non-null  object
 4               7964 non-null   object
 5                27403 non-null  object
 6               27403 non-null  int64
 7              11385 non-null  object
 8               24525 non-null  object
 9                 27403 non-null  float64
 10                27403 non-null  float64
 11               27403 non-null  object
 12               27403 non-null  int64
 13               27403 non-null  int64
 14             1123 non-null   float64
 15             954 non-null    float64
 16              27403 non-null  int64
 17               27403 non-null  int64
 18               27403 non-null  object
 19             27403 non-null  object
 20              27403 non-null  object
dtypes: float64(4), int64(5), object(12)
memory usage: 4.4+ MB
None
```

---

-

```
print(accident.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 230 entries, 0 to 229
Data columns (total 7 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0         230 non-null    object
 1         230 non-null    object
 2        230 non-null    int64
 3        230 non-null    int64
 4        230 non-null    int64
 5        230 non-null    int64
 6       230 non-null    int64
dtypes: int64(5), object(2)
memory usage: 12.7+ KB
None
```

- 1   :

```
accident['  '] = accident['  '].replace(' ', '   ')
accident['  '] = accident['  '].replace(' ', '   ')
accident['  '] = accident['  '].replace(' ', '  ')
accident['  '] = accident['  '].replace(' ', '  ')
accident['  '] = accident['  '].replace(' ', '  ')
accident['  '] = accident['  '].replace(' ', '  ')
accident['  '] = accident['  '].replace(' ', '  ')
accident['  '] = accident['  '].replace(' ', '   ')
accident['  '] = accident['  '].replace(' ', '   ')
accident['  '] = accident['  '].replace(' ', '   ')
accident['  '] = accident['  '].replace(' ', '    ')
accident['  '] = accident['  '].replace(' ', '  ')
accident['  '] = accident['  '].replace(' ', '  ')
accident['  '] = accident['  '].replace(' ', '    ')
accident['  '] = accident['  '].replace(' ', '   ')
accident['  '] = accident['  '].replace(' ', '    ')
accident['  '] = accident['  '].replace(' ', '   ')
```

```
#
theeshold = accident['   '].quantile(0.75)
accident['   '] = accident['   '].apply(lambda x: ' ' if x >= theeshold else ' ')

#
road_type = camera.groupby('  ')['   '].agg(lambda x: x.value_counts().index[0]).reset_index(
road_type = road_type.rename(columns={'   ': '    '})
```

- 1   :

```
#
acc_road = pd.merge(road_type, accident, on='  ')

#    vs
ct = pd.crosstab(acc_road['    '], acc_road['   '])

ct.loc['   '] = ct.loc['   '] + ct.loc['  ']
ct = ct.drop(['  '])
```

```
#
ct.style.set_table_styles([
    {'selector': 'th', 'props': [('background-color', '#f4d9c6'),
                                 ('color', '#333'),
                                 ('text-align', 'center'),
                                 ('font-family', 'Jua'),
                                 ('font-size', '1.1em')]},
    {'selector': 'td', 'props': [('text-align', 'center'),
                                 ('font-family', 'Jua'),
                                 ('padding', '10px')]},
    {'selector': '', 'props': [('border', '1px solid #ddd'),
                               ('border-collapse', 'collapse')]}
]).set_caption("            ") \
  .set_properties(**{
      'background-color': '#fffdf9',
      'border-color': '#eee',
      'border-style': 'solid',
      'border-width': '1px'
  })
```

Table 1:

| | | |
|---|---|---|
| | 13 | 37 |
| | 25 | 74 |
| | 6 | 36 |
| | 13 | 12 |

- 1 :

```
#
from scipy.stats import chi2_contingency
import matplotlib.pyplot as plt
chi2, p, dof, exp = chi2_contingency(ct)

print(chi2, p, dof)
```

```
11.678710326416384 0.008568853463999705 3
```

---

-

  (Chi²): **11.68**   (df): **3**   (p-value): **0.008**

  !

---

-

  [! ]   .

---

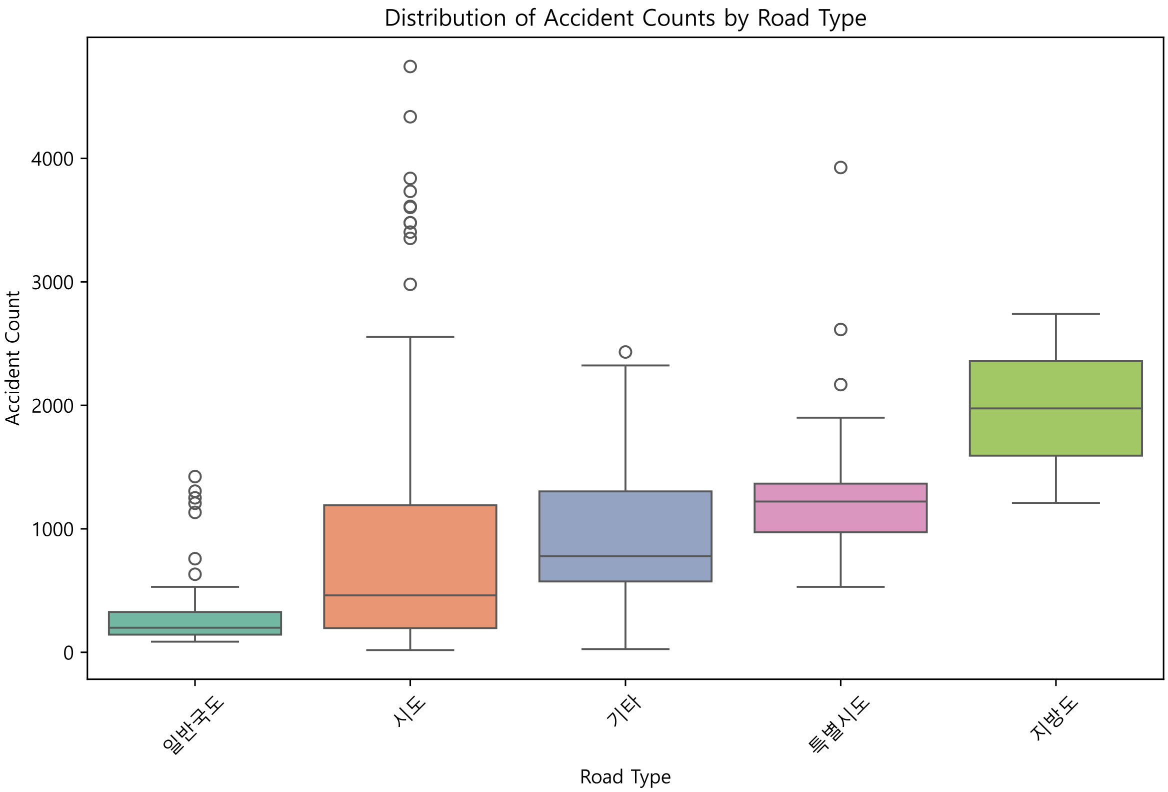
-

```python
import seaborn as sns
import matplotlib.pyplot as plt
plt.figure(figsize=(10,6))
sns.boxplot(x='   ', y='  ', data=acc_road, palette='Set2')
plt.xticks(rotation=45)
plt.title('Distribution of Accident Counts by Road Type')
plt.xlabel('Road Type')
plt.ylabel('Accident Count')
plt.show()
```

```
C:\Users\USER\AppData\Local\Temp\ipykernel_11132\891112310.py:4: FutureWarning:



Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assig
```

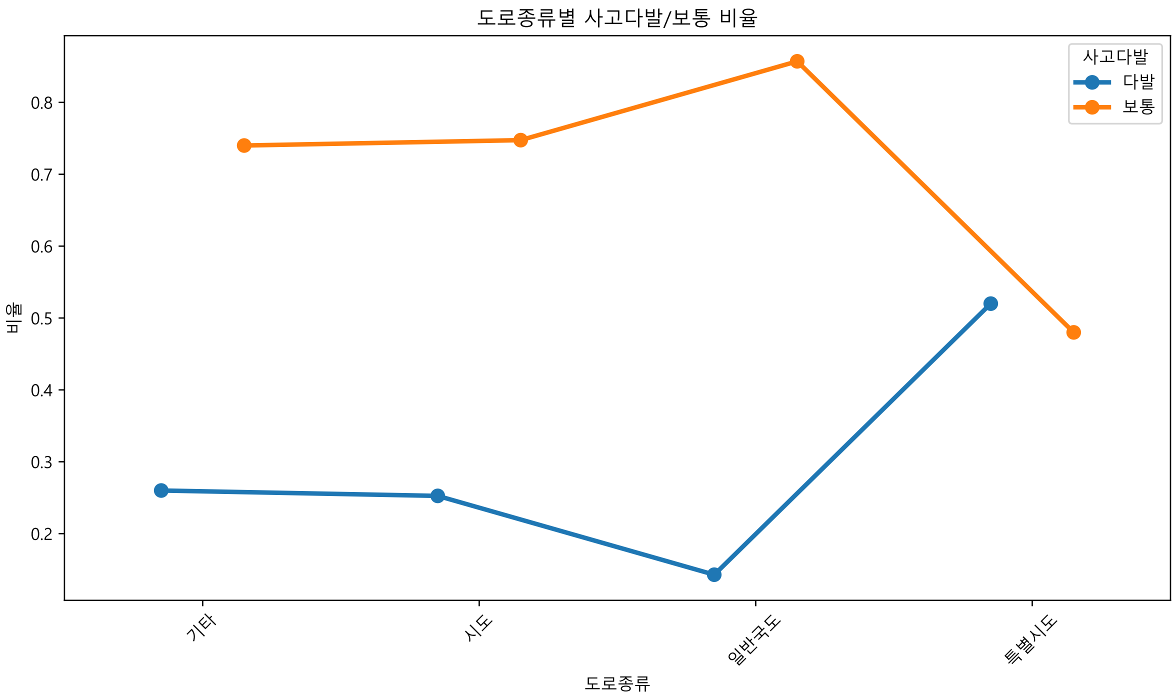


---

-

```
ct_norm = ct.div(ct.sum(axis=1), axis=0)  #
#       long-format
ct_long = ct_norm.reset_index().melt(id_vars='    ', var_name='   ', value_name='  ')

plt.figure(figsize=(10, 6))
sns.pointplot(data=ct_long, x='    ', y=' ', hue='    ', dodge=0.3, markers='o', linestyles='-'
plt.xticks(rotation=45)
plt.title('      /    ')
plt.ylabel('  ')
plt.xlabel('   ')
plt.tight_layout()
plt.show()
```



---

-

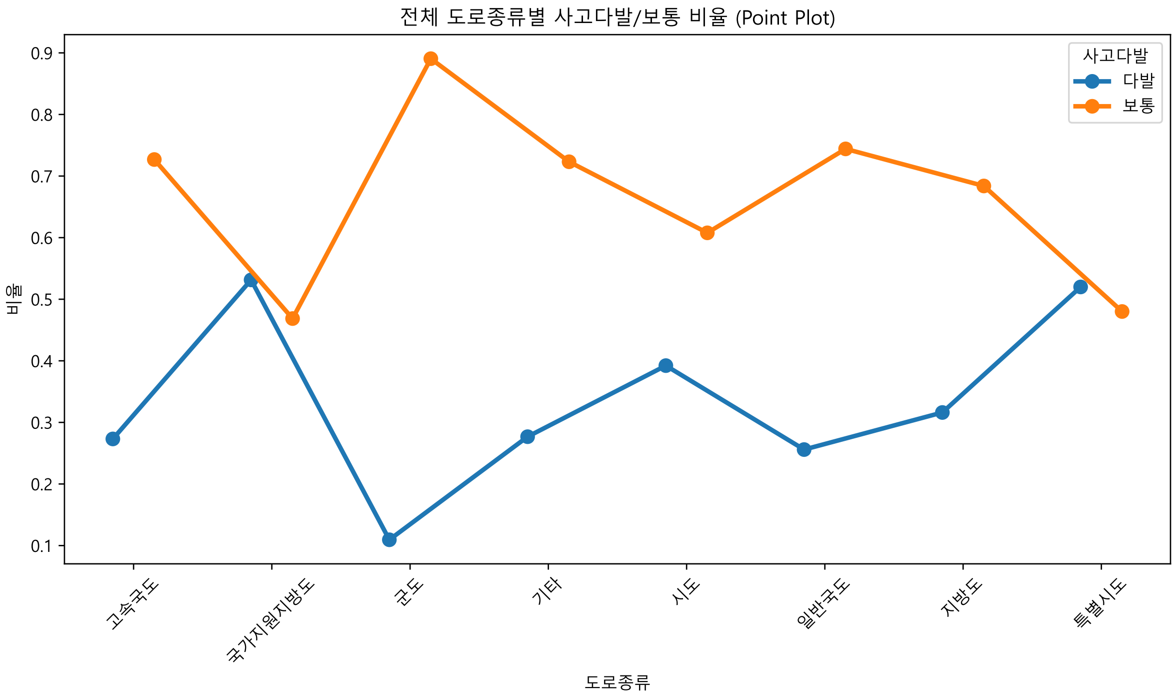
```
# accident          (    )
threshold = accident['   '].quantile(0.75)
accident['   '] = accident['   '].apply(lambda x: ' ' if x >= threshold else ' ')

# camera  accident               (        )
all_road = pd.merge(camera[['  ', '   ']], accident[['  ', '   ', '   ']], on='  ')

#
ct_all = pd.crosstab(all_road['   '], all_road['   '])
```

```
#
ct_all_norm = ct_all.div(ct_all.sum(axis=1), axis=0)
ct_all_long = ct_all_norm.reset_index().melt(id_vars='   ', var_name='   ', value_name='  ')

plt.figure(figsize=(10, 6))
sns.pointplot(data=ct_all_long, x='   ', y='  ', hue='   ', dodge=0.3, markers='o', linestyles=
plt.xticks(rotation=45)
plt.ylabel(' ')
plt.xlabel('  ')
plt.title('        /    (Point Plot)')
plt.tight_layout()
plt.show()
```



---

- Shapiro-Wilk

```
from scipy.stats import shapiro
def check_normality_by_road_type(all_road):
    #
    results = []

    #
    for name, group in all_road.groupby('  '):
        stat, p = shapiro(group['  '])
        result = '    ' if p < 0.05 else '  '
        results.append([name, p, result])
    #
    df_results = pd.DataFrame(results, columns=['  ', 'p-value', '    '])

    #
    return df_results
```

```
df_normality = check_normality_by_road_type(all_road)
df_normality
```

```
C:\Users\USER\.conda\envs\ls_pyun\Lib\site-packages\scipy\stats\_axis_nan_policy.py:586: Use

scipy.stats.shapiro: For N > 5000, computed p-value may not be accurate. Current N is 8801.

C:\Users\USER\.conda\envs\ls_pyun\Lib\site-packages\scipy\stats\_axis_nan_policy.py:586: Use

scipy.stats.shapiro: For N > 5000, computed p-value may not be accurate. Current N is 26516.

C:\Users\USER\.conda\envs\ls_pyun\Lib\site-packages\scipy\stats\_axis_nan_policy.py:586: Use

scipy.stats.shapiro: For N > 5000, computed p-value may not be accurate. Current N is 53789.

C:\Users\USER\.conda\envs\ls_pyun\Lib\site-packages\scipy\stats\_axis_nan_policy.py:586: Use

scipy.stats.shapiro: For N > 5000, computed p-value may not be accurate. Current N is 216655

C:\Users\USER\.conda\envs\ls_pyun\Lib\site-packages\scipy\stats\_axis_nan_policy.py:586: Use

scipy.stats.shapiro: For N > 5000, computed p-value may not be accurate. Current N is 77536.

C:\Users\USER\.conda\envs\ls_pyun\Lib\site-packages\scipy\stats\_axis_nan_policy.py:586: Use

scipy.stats.shapiro: For N > 5000, computed p-value may not be accurate. Current N is 57216.

C:\Users\USER\.conda\envs\ls_pyun\Lib\site-packages\scipy\stats\_axis_nan_policy.py:586: Use

scipy.stats.shapiro: For N > 5000, computed p-value may not be accurate. Current N is 53306.
```

| | p-value |
|---|---|
| 0 | 6.372202e-78 |
| 1 | 4.453160e-29 |

|   | p-value |
|---|---|
| 2 | 8.578405e-119 |
| 3 | 3.224323e-109 |
| 4 | 1.458738e-146 |
| 5 | 6.107652e-135 |
| 6 | 2.914311e-122 |
| 7 | 4.290432e-124 |

---

- Kruskal Wallis H

```
from scipy.stats import kruskal

groups = [g['   '].values for _, g in all_road.groupby('   ')]
stat, p = kruskal(*groups)
print("Kruskal-Wallis H :", p)
```

```
Kruskal-Wallis H : 0.0
```

---

- Kruskal Wallis H

  [! ]                                    .

---

-  2     :

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

#
plt.rcParams['font.family'] ='Malgun Gothic'
plt.rcParams['axes.unicode_minus'] =False

accident_df = pd.read_csv('../team2/data/accident.csv', encoding='cp949')

#    unique
unique_values = accident_df['   '].unique()
```

---

-

```
#
accident_avg = accident_df.groupby('  ')['   '].mean().reset_index()
```

---

-

```
ac_avg = accident_avg.sort_values(by='   ', ascending=False)

ac_avg.head(7)
```

| | |
|---|---|
| 14 | 1975.000000 |
| 1 | 1714.967742 |
| 6 | 1479.600000 |
| 4 | 1394.400000 |
| 8 | 1352.440000 |
| 5 | 1208.888889 |
| 9 | 1166.000000 |

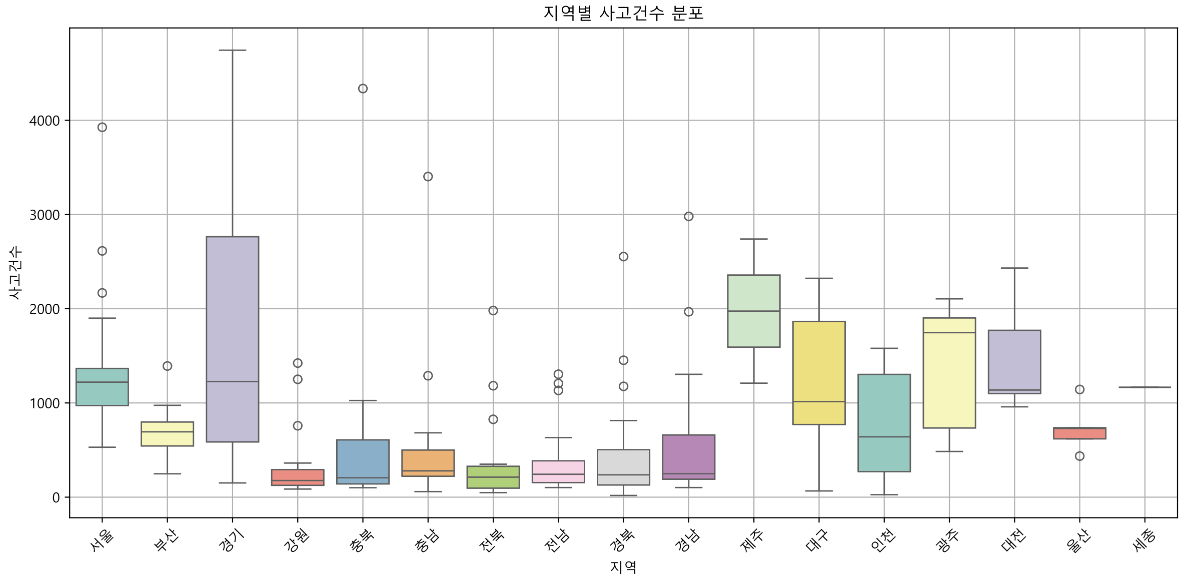
---

-

```
#
plt.figure(figsize=(12, 6))
sns.boxplot(data=accident_df, x='  ', y='   ', palette='Set3')
plt.title('       ')
plt.xlabel(' ')
plt.ylabel('  ')
plt.xticks(rotation=45)
plt.tight_layout()
plt.grid(True)
plt.show()
```

```
C:\Users\USER\AppData\Local\Temp\ipykernel_11132\4283594563.py:3: FutureWarning:
```

```
Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assig
```



-

  [! ]                              !

- ANOVA

```
#             : ANOVA
import statsmodels.api as sm
from statsmodels.formula.api import ols

model = ols('    ~ C(  )', data=accident_df).fit()

anova_results = sm.stats.anova_lm(model, typ=2)
```

H0:                 .    H1:                       .

---

```
from IPython.display import HTML

anova_results = pd.DataFrame(anova_results)

styled = anova_results.style.set_table_styles([
    {'selector': 'th', 'props': [('background-color', '#f4d9c6'),
                                 ('color', '#333'),
                                 ('text-align', 'center'),
                                 ('font-family', 'Jua'),
                                 ('font-size', '1.1em')]},
    {'selector': 'td', 'props': [('text-align', 'center'),
                                 ('font-family', 'Jua'),
                                 ('padding', '10px')]},
    {'selector': '', 'props': [('border', '1px solid #ddd'),
                               ('border-collapse', 'collapse')]}
]).set_caption("  ANOVA     ")

HTML(styled.to_html())
```

Table 4:   ANOVA

| | sum_sq | df | F | PR(>F) |
|---|---|---|---|---|
| C(   ) | 55213437.753574 | 16.000000 | 5.600300 | 0.000000 |
| Residual | 131248122.611644 | 213.000000 | nan | nan |

---

- ANOVA

  [! ]                                        .

---

- Kruskal Wallis H

```
from scipy.stats import kruskal

grouped_values = [group['  '].values for _, group in accident_df.groupby('  ')]
stat, p = kruskal(*grouped_values)
stat, p
```

```
(89.2757873932235, 3.4006688531866835e-12)
```

---

- Kruskal Wallis H

  [! ]                                  !

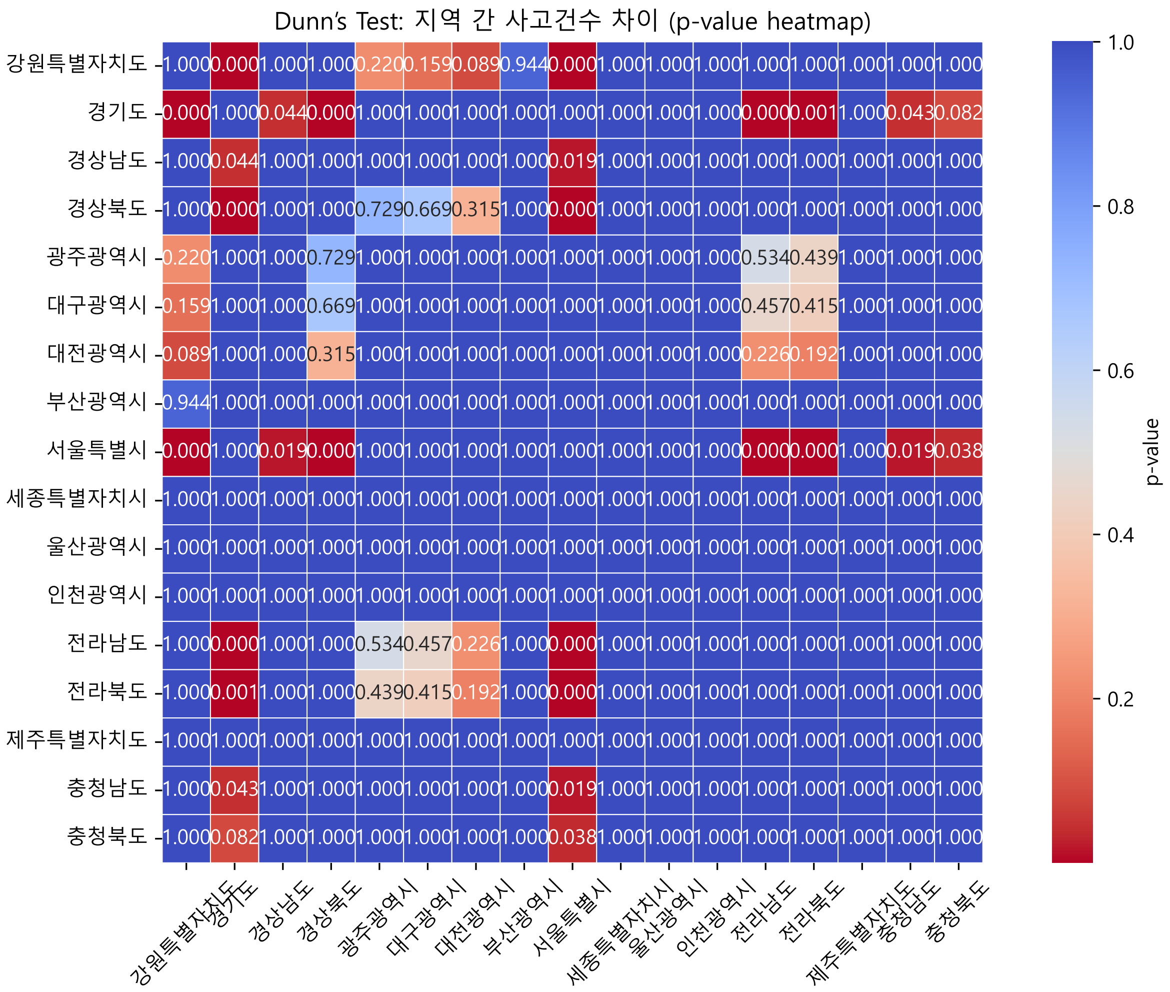
---

- : Dunn’s test

```
import scikit_posthocs as sp_post

posthoc = sp_post.posthoc_dunn(accident, val_col='  ', group_col='  ', p_adjust='bonferroni')
# Bonferroni

#
plt.figure(figsize=(10, 7))
sns.heatmap(posthoc,
            annot=True,
            fmt=".3f",
            cmap="coolwarm_r",
            cbar_kws={'label': 'p-value'},
            linewidths=0.5,
            square=True)
plt.title("Dunn’s Test:          (p-value heatmap)")
plt.xticks(rotation=45)
plt.yticks(rotation=0)
plt.tight_layout()
plt.show()
```



| | 강원특별자치도 | 경기도 | 경상남도 | 경상북도 | 광주광역시 | 대구광역시 | 대전광역시 | 부산광역시 | 서울특별시 | 세종특별자치시 | 울산광역시 | 인천광역시 | 전라남도 | 전라북도 | 제주특별자치도 | 충청남도 | 충청북도 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 강원특별자치도 | 1.000 | 0.000 | 1.000 | 1.000 | 0.220 | 0.159 | 0.089 | 0.944 | 0.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| 경기도 | 0.000 | 1.000 | 0.044 | 0.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 0.000 | 0.001 | 1.000 | 0.043 | 0.082 |
| 경상남도 | 1.000 | 0.044 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 0.019 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| 경상북도 | 1.000 | 0.000 | 1.000 | 1.000 | 0.729 | 0.669 | 0.315 | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| 광주광역시 | 0.220 | 1.000 | 1.000 | 0.729 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 0.534 | 0.439 | 1.000 | 1.000 | 1.000 |
| 대구광역시 | 0.159 | 1.000 | 1.000 | 0.669 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 0.457 | 0.415 | 1.000 | 1.000 | 1.000 |
| 대전광역시 | 0.089 | 1.000 | 1.000 | 0.315 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 0.226 | 0.192 | 1.000 | 1.000 | 1.000 |
| 부산광역시 | 0.944 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| 서울특별시 | 0.000 | 1.000 | 0.019 | 0.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 0.000 | 0.001 | 1.000 | 0.019 | 0.038 |
| 세종특별자치시 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| 울산광역시 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| 인천광역시 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| 전라남도 | 1.000 | 0.000 | 1.000 | 1.000 | 0.534 | 0.457 | 0.226 | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| 전라북도 | 1.000 | 0.001 | 1.000 | 1.000 | 0.439 | 0.415 | 0.192 | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| 제주특별자치도 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| 충청남도 | 1.000 | 0.043 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 0.019 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| 충청북도 | 1.000 | 0.082 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 0.038 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |

---

- 3 :

```
accident_df = pd.read_csv('../team2/data/accident.csv', encoding='cp949')
c_df = pd.read_csv('../team2/data/camera.csv')

sido_map = {
    '   ': '  ',
    '   ': '  ',
    '   ': '  ',
    '   ': '  ',
    '   ': '  ',
    '   ': ' ',
    '   ': ' ',
    '    ': ' ',
    '  ': ' ',
    '    ': ' ',
    '   ': ' ',
    '   ': ' ',
    '    ': ' ',
    '   ': ' ',
    '   ': ' ',
    '   ': ' ',
    '    ': ' '
}
c_df['  '] = c_df['  '].replace(sido_map)
```

---

-

```
import geopandas as gpd
import plotly.express as px
import plotly.graph_objects as go

# geopandas  shp
gdf = gpd.read_file('../team2/data/BND_SIDO_PG.shp')
print(gdf.crs) #

gdf = gdf.to_crs(epsg=4326) # WGS84 (4326)


# GeoJSON
gdf.to_file('../team2/data/BND_SIDO_PG.geojson', driver='GeoJSON')

# geojson
import json
with open('../team2/data/BND_SIDO_PG.geojson', encoding='utf-8') as f:
    geojson_data = json.load(f)
print(geojson_data.keys()) #
print(geojson_data['features'][1]['properties']) #

#       groupby           DataFrame
```

```
agg_cam = camera.groupby('  ',as_index=False)['       '].count()
agg_cam.columns = ['SIDO_NM','   ']
```

```
# plotly
fig = px.choropleth_mapbox(
agg_cam,
geojson=geojson_data,
locations="SIDO_NM",
featureidkey="properties.SIDO_NM",
color="   ",
color_continuous_scale="Blues",
mapbox_style="carto-positron",
center={"lat": 37.5665, "lon": 126.9780},
zoom=5,
opacity=1,
title="      ",
)
fig.update_layout(margin={"r":0,"t":30,"l":0,"b":0})
fig.show()
```

```
#    0
camera_count = c_df.groupby(['  ', '  ']).size().reset_index(name='   ')
camera_count.rename(columns={'  ': '  '}, inplace=True)

#
camera_merged_df = pd.merge(camera_count, accident_df.groupby(['  ', '  '])['  '].sum().reset


#          ( /  /  )
q1 = camera_merged_df['   '].quantile(0.25)
q3 = camera_merged_df['   '].quantile(0.75)

def classify_group(x):
    if x <= q1:
        return '  '
    elif x >= q3:
        return '  '
    else:
        return '  '
```

```
camera_merged_df[' '] = camera_merged_df['    '].apply(classify_group)
```

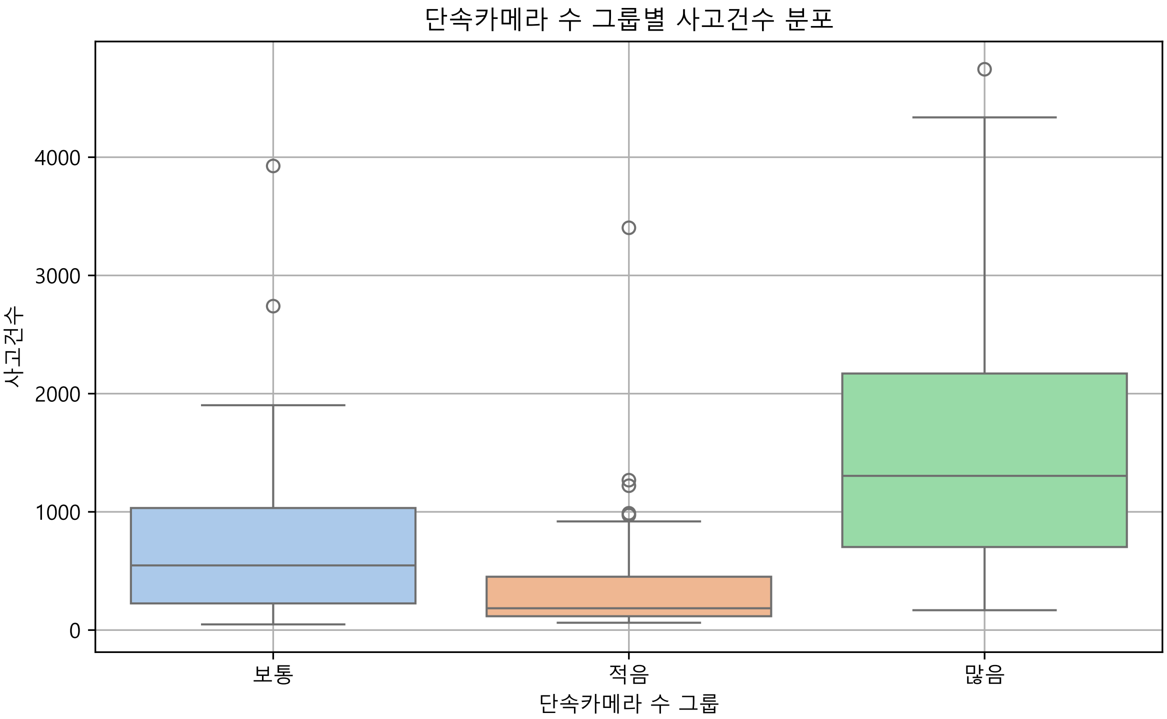
---

```
# Boxplot   :
plt.figure(figsize=(8,5))
sns.boxplot(x=' ', y='  ', data=camera_merged_df, palette='pastel')
plt.title('            ')
plt.xlabel('       ')
plt.ylabel('  ')
plt.grid(True)
plt.tight_layout()
plt.show()
```

```
C:\Users\USER\AppData\Local\Temp\ipykernel_11132\1637060997.py:3: FutureWarning:



Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assi
```



- ANOVA

```
from scipy.stats import f_oneway

group_   = camera_merged_df[camera_merged_df['  '] == '  ']['   ']
group_   = camera_merged_df[camera_merged_df['  '] == '  ']['   ']
group_   = camera_merged_df[camera_merged_df['  '] == '  ']['   ']

f_stat, p_value = f_oneway(group_ , group_ , group_ )
# p-value < 0.05,
```

```
from IPython.display import HTML

anova_result_df = pd.DataFrame({
    'F-  ': [f_stat],
    'p- ': [p_value]
})

styled = anova_result_df.style.set_table_styles([
    {'selector': 'th', 'props': [('background-color', '#f4d9c6'),
                                 ('color', '#333'),
                                 ('text-align', 'center'),
                                 ('font-family', 'Jua'),
                                 ('font-size', '1.1em')]},
    {'selector': 'td', 'props': [('text-align', 'center'),
                                 ('font-family', 'Jua'),
                                 ('padding', '10px')]},
    {'selector': '', 'props': [('border', '1px solid #ddd'),
                               ('border-collapse', 'collapse')]}
]).set_caption("  ANOVA     ")

#
HTML(styled.to_html())
```

Table 5: ANOVA

| | F- | p- |
|---|---|---|
| 0 | 43.666575 | 0.000000 |

H0: . ( ) H1: .

---

- Shapiro-Wilk / Levene

```
results = []
# shapiro:
from scipy.stats import shapiro

for name, group in zip([' ', ' ', ' '], [group_ , group_ , group_ ]):
    stat, p = shapiro(group)
    norm_check = '   O' if p > 0.05 else '   X'
    results.append([name, round(p, 4), norm_check])

# Levene:
from scipy.stats import levene

stat, p = levene(group_ , group_ , group_ )
equal_check = '   O' if p > 0.05 else '   X'
results.append(['     ', round(p, 4), equal_check])

df_results = pd.DataFrame(results, columns=[' ', 'p-value', ' '])

df_results
```

| | | p-value | |
|---|---|---|---|
| 0 | | 0.0000 | X |
| 1 | | 0.0000 | X |
| 2 | | 0.0003 | X |
| 3 | | 0.0000 | X |

---

- Kruskal Wallis H

```
from scipy.stats import kruskal

h_stat, p_value = kruskal(group_ , group_ , group_ )
print(f"Kruskal-Wallis H     : {h_stat:.3f}, p-value: {p_value:.4f}")
```

```
Kruskal-Wallis H     : 71.014, p-value: 0.0000
```

---

- Kruskal Wallis H

  [! ]                                        !

---

- : Dunn's test

```
import scikit_posthocs as sp

dunn_result = sp.posthoc_dunn(camera_merged_df, val_col='  ', group_col=' ', p_adjust='bonfe

#
pairs = []
for i, group_a in enumerate(dunn_result.index):
    for j, group_b in enumerate(dunn_result.columns):
        if i < j:  #     (     )
            pairs.append({
                '  A': group_a,
                '  B': group_b,
                'p-value': dunn_result.iloc[i, j]
            })

#
dunn_pairs_df = pd.DataFrame(pairs)

dunn_pairs_df
```

|   | A | B | p-value |
|---|---|---|---|
| 0 |  |  | 1.598669e-07 |
| 1 |  |  | 1.399293e-16 |

|   | A | B | p-value |
|---|---|---|---|
| 2 |   |   | 6.733709e-05 |

- 4     :

```
#            True,    False
c_df['   '] = c_df['   '].notna()

#
roadline_status = c_df.groupby(['  ', '  '])['   '] \
                      .agg(lambda x: x.value_counts().index[0]) \
                      .reset_index(name='   ')

roadline_status.rename(columns={'  ': '  '}, inplace=True)

accident_with_roadline = pd.merge(
    accident_df[['  ', '  ', '  ']],
    roadline_status,
    on=['  ', '  '])
```

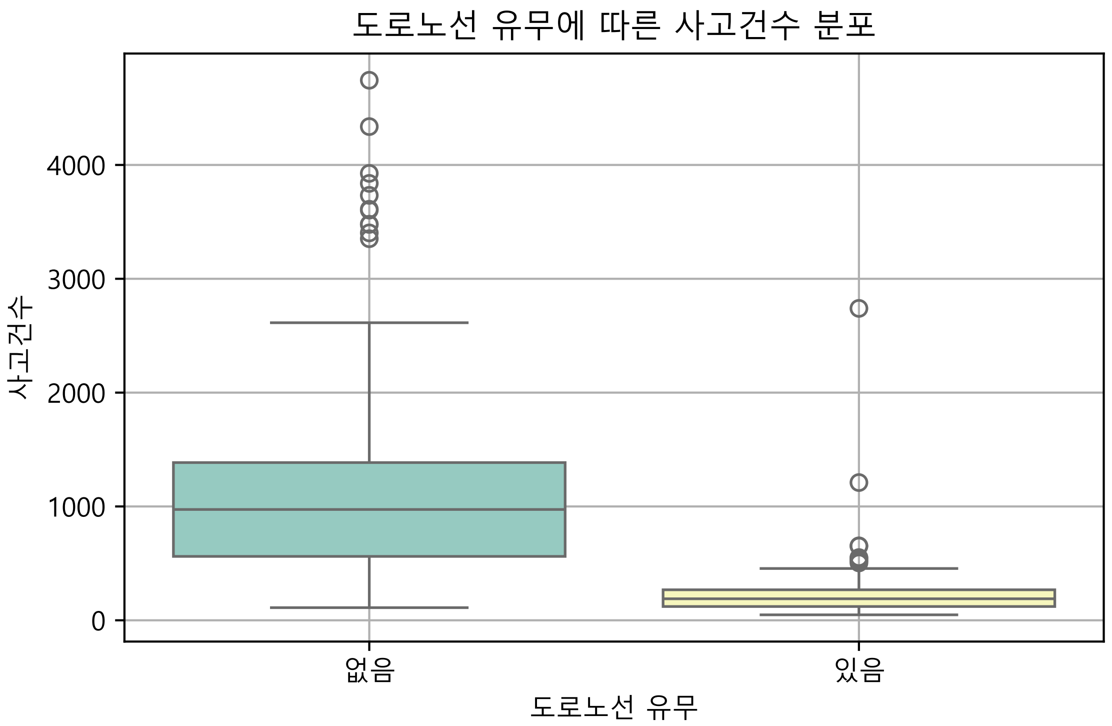
```
accident_with_roadline['   '] = accident_with_roadline['   '].map({True: ' ', False: ' '})

plt.figure(figsize=(6, 4))
sns.boxplot(data=accident_with_roadline, x='   ', y='   ', palette='Set3')
plt.title('            ')
plt.xlabel('    ')
plt.ylabel('  ')
plt.grid(True)
plt.tight_layout()
plt.show()
```

```
C:\Users\USER\AppData\Local\Temp\ipykernel_11132\2032049726.py:4: FutureWarning:



Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assi
```



---

- 2 T

```
from scipy.stats import shapiro, ttest_ind, mannwhitneyu

group_with = accident_with_roadline[accident_with_roadline['    '] == ' ']['  ']
group_without = accident_with_roadline[accident_with_roadline['    '] == ' ']['  ']

t_stat, p_value = ttest_ind(group_with, group_without, equal_var=False)

results = t_stat, p_value
```

---

- 2 T

```
  = '   ' if p_value < 0.05 else '    '

t_stat, p_value = ttest_ind(group_with, group_without, equal_var=False)

results = [['            ', f"{p_value:.3f}",  ]]

# DataFrame
df_results = pd.DataFrame(results, columns=['   ', 'p_value', ' '])

df_results
```

|  |  | p_value |
|---|---|---|
| 0 |  | 0.000 |

---

- Shaprio-Wilk    / Mann-Whitney U

```
from scipy.stats import shapiro, ttest_ind, mannwhitneyu

stat, p = shapiro(group_with)
out_stat, out_p = shapiro(group_without)

stat, p = mannwhitneyu(group_with, group_without)
```

---

- Shaprio-Wilk    / Mann-Whitney U

```
#
results = []
stat1, p1 = shapiro(group_with)
result1 = '   ' if p1 > 0.05 else '     '
results.append(['    ', round(p1, 4), result1])

stat2, p2 = shapiro(group_without)
result2 = '   ' if p2 > 0.05 else '     '
results.append(['    ', round(p2, 4), result2])

#      (Mann-Whitney U)
```

```
mw_stat, mw_p = mannwhitneyu(group_with, group_without)

#   DataFrame
results.append(['Mann-Whitney U  ', round(mw_p, 4), '      ' if mw_p < 0.05 else '    '])

df_results = pd.DataFrame(results, columns=['    /   ', 'p-value', ' '])

styled_df = df_results.style.set_table_attributes('class="styled-table"')

styled_df
```

Table 9

| | / | p-value |
|---|---|---|
| 0 | | 0.000000 |
| 1 | | 0.000000 |
| 2 | Mann-Whitney U | 0.000000 |

---

- Levene    / Brunner Munzel

```
#
from scipy.stats import levene
stat, p = levene(group_with, group_without)

from scipy.stats.mstats import brunnermunzel
stat, pvalue = brunnermunzel(group_with, group_without, alternative='two-sided')
```

---

- Levene    / Brunner Munzel

```
#
results = []
stat, p = levene(group_with, group_without)
result1 = '  ' if p1 > 0.05 else '    '
results.append(['Levene', round(p1, 4), result1])

stat, pvalue = brunnermunzel(group_with, group_without, alternative='two-sided')
```

```
result2 = '    ' if p2 > 0.05 else '     X'
results.append(['BM', round(p2, 4), result2])

df_results = pd.DataFrame(results, columns=[' ', 'p-value', ' '])

styled_df = df_results.style.set_table_attributes('class="styled-table"')

styled_df
```

Table 10

|   |   | p-value |   |
|---|---|---|---|
| 0 | Levene | 0.000000 |   |
| 1 | BM | 0.000000 | X |

---

- Brunner Munzel

  [! ]                                . ,                                 !

---

-

```
camera['   '] = camera['   '].notna()
camera.rename(columns={'  ': '  '}, inplace=True)

camera_count = camera.groupby(['  ', '  ']).size().reset_index(name='   ')
camera_merged_df = pd.merge(camera_count, accident.groupby(['  ', '  '])['  '].sum().reset_in

merged_df = pd.merge(camera_merged_df, camera[['  ', '  ', '  ', '   ']], on=['  ', '  '])

summary_df = (
    merged_df.groupby(['  ', '  '])
    .agg({
        '   ': 'first',
        '  ': 'first',
        '   ': 'mean'  # True
    })
    .reset_index()
)

#
summary_df = summary_df[['  ', '   ', '  ', '   ']]
summary_df.rename(columns={'   ': '   '}, inplace=True)
```

---

-

```
pop_df = pd.read_csv('../team2/data/ _ _ .csv')

pop_df = pop_df.rename(columns={pop_df.columns[0]: '  ', pop_df.columns[1]: '  '})
pop_df = pop_df[~pop_df['  '].str.contains('  |  ')]  #
pop_df = pop_df.dropna()  #     NaN
pop_df['  '] = pop_df['  '].astype(int)

final_df = pd.merge(summary_df, pop_df, on='  ', how='left')
```

---

-

```
from statsmodels.formula.api import ols

model = ols('   ~     +     +   ', final_df).fit()
```

1. y :
2. X : , , /

---

```
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                        R-squared:                        0.695
Model:                            OLS   Adj. R-squared:                  0.692
Method:                 Least Squares   F-statistic:                     244.1
Date:                Thu, 17 Apr 2025   Prob (F-statistic):           1.38e-82
```

```
Time:                        20:00:30   Log-Likelihood:                -2449.5
No. Observations:                 326   AIC:                             4907.
Df Residuals:                     322   BIC:                             4922.
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept    110.4797     55.679      1.984      0.048       0.939     220.021
            3.2748      0.384      8.517      0.000       2.518       4.031
         -305.3707    109.047     -2.800      0.005    -519.905     -90.836
              0.0022      0.000     12.598      0.000       0.002       0.003
==============================================================================
Omnibus:                       71.350   Durbin-Watson:                   1.188
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              196.981
Skew:                           1.004   Prob(JB):                     1.68e-43
Kurtosis:                       6.236   Cond. No.                     1.39e+06
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.39e+06. This might indicate that there are
strong multicollinearity or other numerical problems.
```

1. Adj. R-squared: 0.699 !
2. Prob (F-statistic): 2.44e-87 !

---

- OLS

```
coef = model.params
p_values = model.pvalues
conf_int = model.conf_int()

#      (       )
variables = ['Intercept', '    ', '    ', '  ']

#
summary_table = pd.DataFrame({
    ' ': variables,
    'coef': [coef[var] for var in variables],
    'p': [p_values[var] for var in variables],
    '0.025': [conf_int.loc[var][0] for var in variables],
    '0.975': [conf_int.loc[var][1] for var in variables]
})

r2 = model.rsquared
adj_r2 = model.rsquared_adj

#
display(summary_table.round(3))
```

| | | coef | p | 0.025 | 0.975 |
|---|---|---|---|---|---|
| 0 | Intercept | 110.480 | 0.048 | 0.939 | 220.021 |
| 1 | | 3.275 | 0.000 | 2.518 | 4.031 |
| 2 | | -305.371 | 0.005 | -519.905 | -90.836 |
| 3 | | 0.002 | 0.000 | 0.002 | 0.003 |

1. p-value 5%
2. “ ”

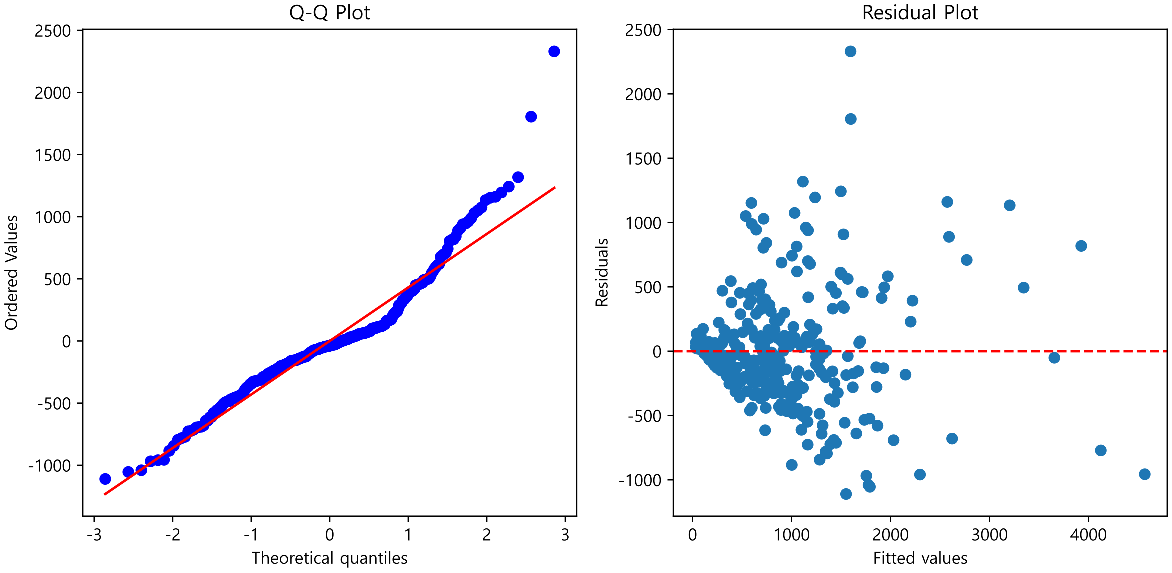
---

```
import matplotlib.pyplot as plt
import scipy.stats as stats
#| echo: true
#| code-fold: true
residuals = model.resid
#      (1  2 )
fig, axes = plt.subplots(1, 2, figsize=(10, 5))

# Q-Q plot
stats.probplot(residuals, dist="norm", plot=axes[0])
axes[0].set_title("Q-Q Plot")

# Residual plot
axes[1].scatter(model.fittedvalues, residuals)
axes[1].axhline(0, color='red', linestyle='--')
axes[1].set_xlabel('Fitted values')
axes[1].set_ylabel('Residuals')
axes[1].set_title('Residual Plot')
```

```
#
plt.tight_layout()
plt.show()
```



```
import scipy.stats as sp
W, p = sp.shapiro(model.resid)
```

---

-

```
results = []
W, p = shapiro(model.resid)
result1 = '  ' if p > 0.05 else '   X'
results.append(['shapiro', round(p, 4), result1])

df_results = pd.DataFrame(results, columns=['  ', 'p-value', '  '])

styled_df = df_results.style.set_table_attributes('class="styled-table"')

styled_df
```

Table 12

| | | p-value | |
|---|---|---|---|
| 0 | shapiro | 0.000000 | X |

, Durbin-Watson !

. .

---

---

-

, , , , .

-

1. .
2. .
3. .

---

**!**