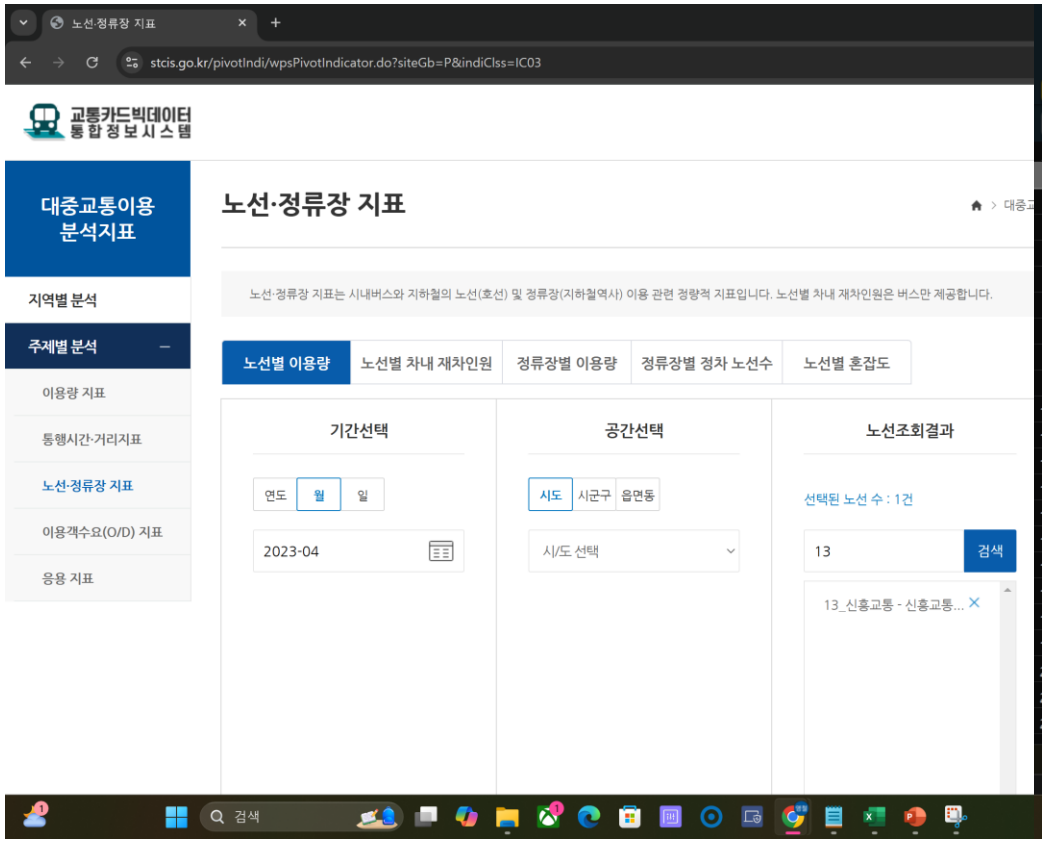
빅데이터처리 프로젝트
학교주변 버스 이용 현황 통계

컴퓨터정보과 202044089 조영철

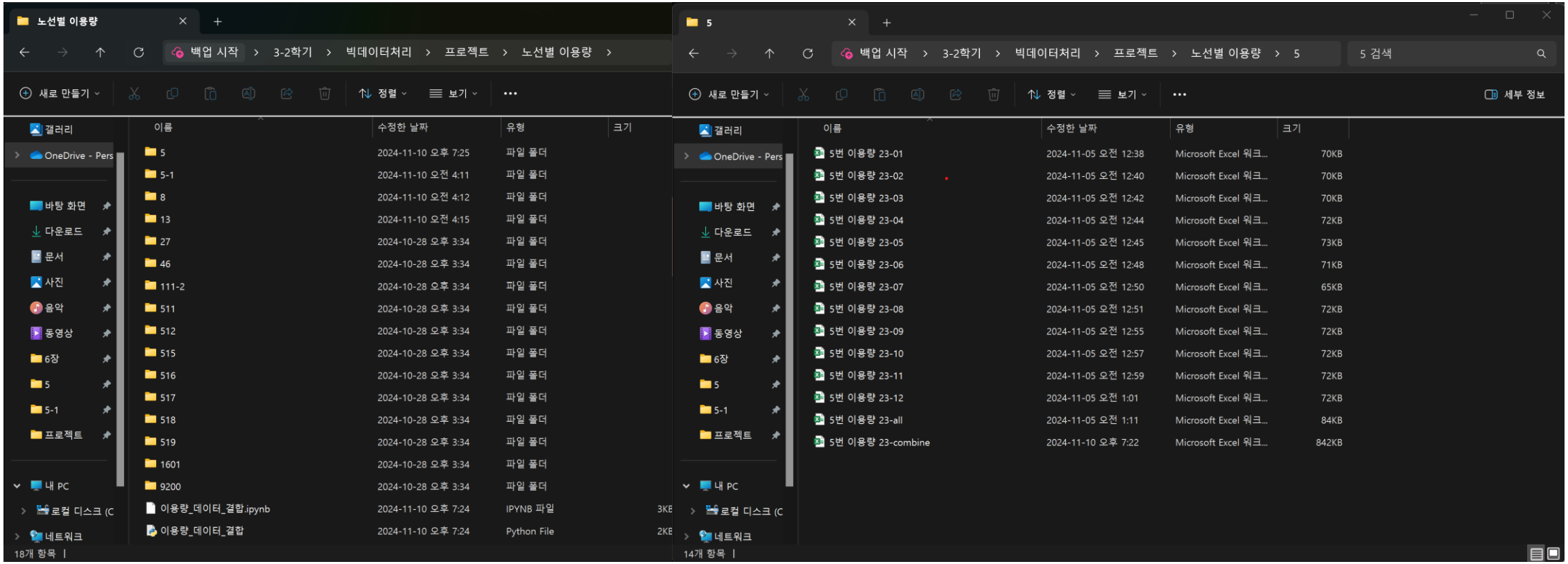
1. 데이터 수집 과정

- 교통카드빅데이터 통합정보시스템 사이트에서 학교 주변을 지나는 노선의 노선별 이용량 데이터 수집



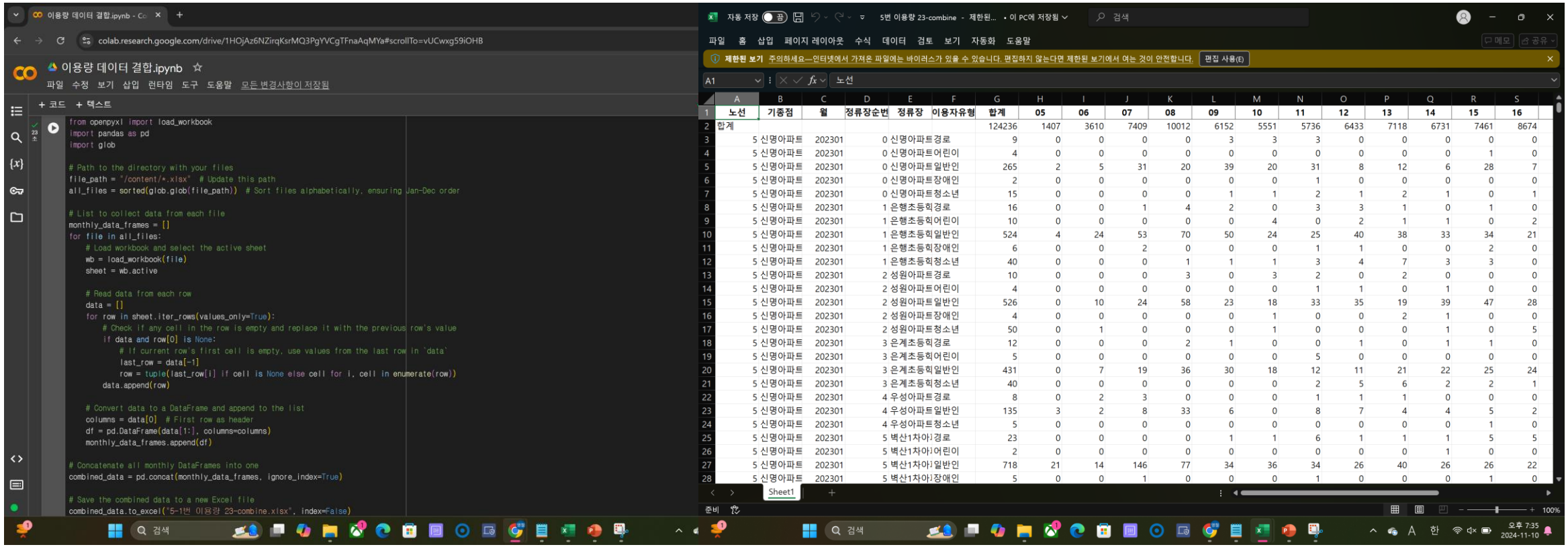
1. 데이터 수집 과정

- 각 노선별로 23년도 데이터 수집



2. 데이터 전처리 과정

- 수집한 이용량 데이터를 구글 코랩에서 하나의 데이터로 결합

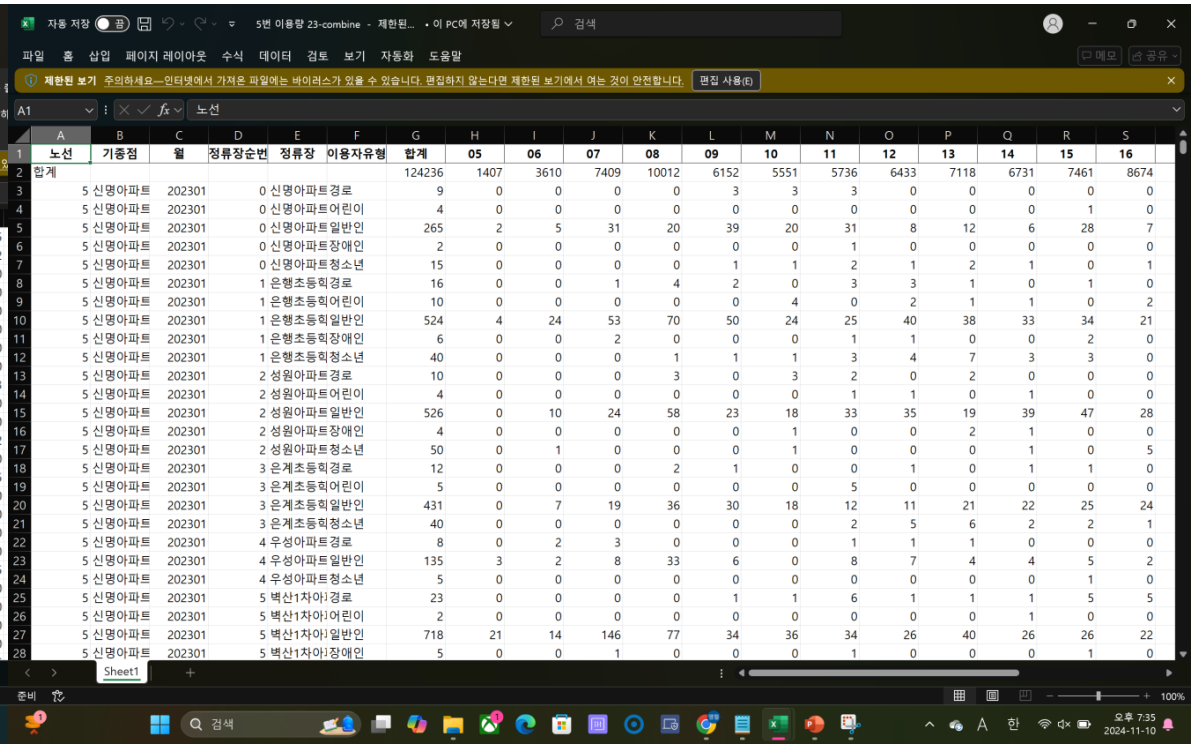


2. 데이터 전처리 과정

- 결합한 데이터의 '노선' 열의 값이 '합계'인 행을 찾아 해당 행의 '노선'열의 데이터는 null값으로 초기화하고, '이용자유형' 열의 데이터를 '합계'로 변경한 다음 나머지 null인 데이터를 bfill을 사용하여 채운다.

```python
#노선별 이용량 전처리
import pandas as pd
import numpy as np

# Replace 'your_file.csv' with the path to your CSV file
file_path = "/content/노선별 이용량 23-combine.csv"

# Load the CSV file into a pandas DataFrame
df = pd.read_csv(file_path, encoding='utf-8-sig')

# Replace this list with the column names you want to check
columns_to_check = ["노선", "기종점", "월", "정류장순번", "정류장", "이용자유형"]

# Extract rows where any of the specified columns have null values
rows_with_nulls_in_columns = df[df[columns_to_check].isnull().any(axis=1)]

index_numbers_with_nulls = rows_with_nulls_in_columns.index

# Print the index numbers
print(f"Index numbers of rows with null values in columns {columns_to_check}:")
print(index_numbers_with_nulls)

df['노선'] = df['노선'].replace('합계', np.nan)

df.loc[index_numbers_with_nulls, '이용자유형'] = '합계'

df[columns_to_check] = df[columns_to_check].fillna(method='bfill')

df.fillna(0)

df.to_csv('노선별 이용량 전처리 23-combine.csv', index=False, encoding='utf-8-sig')
```

노선	기종점	월	정류장순번	정류장	이용자유형	합계
111-2	동춘동차고	202301	1	동춘동종점	합계	191551
111-2	동춘동차고	202301	1	동춘동종점	경로	1
111-2	동춘동차고	202301	1	동춘동종점	일반인	49
111-2	동춘동차고	202301	1	동춘동종점	청소년	25
111-2	동춘동차고	202301	2	무지개아파	경로	6
111-2	동춘동차고	202301	2	무지개아파	국가유공자	1
111-2	동춘동차고	202301	2	무지개아파	어린이	8
111-2	동춘동차고	202301	2	무지개아파	일반인	611
111-2	동춘동차고	202301	2	무지개아파	장애인	33
111-2	동춘동차고	202301	2	무지개아파	청소년	31
111-2	동춘동차고	202301	3	동춘2동행	경로	27
111-2	동춘동차고	202301	3	동춘2동행	어린이	7
111-2	동춘동차고	202301	3	동춘2동행	일반인	445
111-2	동춘동차고	202301	3	동춘2동행	장애인	11
111-2	동춘동차고	202301	3	동춘2동행	청소년	42
111-2	동춘동차고	202301	4	한양2차아	경로	10
111-2	동춘동차고	202301	4	한양2차아	어린이	1
111-2	동춘동차고	202301	4	한양2차아	일반인	616
111-2	동춘동차고	202301	4	한양2차아	청소년	46
111-2	동춘동차고	202301	5	나사랫국저	경로	92
111-2	동춘동차고	202301	5	나사랫국저	국가유공자	3
111-2	동춘동차고	202301	5	나사랫국저	어린이	38



노선	기종점	월	정류장순번	정류장	이용자유형	합계	05	06	07	08	09	10	11	12	13	14	15	16
합계						124236	1407	3610	7409	10012	6152	5551	5736	6433	7118	6731	7461	8674
5	신명아파트	202301	0	신명아파트	경로	9	0	0	0	0	3	3	3	0	0	0	0	0
5	신명아파트	202301	0	신명아파트	어린이	4	0	0	0	0	0	0	0	0	0	0	1	0
5	신명아파트	202301	0	신명아파트	일반인	265	2	5	31	20	39	20	31	8	12	6	28	7
5	신명아파트	202301	0	신명아파트	장애인	2	0	0	0	0	0	0	1	0	0	0	0	0
5	신명아파트	202301	0	신명아파트	청소년	15	0	0	0	0	1	1	2	1	2	1	0	1
5	신명아파트	202301	1	은행초등학	경로	16	0	0	1	4	2	0	3	3	1	0	1	0
5	신명아파트	202301	1	은행초등학	어린이	10	0	0	0	0	0	4	0	2	1	1	0	2
5	신명아파트	202301	1	은행초등학	일반인	524	4	24	53	70	50	24	25	40	38	33	34	21
5	신명아파트	202301	1	은행초등학	장애인	6	0	0	2	0	0	0	1	1	0	0	2	0
5	신명아파트	202301	1	은행초등학	청소년	40	0	0	0	1	1	1	3	4	7	3	3	0
5	신명아파트	202301	2	성원아파트	경로	10	0	0	0	3	0	3	2	0	2	0	0	0
5	신명아파트	202301	2	성원아파트	어린이	4	0	0	0	0	0	0	1	1	0	1	0	0
5	신명아파트	202301	2	성원아파트	일반인	526	0	10	24	58	23	18	33	35	19	39	47	28
5	신명아파트	202301	2	성원아파트	장애인	4	0	0	0	0	0	1	0	0	2	1	0	0
5	신명아파트	202301	2	성원아파트	청소년	50	0	1	0	0	0	1	0	0	0	1	0	5
5	신명아파트	202301	3	은계초등학	경로	12	0	0	0	2	1	0	0	1	0	1	1	0
5	신명아파트	202301	3	은계초등학	어린이	5	0	0	0	0	0	0	5	0	0	0	0	0
5	신명아파트	202301	3	은계초등학	일반인	431	0	7	19	36	30	18	12	11	21	22	25	24
5	신명아파트	202301	3	은계초등학	청소년	40	0	0	0	0	0	0	2	5	6	2	2	1
5	신명아파트	202301	4	우성아파트	경로	8	0	2	3	0	0	0	1	1	1	0	0	0
5	신명아파트	202301	4	우성아파트	일반인	135	3	2	8	33	6	0	8	7	4	4	5	2
5	신명아파트	202301	4	우성아파트	청소년	5	0	0	0	0	0	0	0	0	0	0	1	0
5	신명아파트	202301	5	벽산1차아	경로	23	0	0	0	0	1	1	6	1	1	1	5	5
5	신명아파트	202301	5	벽산1차아	어린이	2	0	0	0	0	0	0	0	0	0	1	0	0
5	신명아파트	202301	5	벽산1차아	일반인	718	21	14	146	77	34	36	34	26	40	26	26	22
5	신명아파트	202301	5	벽산1차아	장애인	5	0	0	1	0	0	0	1	0	0	0	1	0

2. 데이터 전처리 과정

- 노선, 이용자유형, 월별 '합계' 열과 시간열의 이용량이 가장 많은 데이터를 추출 후 결합

```python
#노선, 이용자유형, 월별 '합계' 열의 이용량이 가장 많은 데이터 추출
import pandas as pd
import numpy as np

file_path = "/content/노선별 이용량 전처리 23-combine.csv"

df = pd.read_csv(file_path, encoding='utf-8-sig', dtype={'노선': str})

df_g = df.loc[df.groupby(['노선', '이용자유형', '월'])['합계'].idxmax()]

df_g.to_csv('노선_이용자유형_월_max_합계.csv', index=False, encoding='utf-8-sig')
```

```python
#노선, 이용자유형, 월별 특정 시간대의 이용량이 가장 많은 데이터 추출
import pandas as pd
import numpy as np

file_path = "/content/노선별 이용량 전처리 23-combine.csv"

df = pd.read_csv(file_path, encoding='utf-8-sig', dtype={'노선': str})

for month in range(5, 24):
    month_str = f"{month:02}"

    if month_str in df.columns:
        df_g = df.loc[df.groupby(['노선', '이용자유형', '월'])[month_str].idxmax()]
        df_g[df_g['이용자유형'] == '합계'].drop

        output_file = f"노선_이용자유형_월_max_{month_str}.csv"
        df_g.to_csv(output_file, index=False, encoding='utf-8-sig')
        print(f"Saved file for column '{month_str}' to '{output_file}'")
    else:
        print(f"Column '{month_str}' does not exist in the DataFrame. Skipping...")
```

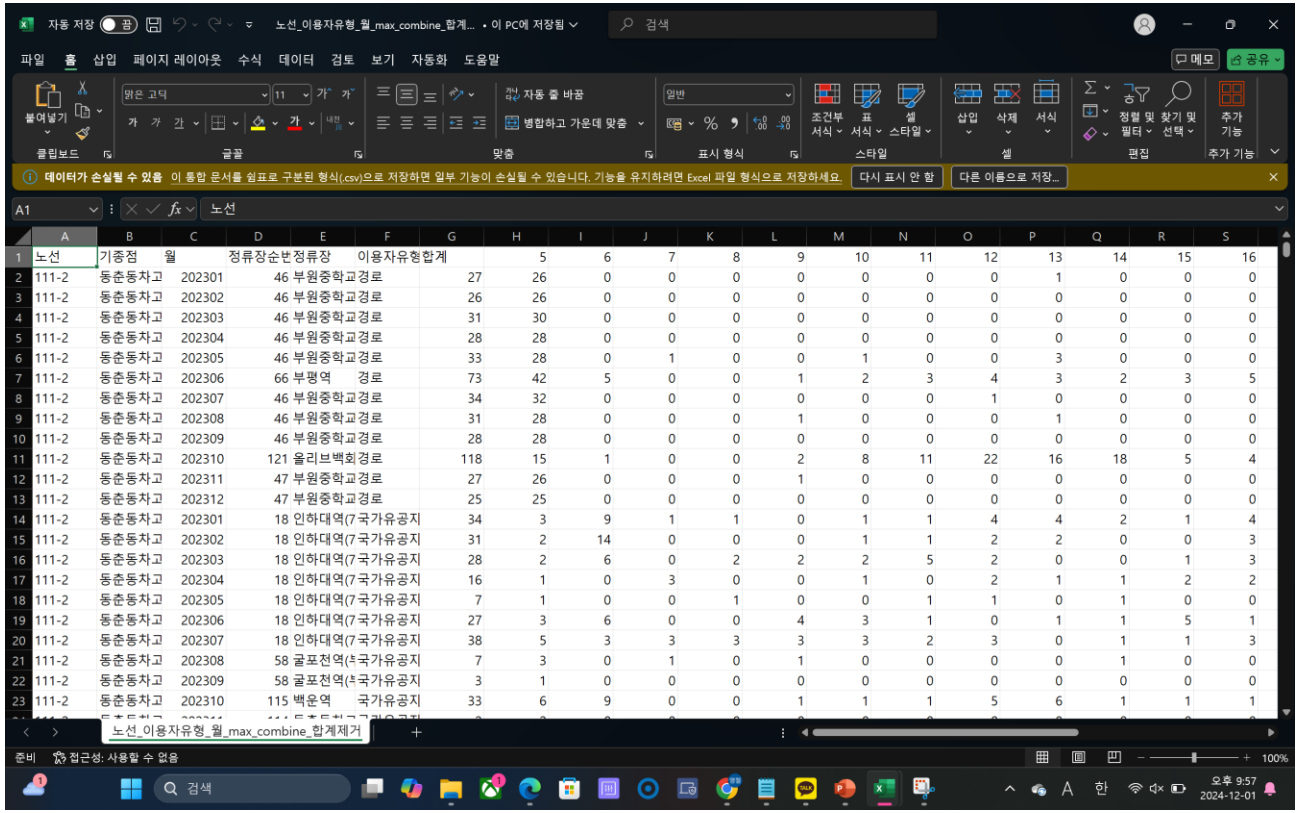
```python
#'이용자유형' 열의 값이 '합계'인 행 제거
import pandas as pd
import numpy as np

file_path = "/content/노선_이용자유형_월_max_combine.csv"

df = pd.read_csv(file_path, encoding='utf-8-sig')

df = df[df['이용자유형'] != '합계']

df.to_csv('노선_이용자유형_월_max_combine_합계제거.csv', index=False, encoding='utf-8-sig')
```

```python
#노선, 이용자유형, 월별 이용량이 가장 많은 데이터 합산
from openpyxl import load_workbook
import pandas as pd
import glob


file_path = "/content/노선_이용자유형_월_max_*.csv"
all_files = sorted(glob.glob(file_path))

df_list = []

for file in all_files:
    df = pd.read_csv(file, encoding='utf-8-sig')
    df_list.append(df)

combined_df = pd.concat(df_list, ignore_index=True)

combined_df.to_csv('노선_이용자유형_월_max_combine.csv', index=False, encoding='utf-8-sig')
```

2. 데이터 전처리 과정

- 결합한 데이터의 '이용자유형'열의 값이 '합계'인 행을 제거한다.

```python
#'이용자유형' 열의 값이 '합계'인 행 제거
import pandas as pd
import numpy as np

file_path = "/content/노선_이용자유형_월_max_combine.csv"

df = pd.read_csv(file_path, encoding='utf-8-sig')

df = df[df['이용자유형'] != '합계']

df.to_csv('노선_이용자유형_월_max_combine_합계제거.csv', index=False, encoding='utf-8-sig')
```

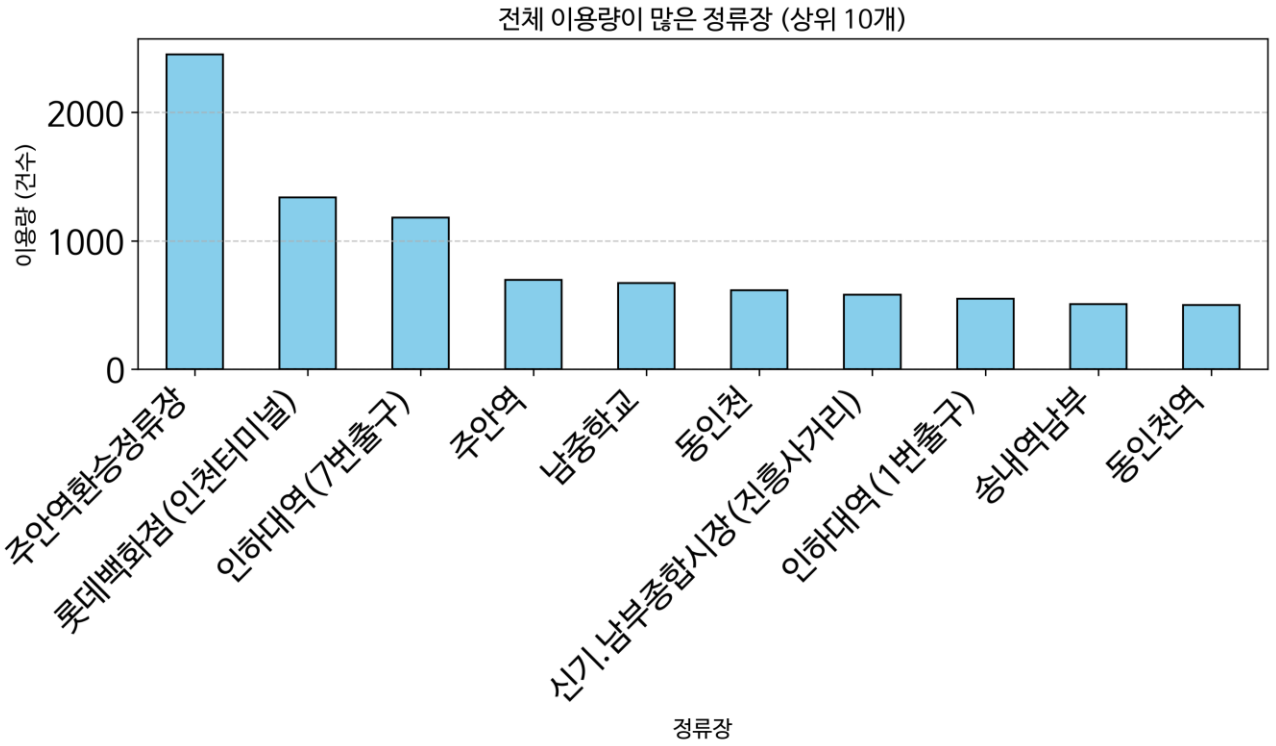


노선	기종점	월	정류장순번	정류장	이용자유형	합계	5	6	7	8	9	10	11	12	13	14	15	16
111-2	동춘동차고	202301	46	부원중학교	경로	27	26	0	0	0	0	0	0	0	1	0	0	0
111-2	동춘동차고	202302	46	부원중학교	경로	26	26	0	0	0	0	0	0	0	0	0	0	0
111-2	동춘동차고	202303	46	부원중학교	경로	31	30	0	0	0	0	0	0	0	0	0	0	0
111-2	동춘동차고	202304	46	부원중학교	경로	28	28	0	0	0	0	0	0	0	0	0	0	0
111-2	동춘동차고	202305	46	부원중학교	경로	33	28	0	1	0	0	1	0	0	3	0	0	0
111-2	동춘동차고	202306	66	부평역	경로	73	42	5	0	0	1	2	3	4	3	2	3	5
111-2	동춘동차고	202307	46	부원중학교	경로	34	32	0	0	0	0	0	0	1	0	0	0	0
111-2	동춘동차고	202308	46	부원중학교	경로	31	28	0	0	0	0	1	0	0	1	0	0	0
111-2	동춘동차고	202309	46	부원중학교	경로	28	28	0	0	0	0	0	0	0	0	0	0	0
111-2	동춘동차고	202310	121	올리브백회	경로	118	15	1	0	0	2	8	11	22	16	18	5	4
111-2	동춘동차고	202311	47	부원중학교	경로	27	26	0	0	0	1	0	0	0	0	0	0	0
111-2	동춘동차고	202312	47	부원중학교	경로	25	25	0	0	0	0	0	0	0	0	0	0	0
111-2	동춘동차고	202301	18	인하대역(7	국가유공지	34	3	9	1	1	0	1	1	4	4	2	1	4
111-2	동춘동차고	202302	18	인하대역(7	국가유공지	31	2	14	0	0	0	1	1	2	2	0	0	3
111-2	동춘동차고	202303	18	인하대역(7	국가유공지	28	2	6	0	2	2	2	5	2	0	0	1	3
111-2	동춘동차고	202304	18	인하대역(7	국가유공지	16	1	0	3	0	0	1	0	2	1	1	2	2
111-2	동춘동차고	202305	18	인하대역(7	국가유공지	7	1	0	0	1	0	0	1	1	0	1	0	0
111-2	동춘동차고	202306	18	인하대역(7	국가유공지	27	3	6	0	0	4	3	1	0	1	1	5	1
111-2	동춘동차고	202307	18	인하대역(7	국가유공지	38	5	3	3	3	3	3	2	3	0	1	1	3
111-2	동춘동차고	202308	58	굴포천역(5	국가유공지	7	3	0	1	0	1	0	0	0	0	1	0	0
111-2	동춘동차고	202309	58	굴포천역(5	국가유공지	3	1	0	0	0	0	0	0	0	0	0	0	0
111-2	동춘동차고	202310	115	백운역	국가유공지	33	6	9	0	0	1	1	1	5	6	1	1	1

3. 데이터 시각화 과정

- 전처리한 데이터의 정류장의 수를 count하여 전체 이용량이 많은 상위 10개 정류장을 시각화한다.

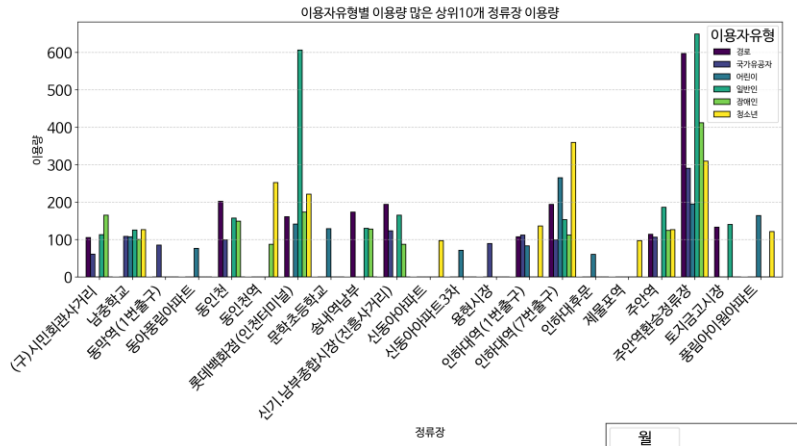
```python
#이용량 많은 상위 10개 정류장 이용량 시각화
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from google.colab import files

file_path = "/content/노선_이용자유형_월_max_combine_합계제거.csv"

df = pd.read_csv(file_path, encoding='utf-8-sig')

#전체 정류장 수count
df_g = df['정류장'].value_counts()

#데이터 시각화
#데이터의 상위10개 데이터를 그래프로 시각화
top_10 = df_g.head(10)  # 상위 10개 추출
plt.figure(figsize=(10, 6))  # 그래프 크기 설정
top_10.plot(kind='bar', color='skyblue', edgecolor='black')
plt.title(f"전체 이용량이 많은 정류장 (상위 10개)", fontsize=14)
plt.xlabel("정류장", fontsize=12)
plt.ylabel("이용량 (건수)", fontsize=12)
plt.xticks(rotation=45, ha='right')  # x축 라벨 회전
plt.tight_layout()  # 그래프 여백 조정
plt.grid(axis='y', linestyle='--', alpha=0.7)

#그래프 저장
graph_file = f"전체 이용량 많은 정류장.png"
plt.savefig(graph_file, dpi=300)
plt.close()

#그래프 다운로드
files.download(graph_file)

df_g.to_csv('이용량 많은 정류장 count.csv', index=True, encoding='utf-8-sig')
```

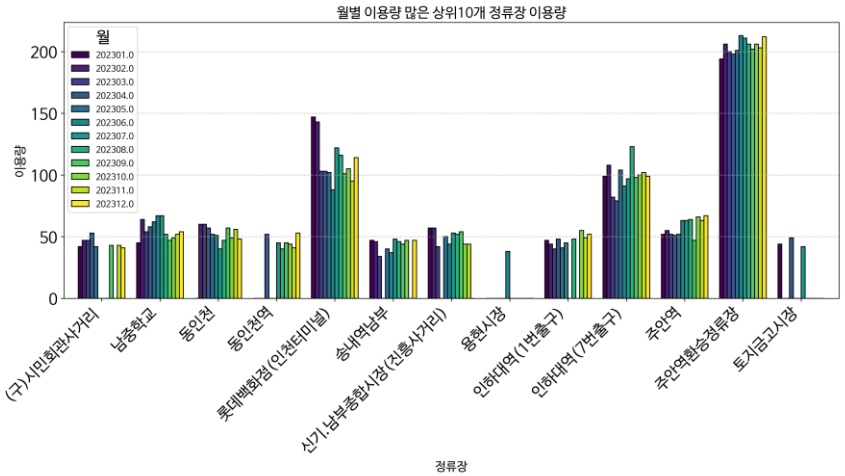


3. 데이터 시각화 과정

- 전처리한 데이터를 월별 또는 이용자유형 별로 그룹화하여 정류장의 수를 count하여 전체 이용량이 많은 상위 10개 정류장을 시각화한다.

```python
#월별, 이용자유형별 이용량 많은 상위10개 정류장 이용량 시각화
import pandas as pd
import matplotlib.pyplot as plt
from google.colab import files

# File path
file_path = "/content/노선_이용자유형_월_max_combine_합계제거.csv"

# Load data
df = pd.read_csv(file_path, encoding='utf-8-sig')

# 월별 정류장 수count
#df_g = df.groupby(['월'])['정류장'].value_counts().reset_index(name='count')

#이용자유형별 정류장 수count
#df_g = df.groupby(['이용자유형'])['정류장'].value_counts().reset_index(name='count')

#월별, 이용자유형별로 그룹을 만들어 상위10개 데이터 추출
top_5_per_month = df_g.groupby('노선').apply(lambda x: x.nlargest(10, 'count')).reset_index(drop=True)

# Pivot the data for grouped bar chart
pivot_data = top_5_per_month.pivot(index='정류장', columns='노선', values='count').fillna(0)

# Plot grouped bar chart
pivot_data.plot(kind='bar', figsize=(14, 8), width=0.8, colormap='viridis', edgecolor='black')
# Set plot styles
plt.title("노선별 이용량 많은 상위10개 정류장 이용량", fontsize=16)
plt.xlabel("정류장", fontsize=14)
plt.ylabel("이용량", fontsize=14)
plt.xticks(rotation=45, ha='right')
plt.legend(title="노선", fontsize=10)
plt.tight_layout()
plt.grid(axis='y', linestyle='--', alpha=0.7)

# Save and download the graph
graph_file = "노선별 이용량 많은 상위10개 정류장 이용량.png"
plt.savefig(graph_file, dpi=300)
plt.close()
files.download(graph_file)

# Save and download the top 5 stops per month as a CSV
output_csv = "노선별 이용량 많은 상위10개 정류장 이용량.csv"
top_5_per_month.to_csv(output_csv, index=False, encoding='utf-8-sig')
files.download(output_csv)
```

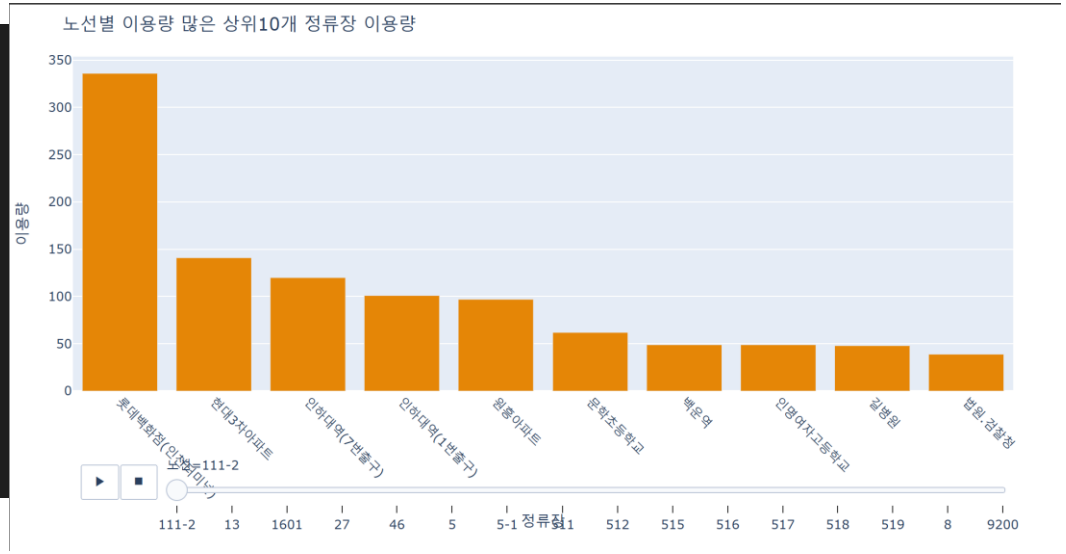




3. 데이터 시각화 과정

- 전처리한 데이터를 노선별로 그룹화하여 정류장의 수를 count하여 노선별로 전체 이용량이 많은 상위 10개 정류장을 시각화한다.

```python
#노선별 이용량 많은 상위10개 정류장 이용량 시각화
import pandas as pd
import matplotlib.pyplot as plt
from google.colab import files
import plotly.express as px

# File path
file_path = "/content/노선_이용자유형_월_max_combine_합계제거.csv"

# Load data
df = pd.read_csv(file_path, encoding='utf-8-sig')

#노선별 정류장 수count
df_g = df.groupby(['노선'])['정류장'].value_counts().reset_index(name='count')

#노선별로 그룹을 만들어 상위10개 데이터 추출
top_10_per_route = df_g.groupby('노선').apply(lambda x: x.nlargest(10, 'count')).reset_index(drop=True)

# Create the interactive plot
fig = px.bar(
    top_10_per_route,
    x="정류장",
    y="count",
    color="노선",
    animation_frame="노선",  # Adds a dropdown to select the route
    title="노선별 이용량 많은 상위10개 정류장 이용량",
    labels={"count": "이용량", "정류장": "정류장", "노선": "노선"},
    color_discrete_sequence=px.colors.qualitative.Vivid
)

# Customize layout
fig.update_layout(
    xaxis=dict(title="정류장", tickangle=45),
    yaxis=dict(title="이용량"),
    legend=dict(title="노선"),
    margin=dict(l=50, r=50, t=50, b=50),
    showlegend=False,
)

# Show the plot
fig.show()

# Save as an interactive HTML file
fig.write_html("노선별 이용량 많은 상위10개 정류장 이용량.html")
files.download("노선별 이용량 많은 상위10개 정류장 이용량.html")  # Download the HTML file

# Save the data to CSV
output_csv = "노선별 이용량 많은 상위10개 정류장 이용량.csv"
top_5_per_month.to_csv(output_csv, index=False, encoding='utf-8-sig')
files.download(output_csv)
```

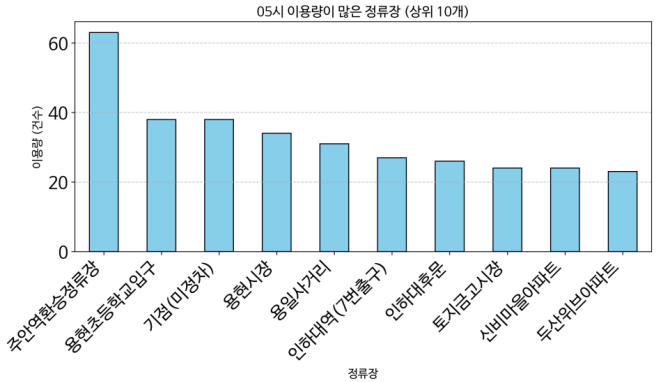


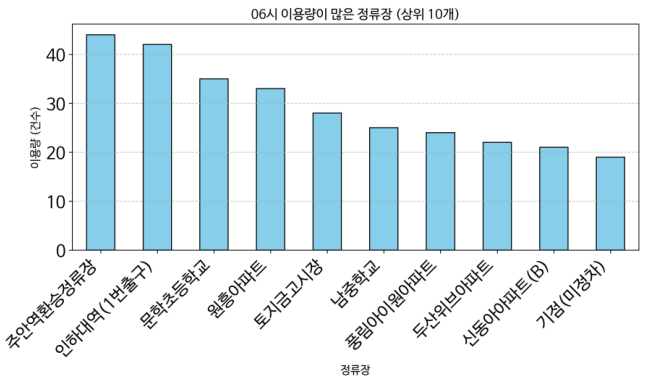
3. 데이터 시각화 과정

- 전처리한 데이터를 '노선', '이용자유형', '월'별로 그룹화하고, for문을 통해 시간열의 최대값을 구한 각각의 데이터의 정류장의 수를 count하여 상위 10개의 데이터를 그래프로 시각화한다.

```python
#노선, 이용자유형, 월별 특정 시간대의 이용량이 가장 많은 데이터 추출
#추출한 데이터의 정류장 수count한 데이터 추출
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from google.colab import files

file_path = "/content/노선별 이용량 전처리 23-combine.csv"

df = pd.read_csv(file_path, encoding='utf-8-sig', dtype={'노선': str})

for time in range(5, 24):
    time_str = f"{time:02}"

    if time_str in df.columns:
        df_g = df.loc[df.groupby(['노선', '이용자유형', '월'])[time_str].idxmax()]
        df_g = df_g[df_g['이용자유형'] != '합계']

        #전체 정류장 수count
        df_g = df_g['정류장'].value_counts()

        #시간별 데이터의 상위10개 데이터를 그래프로 시각화
        top_10 = df_g.head(10)  # 상위 10개 추출
        plt.figure(figsize=(10, 6))  # 그래프 크기 설정
        top_10.plot(kind='bar', color='skyblue', edgecolor='black')
        plt.title(f"{time_str}시 이용량이 많은 정류장 (상위 10개)", fontsize=14)
        plt.xlabel("정류장", fontsize=12)
        plt.ylabel("이용량 (건수)", fontsize=12)
        plt.xticks(rotation=45, ha='right')  # x축 라벨 회전
        plt.tight_layout()  # 그래프 여백 조정
        plt.grid(axis='y', linestyle='--', alpha=0.7)
```

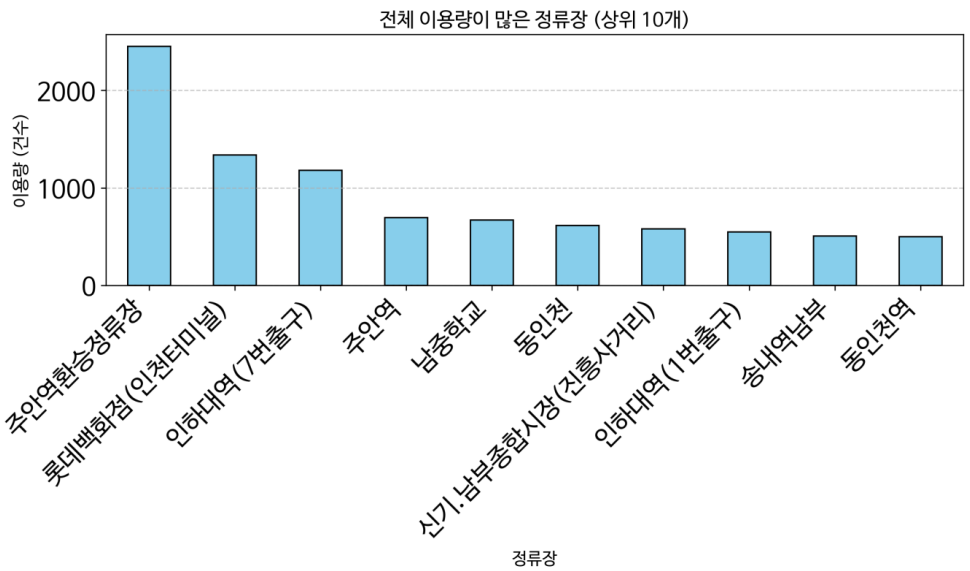
```python
    #그래프 저장
    graph_file = f"시간대별 이용량 많은 정류장_{time_str}.png"
    plt.savefig(graph_file, dpi=300)
    plt.close()
    print(f"Saved graph for column '{time_str}' to '{graph_file}'")

    #그래프 다운로드
    if(time >= 15):
      files.download(graph_file)

    output_file = f"시간대별 이용량 많은 정류장_{time_str}.csv"
    df_g.to_csv(output_file, index=True, encoding='utf-8-sig')
    print(f"Saved file for column '{time_str}' to '{output_file}'")
else:
    print(f"Column '{time_str}' does not exist in the DataFrame. Skipping...")
```

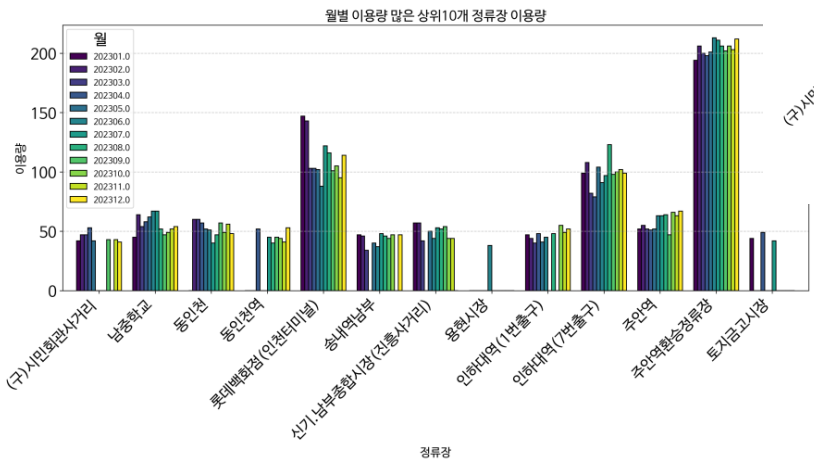


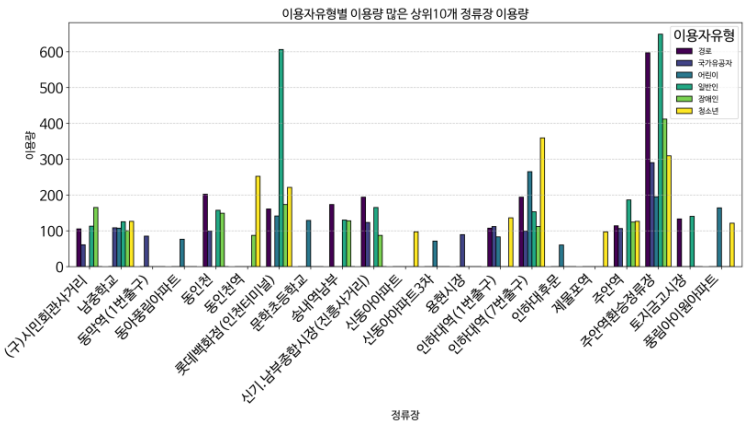


4. 데이터 분석

- 시각화한 데이터를 보면 주안역환승정류장과 롯데백화점(인천터미널)의 이용량이 가장 많으며 학교 주변의 정류장은 인하대역(1번출구), 인하대역(7번출구), 인하대후문이 있고, 월별 정류장 이용량은 크게 차이나지 않는다.
- 이용자유형별 데이터를 보면 일반인은 주안역환승정류장과 롯데백화점(인천터미널)의 이용량이 가장 많고, 경로, 국가유공자, 장애인은 주안역 환승정류장 이용량이 많고, 어린이는 인하대역(7번출구)이용량이 많고, 청소년은 인하대역(7번출구), 동인천역, 주안역환승정류장, 롯데백화점(인천터미널) 이용량이 많다.

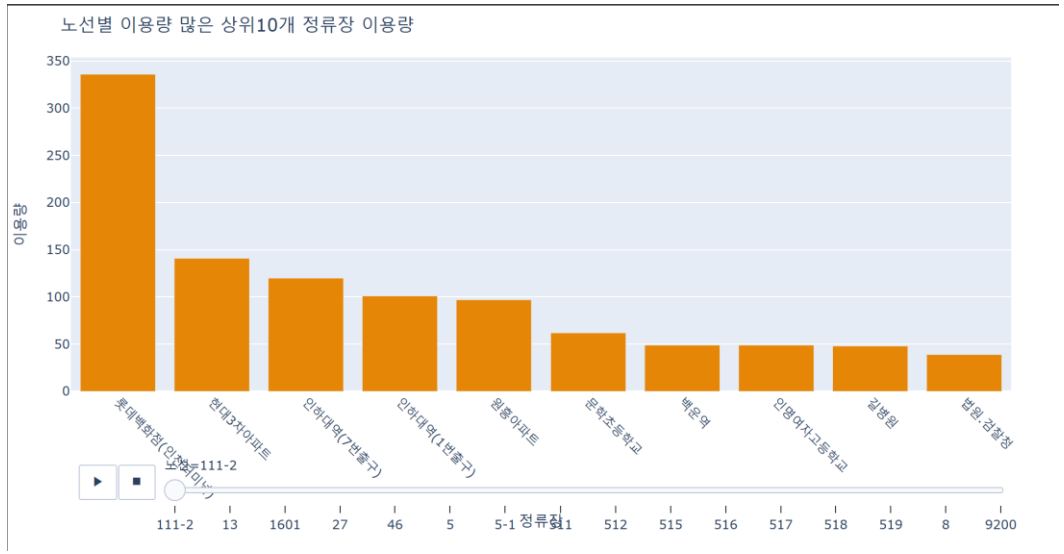


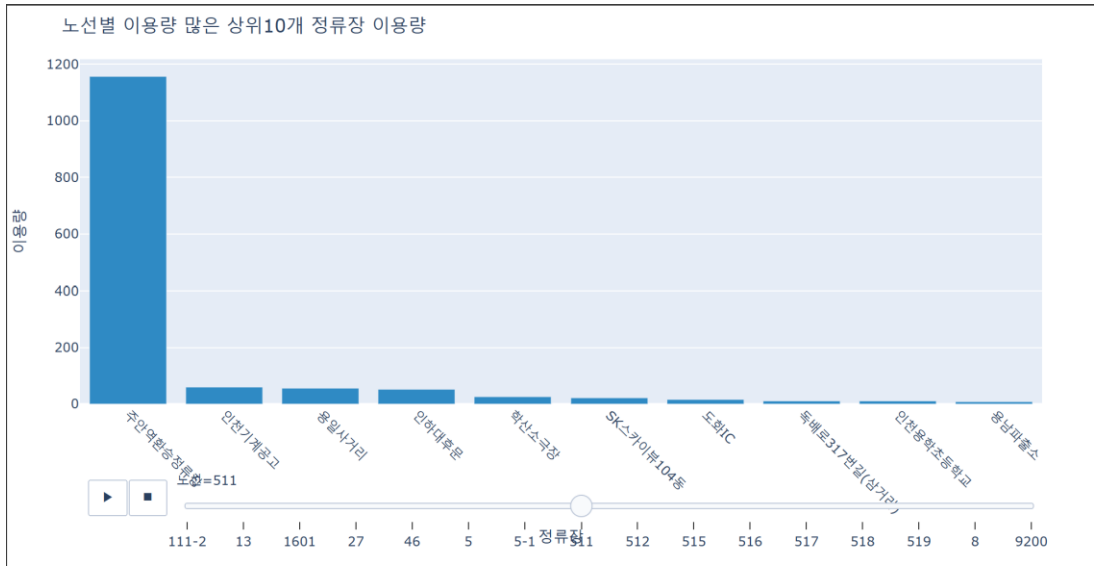




4. 데이터 분석

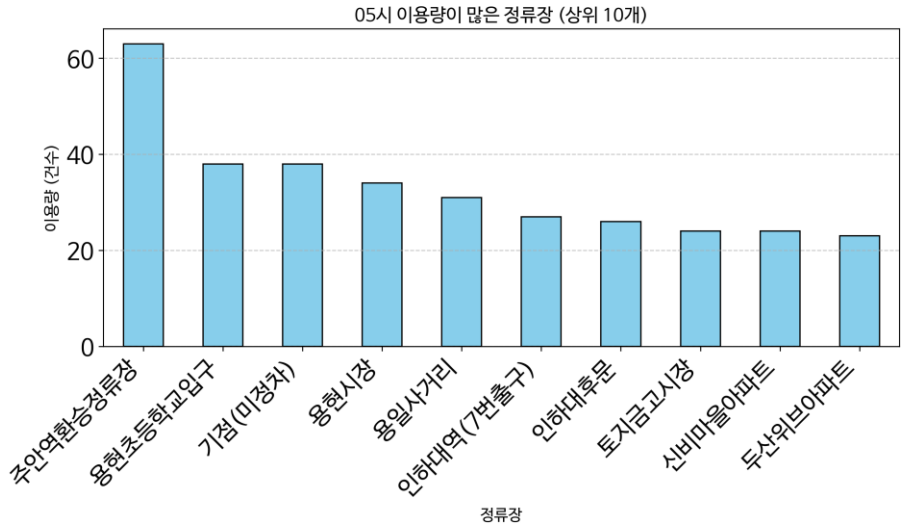
- 시각화한 데이터를 보면 511번 버스의 주안역환승정류장 이용량이 가장 많고, 이용량이 많은 정류장 중에 우리학교 주변 정류장은 인하대역(1번출구), 인하대역(7번출구), 장미아파트, 인하대후문, 정석항공과학고가 있다.

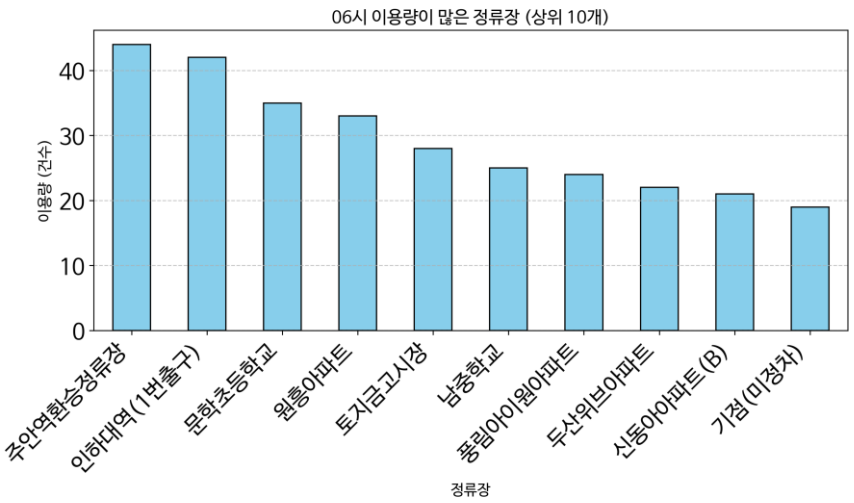




4. 데이터 분석

- 시간대별 시각화 자료를 보면 모든 시간대의 가장 많이 이용한 정류장은 주안역환승정류장이며, 인하대역(7번출구)정류장이 06시와 16시를 제외한 모든 시간대에 상위10정류장 안에 들어간다.
- 수업이 종료되는 13시에는 정석항공과학고 정류장 이용량도 늘어나지만, 그 이외의 학교주변 정류장은 없다.





5. 결론

- 우리학교 주변을 지나는 버스는 주안역환승정류장과 롯데백화점(인천터미널) 정류장의 이용량이 많으며, 인하대역(1번출구)와 인하대역(7번출구)의 이용량이 많아 해당 정류장의 이용량이 많은 시간대에는 학교주변의 정류장 또한 내리지 않은 사람들로 인해 붐빌 수 있다.
- 우리 학교 주변 정류장에서는 일반인 이외에는 인하대역(7번출구)에서 청소년과 어린이 이용량이 가장 많고, 경로, 국가유공자, 장애인은 주안역환승정류장 외에는 이용량이 비교적 저조하다.
- 학교주변에서 버스 이용시에 인하대역에서 오는 버스 이용시 혼잡할 수 있다는 것을 인지하고, 인하대역 쪽으로 가는 버스 또한 인하대역에서 사람이 많이 탈 수 있다는 것을 인지하면 좋을것이다.
- 주안역환승정류장쪽에서 오는 버스 탑승시에는 노약자들이 탈 가능성이 높기 때문에 노약자석을 비우고 자리를 양보해야 할 수도 있다는 것을 인지하면 좋다.