

미니프로젝트 5차

데이터 분석 종합

# 119 병원선정 솔루션 기획

Day4

Day4

소통을 위한 약속

카메라는 'ON'

마이크는 'OFF'

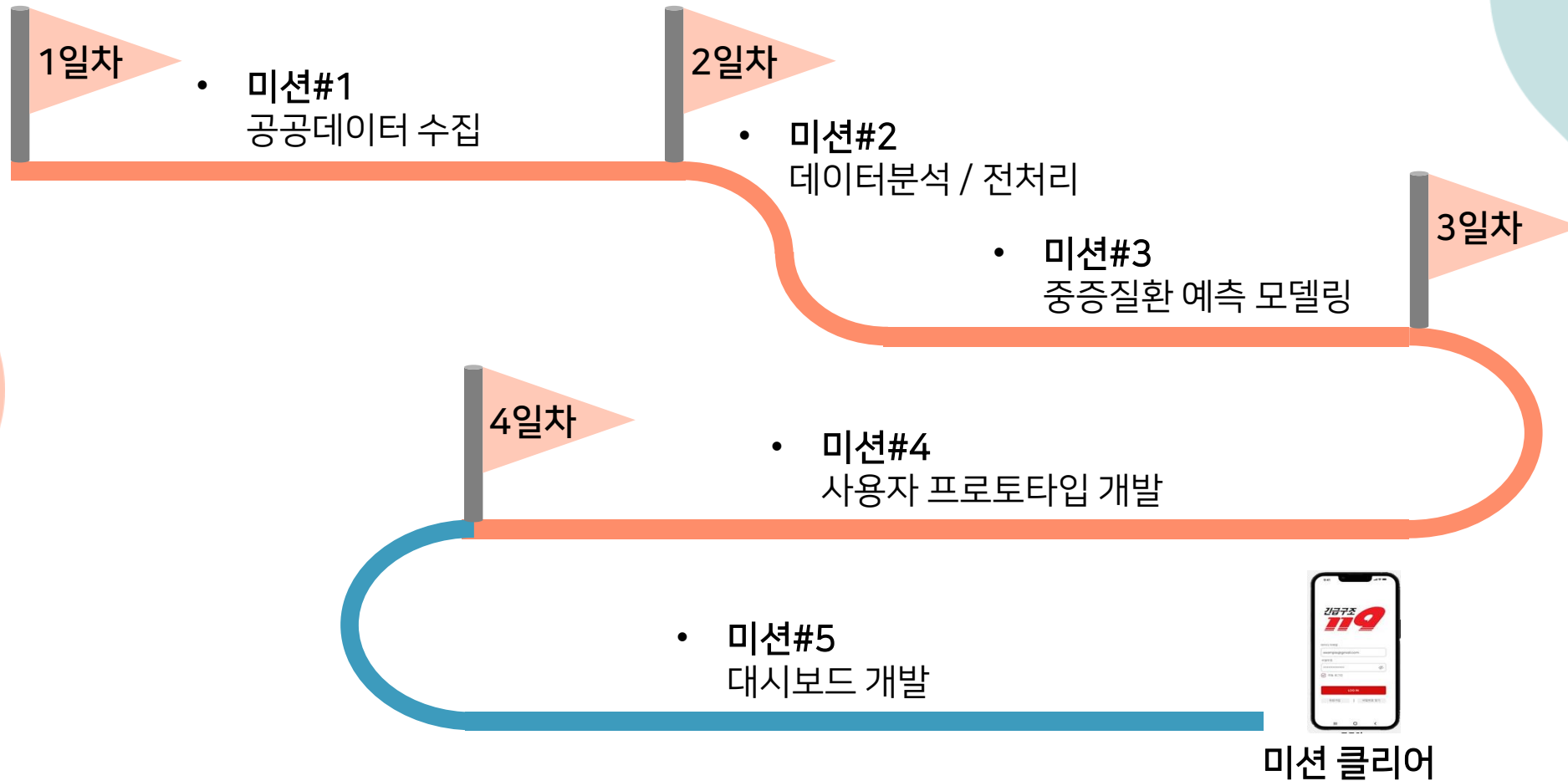


비대면 교육의 특성상

소통을 위한  
약속이 중요합니다.

Day4

## 프로젝트 수행절차



## Day4

# 미니프로젝트 개요

## 미니프로젝트 5차 Day4 미션

**메인목표: 응급환자의 빠른 병원 이송을 위한 119 구급대원의 병원선정 솔루션 프로토타입 제작**

미니프로젝트 모듈	일일 목표	일일 미션	실습 난이도
미니프로젝트 1차 데이터 전처리	데이터 확보 공공데이터 수집 및 가공	Day.1 API 데이터 수집	Level.1- 중급
미니프로젝트 2차 데이터 분석	최적 모델 만들기 Feature 특성 이해 및 최적 모델 찾기	Day.2 데이터분석및 모델링	
미니프로젝트 3차 머신러닝	프로토타입 개발 사용자 필요 정보 입력 분석 정보 출력	Day.3 프로토타입 개발	Level.2- 고급
미니프로젝트 4차 딥러닝	대시보드 개발 구급출동 데이터의 통계 정보 제시	Day.4 대시보드 개발	
미니프로젝트 5차 데이터 분석 종합			
미니프로젝트 6차 IT 인프라 설계			
미니프로젝트 7차 제안전략/PT			

Level.1- 중급

미션 안내  
가이드 코드 제공  
프로토타입 UI 기획

Level.2- 고급

미션 안내  
프로토타입 UI 기획  
도전미션 수행

## Day4

## 프로젝트 일정

## 4일차 | 대시보드 개발

구분	활동	시간	실습	강의장	내용
1	미니프로젝트 소개	9:40~10:30 (50분)	개인	전체	프로젝트 소개, 수행 방법 안내
2	미션5 개인실습	10:40~11:30 (50분)	개인	전체	개인별 프로젝트 수행
3		11:40~12:30 (50분)	개인	전체	개인별 프로젝트 수행, 미션 중간점검
점심					
4	미션5 조별실습	13:30~14:30 (60분)	조	조별	조별 프로젝트 수행
5		14:40~15:40 (60분)	조	조별	조별 프로젝트 수행
6	Summary	15:50~16:40 (50분)	조	발표방	미션5 리뷰, 4일차 과정 Wrap-up
7	셀프테스트	16:50~17:10 (20분)	개인	테스트	셀프테스트, 설문조사



## *미니프로젝트 5차*

4일차 미션



# 미션5. 대시보드 개발

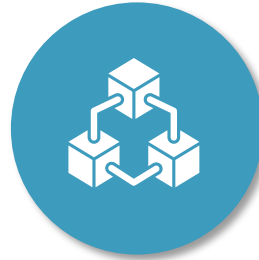
## Day4

# 대시보드 개발 순서



### 정보 정의

- 당일 출동이력
- 출동 건수
- 이송시간
- 중증질환 비율



### 통계 데이터 집계

- 일별 집계
- 주별 집계
- 월별 집계



### 화면 구성

- line\_chart
- bar\_chart
- py\_chart

# UI/UX 예제







# Day4

## 제작 Preview



# Dashboard

활용 라이브러리 및 함수 알아보기

## Day4

## 활용 라이브러리/함수

라이브러리/함수	용도
streamlit	웹 시각화 프레임워크
streamlit 함수1	<ul style="list-style-type: none"><li>• layout : tabs, columns, expander</li><li>• print : write, markdown, info, dataframe</li><li>• input widget : date_input, time_input, number_input, text_input, radio, checkbox, selectbox, slider</li><li>• form : form, submit_button</li><li>• 실행파일 생성: %%writefile dashboard.py</li></ul>
streamlit 함수2	<ul style="list-style-type: none"><li>• line_chart : 꺾은선 그래프 웹 출력</li><li>• bar_chart : 막대 그래프 웹 출력</li><li>• plotly_chart : plotly 라이브러리의 파이(도넛) 그래프 웹 출력</li></ul>
plotly	<ul style="list-style-type: none"><li>• 파이썬 데이터 시각화(차트 출력)</li></ul>

## Day4

## streamlit 함수 사용

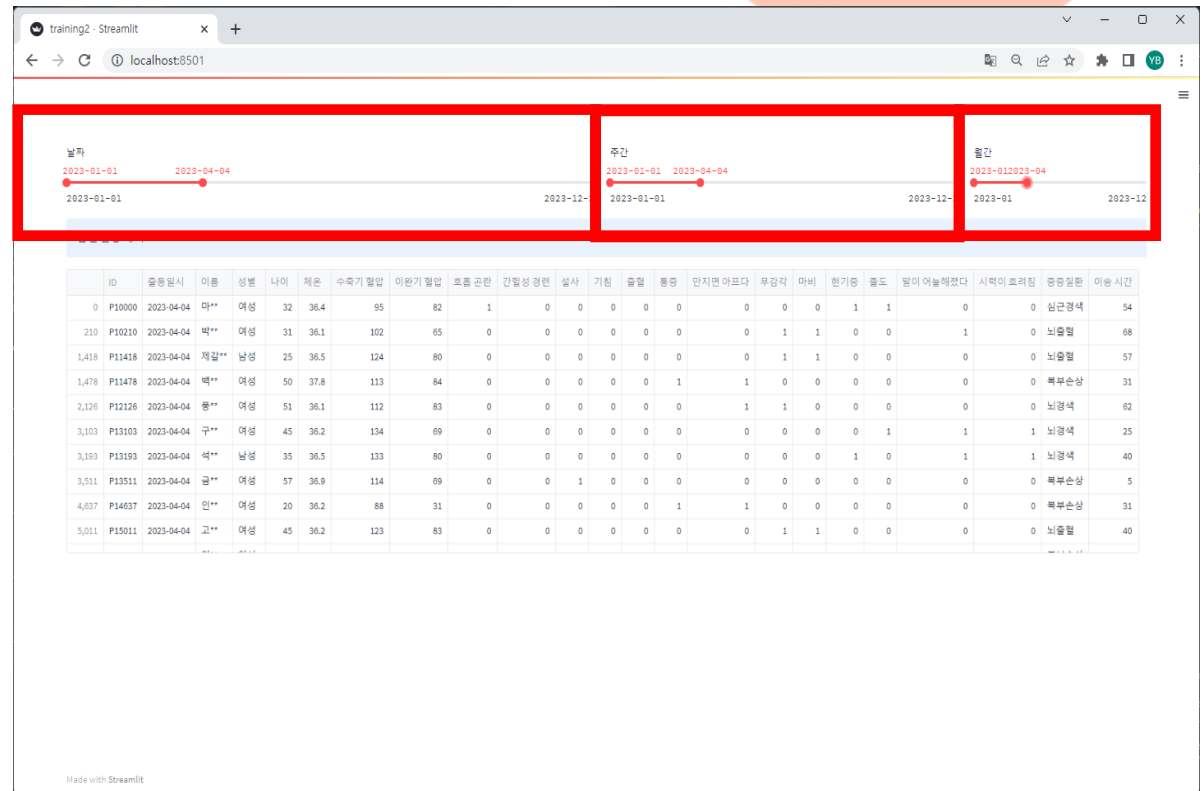
- columns, slider(date), dataframe

```
## 오늘 날짜
now_date = datetime.datetime.now(datetime.timezone.utc) + datetime.timedelta(hours=9)
now_date2 = datetime.datetime.strptime(now_date.strftime("%Y-%m-%d"), "%Y-%m-%d")

## 출동 이력의 최소 날짜, 최대 날짜
min_date = datetime.datetime.strptime("2023-1-1", "%Y-%m-%d")
max_date = datetime.datetime.strptime("2023-12-31", "%Y-%m-%d")
today_date = now_date.strftime("%Y-%m-%d")

st.set_page_config(layout="wide")
col210, col211, col212 = st.columns([0.3, 0.2, 0.1])
with col210:
    slider_date = st.slider("날짜", min_value=min_date, max_value=max_date,
                           value=(min_date, now_date2))
with col211:
    slider_week = st.slider("주간", min_value=min_date, max_value=max_date,
                           step=datetime.timedelta(weeks=1), value=(min_date, now_date2))
with col212:
    slider_month = st.slider("월간", min_value=min_date, max_value=max_date,
                           step=datetime.timedelta(weeks=1), value=(min_date, now_date2), format="%Y-%m-%d")

data = pd.read_csv("../119_emergency_dispatch.csv", encoding="cp949")
st.info("금일 출동 내역")
data['datetime'] = pd.to_datetime(data['출동일시'])
st.dataframe(data[(slider_date[0] <= data['datetime']) & (data['datetime'] <= slider_date[1])])
```



## Day4

## streamlit 함수 사용

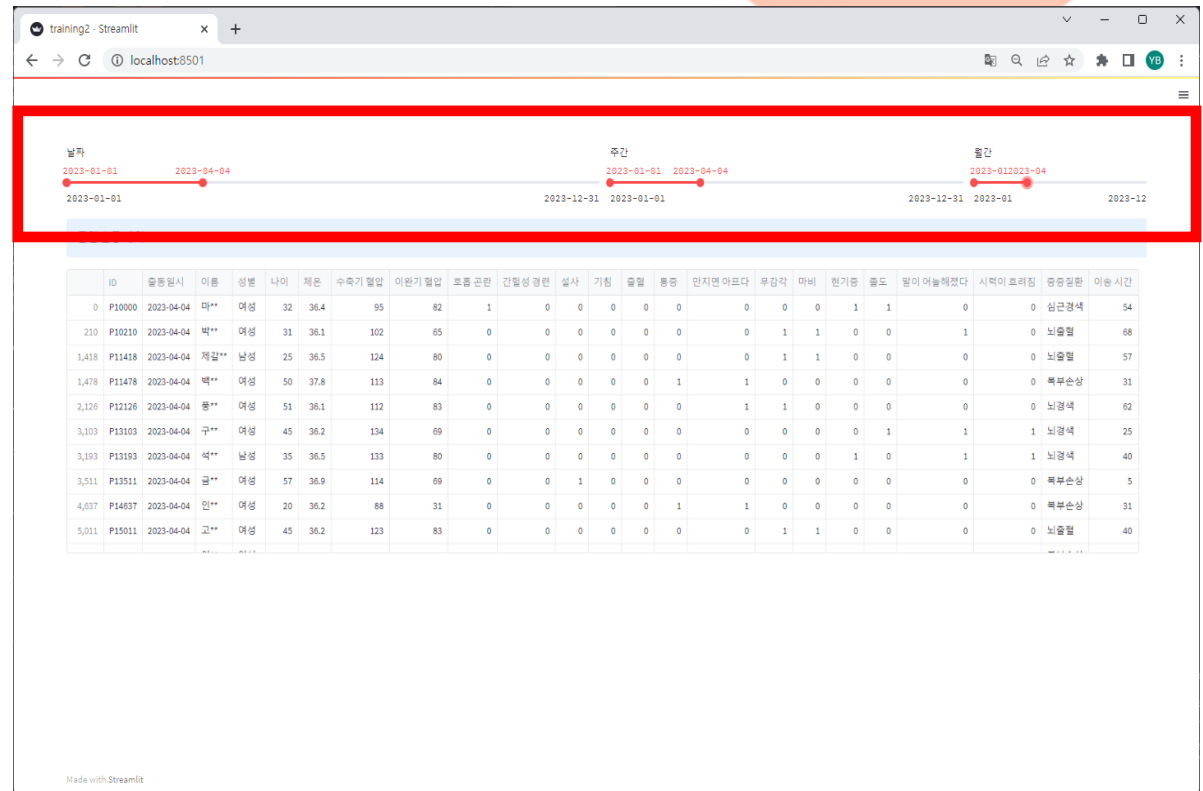
- columns, **slider(date)**, dataframe

```
## 오늘 날짜
now_date = datetime.datetime.now(datetime.timezone.utc) + datetime.timedelta(hours=9)
now_date2 = datetime.datetime.strptime(now_date.strftime("%Y-%m-%d"), "%Y-%m-%d")

## 출동 이력의 최소 날짜, 최대 날짜
min_date = datetime.datetime.strptime("2023-1-1", "%Y-%m-%d")
max_date = datetime.datetime.strptime("2023-12-31", "%Y-%m-%d")
today_date = now_date.strftime("%Y-%m-%d")

st.set_page_config(layout="wide")
col210, col211, col212 = st.columns([0.3, 0.2, 0.1])
with col210:
    slider_date = st.slider("날짜 ", min_value=min_date, max_value=max_date,
                           value=(min_date, now_date2))
with col211:
    slider_week = st.slider("주간", min_value=min_date, max_value=max_date,
                           step=datetime.timedelta(weeks=1), value=(min_date, now_date2))
with col212:
    slider_month = st.slider("월간", min_value=min_date, max_value=max_date,
                           step=datetime.timedelta(weeks=1), value=(min_date, now_date2), format="%Y-%m-%d")

data = pd.read_csv('./119_emergency_dispatch.csv', encoding="cp949")
st.info("금일 출동 내역")
data['datetime'] = pd.to_datetime(data['출동일시'])
st.dataframe(data[(slider_date[0] <= data['datetime']) & (data['datetime'] <= slider_date[1])])
```



## Day4

## streamlit 함수 사용

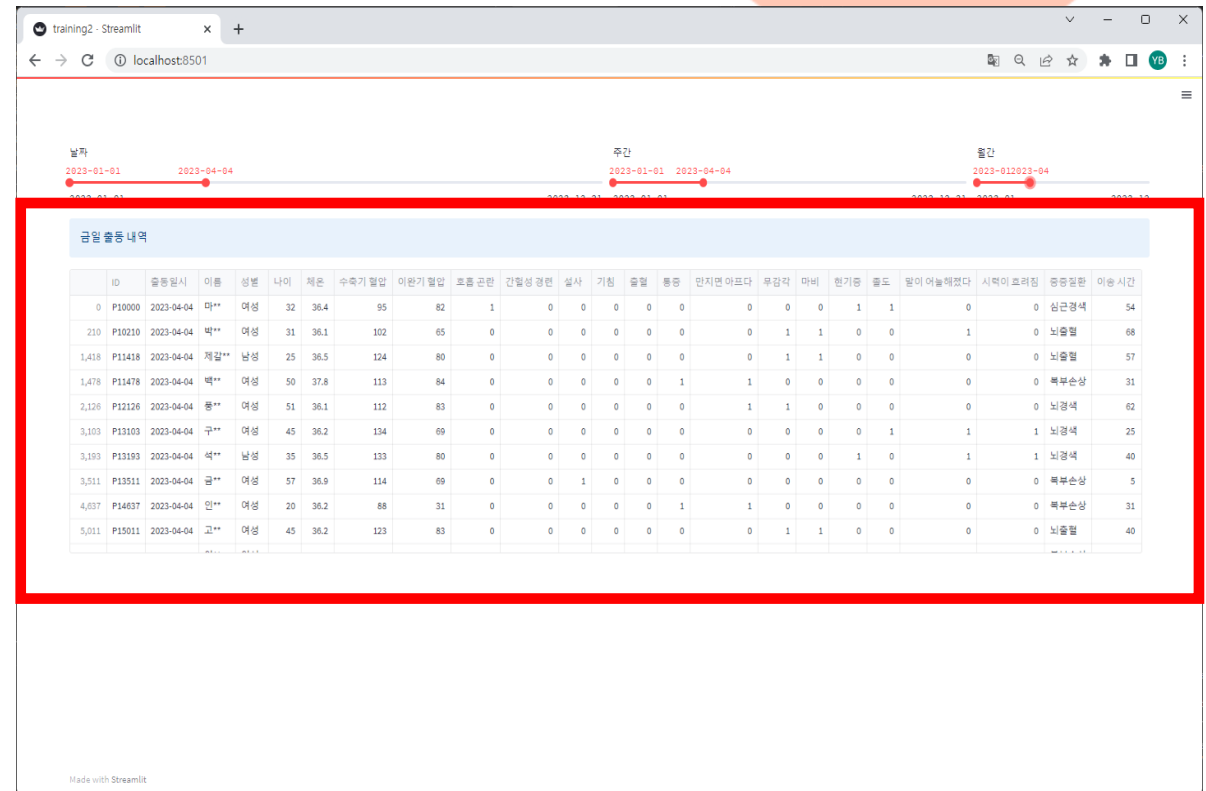
- columns, slider(date), dataframe

```
## 오늘 날짜
now_date = datetime.datetime.now(datetime.timezone.utc) + datetime.timedelta(hours=9)
now_date2 = datetime.datetime.strptime(now_date.strftime("%Y-%m-%d"), "%Y-%m-%d")

## 출동 이력의 최소 날짜, 최대 날짜
min_date = datetime.datetime.strptime("2023-1-1", "%Y-%m-%d")
max_date = datetime.datetime.strptime("2023-12-31", "%Y-%m-%d")
today_date = now_date.strftime("%Y-%m-%d")

st.set_page_config(layout="wide")
col210, col211, col212 = st.columns([0.3,0.2,0.1])
with col210:
    slider_date = st.slider("날짜 ", min_value=min_date, max_value=max_date,
                             value=(min_date, now_date2))
with col211:
    slider_week = st.slider("주간", min_value=min_date, max_value=max_date,
                             step=datetime.timedelta(weeks=1), value=(min_date, now_date2))
with col212:
    slider_month = st.slider("월간", min_value=min_date, max_value=max_date,
                             step=datetime.timedelta(weeks=1), value=(min_date, now_date2), format="%Y-%m-%d")

data = pd.read_csv('./119_emergency_dispatch.csv', encoding="cp949")
st.info("금일 출동 내역")
data['datetime'] = pd.to_datetime(data['출동일시'])
st.dataframe(data[(slider_date[0] <= data['datetime']) & (data['datetime'] <= slider_date[1])])
```



## Day4

# streamlit 함수 사용

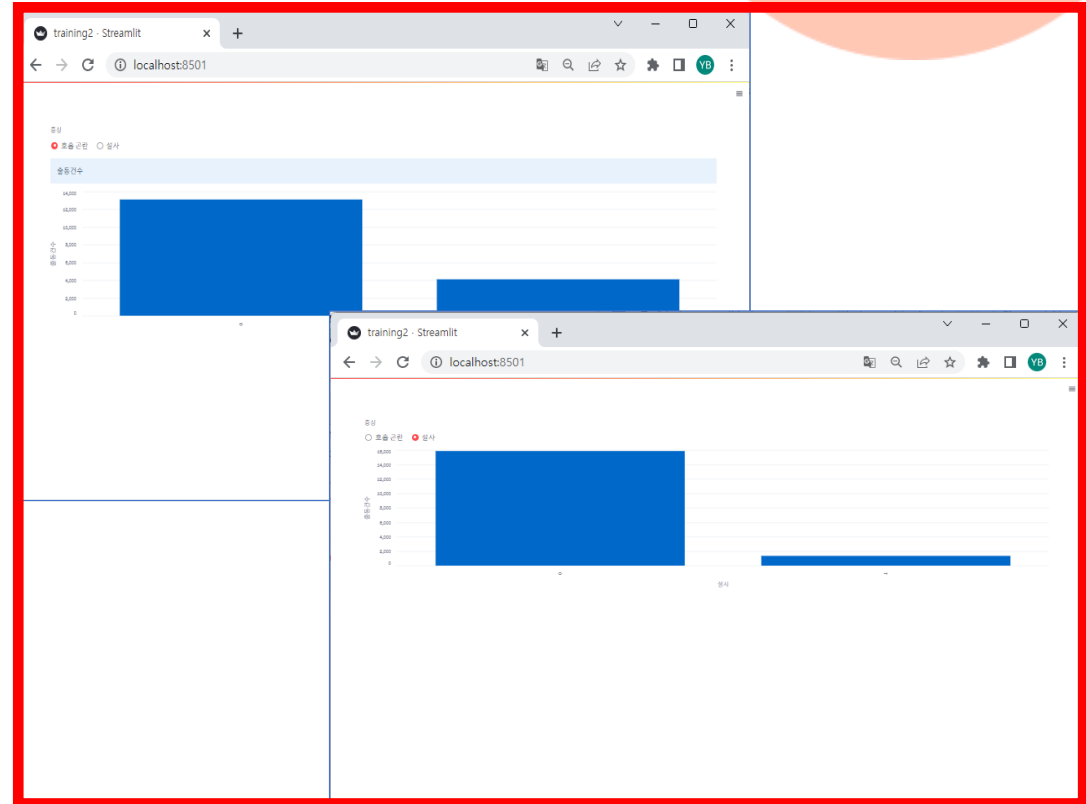
- radio, bar\_chart

```
data = pd.read_csv('./119_emergency_dispatch.csv', encoding="cp949")
st.set_page_config(layout="wide")

select_bins = st.radio("증상", ('호흡 곤란', '설사'), horizontal=True)

if select_bins == '호흡 곤란':
    st.info("출동건수")
    group_day = data.groupby(by='호흡 곤란', as_index=False)['ID'].count()
    group_day = group_day.rename(columns={"ID": '출동건수'})
    st.bar_chart(data=group_day, x='호흡 곤란', y='출동건수', use_container_width=True)

else:
    group_month = data.groupby(by='설사', as_index=False)['ID'].count()
    group_month = group_month.rename(columns={"ID": '출동건수'})
    group_month = group_month.sort_values('설사', ascending=True)
    st.bar_chart(data=group_month, x='설사', y='출동건수', use_container_width=True)
```



## Day4

# streamlit 함수 사용

- **radio**, bar\_chart

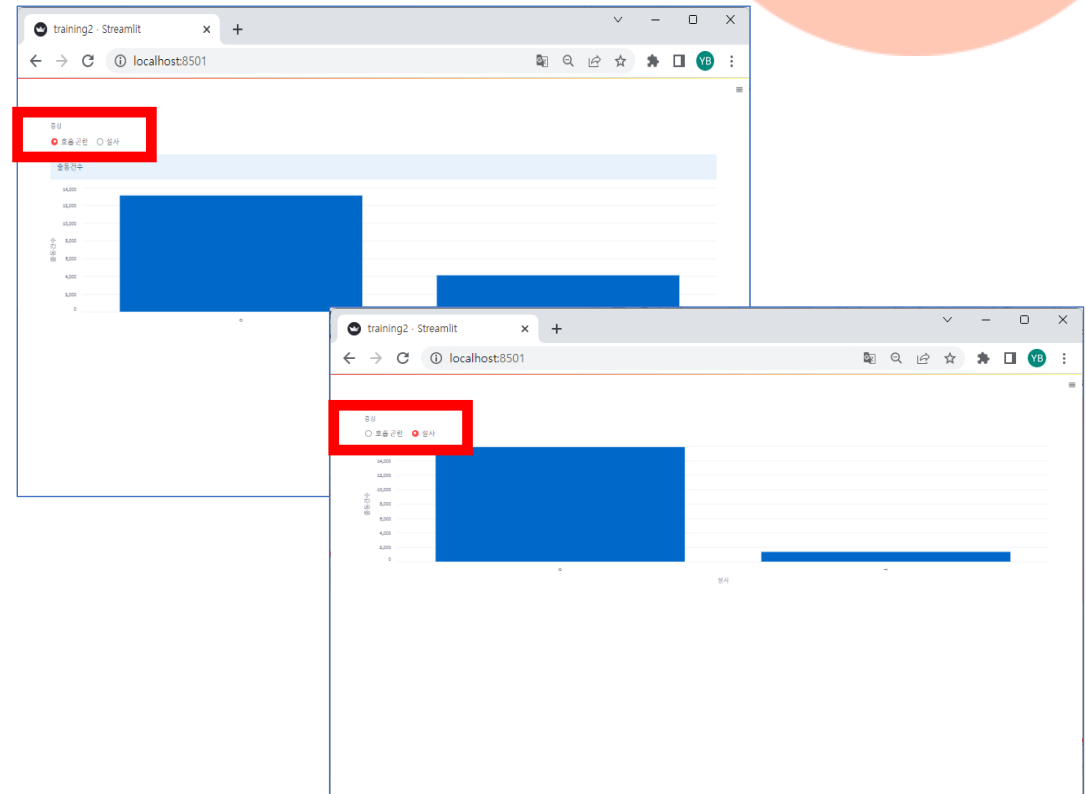
```
data = pd.read_csv('./119_emergency_dispatch.csv', encoding="cp949")

st.set_page_config(layout="wide")

select_bins = st.radio("증상", ('호흡 곤란', '설사'), horizontal=True)

if select_bins == '호흡 곤란':
    st.info("출동건수")
    group_day = data.groupby(by='호흡 곤란', as_index=False)['ID'].count()
    group_day = group_day.rename(columns={"ID": '출동건수'})
    st.bar_chart(data=group_day, x='호흡 곤란', y='출동건수', use_container_width=True)

else:
    group_month = data.groupby(by='설사', as_index=False)['ID'].count()
    group_month = group_month.rename(columns={"ID": '출동건수'})
    group_month = group_month.sort_values('설사', ascending=True)
    st.bar_chart(data=group_month, x='설사', y='출동건수', use_container_width=True)
```



## Day4

# streamlit 함수 사용

- radio, bar\_chart

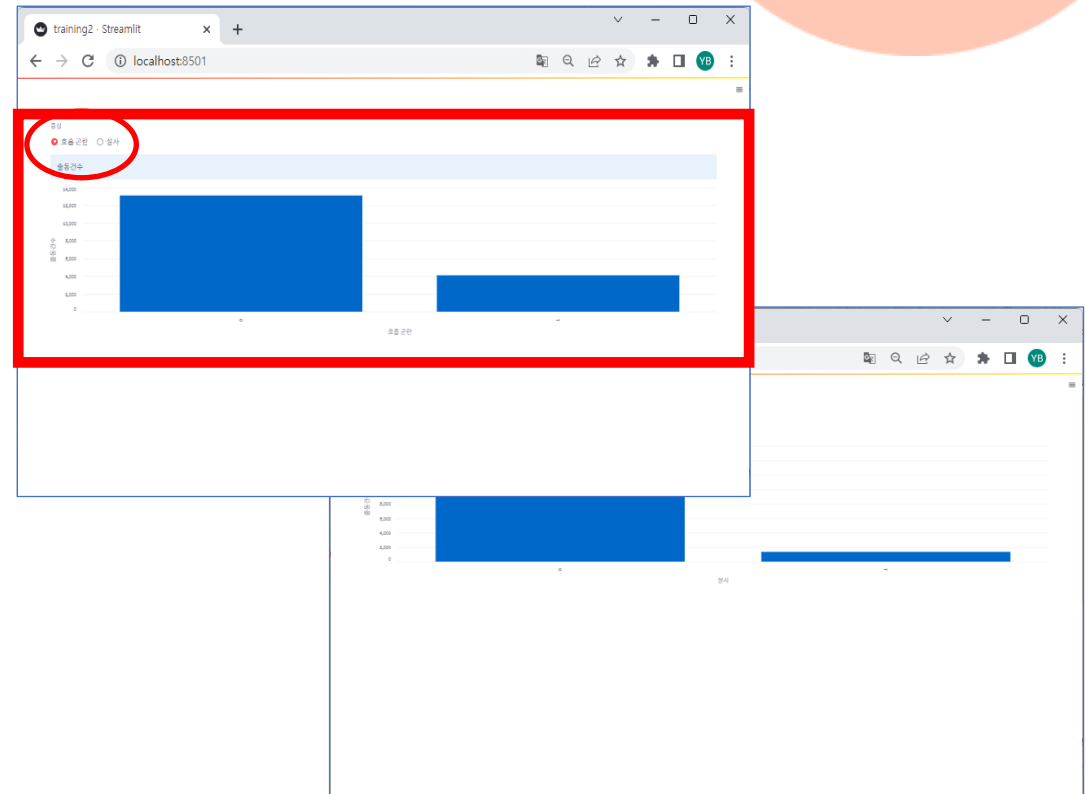
```
data = pd.read_csv('./119_emergency_dispatch.csv', encoding="cp949")

st.set_page_config(layout="wide")

select_bins = st.radio("증상", ('호흡 곤란', '설사'), horizontal=True)

if select_bins == '호흡 곤란':
    st.info("출동건수")
    group_day = data.groupby(by='호흡 곤란', as_index=False)['ID'].count()
    group_day = group_day.rename(columns={"ID": '출동건수'})
    st.bar_chart(data=group_day, x='호흡 곤란', y='출동건수', use_container_width=True)

else:
    group_month = data.groupby(by='설사', as_index=False)['ID'].count()
    group_month = group_month.rename(columns={"ID": '출동건수'})
    group_month = group_month.sort_values('설사', ascending=True)
    st.bar_chart(data=group_month, x='설사', y='출동건수', use_container_width=True)
```





## Day4

# streamlit 함수 사용

- radio, bar\_chart

```
data = pd.read_csv('./119_emergency_dispatch.csv', encoding="cp949")

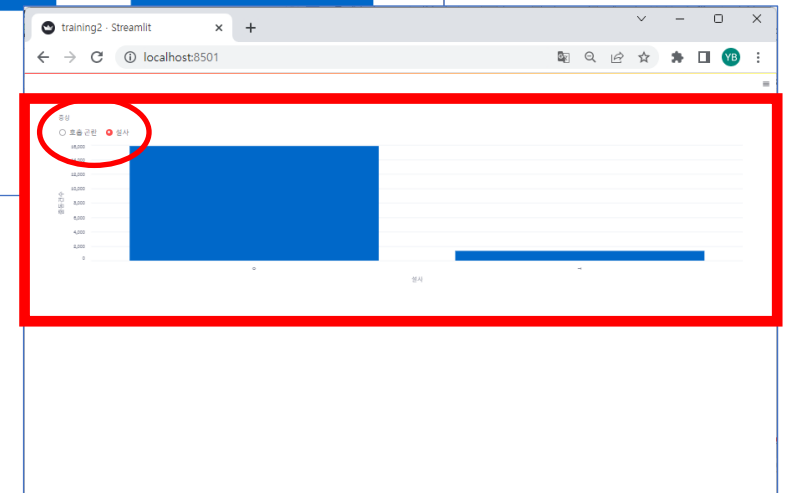
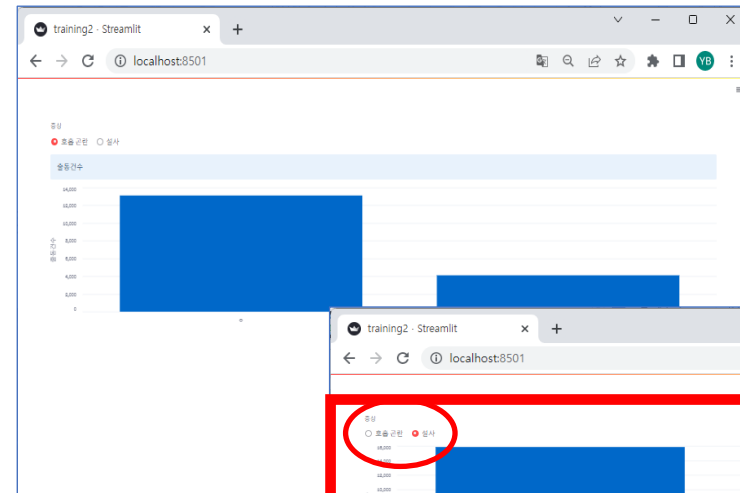
st.set_page_config(layout="wide")

select_bins = st.radio("증상", ('호흡 곤란', '설사'), horizontal=True)

if select_bins == '호흡 곤란':
    st.info("출동건수")
    group_day = data.groupby(by='호흡 곤란', as_index=False)['ID'].count()
    group_day = group_day.rename(columns={"ID": '출동건수'})
    st.bar_chart(data=group_day, x='호흡 곤란', y='출동건수', use_container_width=True)
```

else:

```
    group_month = data.groupby(by='설사', as_index=False)['ID'].count()
    group_month = group_month.rename(columns={"ID": '출동건수'})
    group_month = group_month.sort_values('설사', ascending=True)
    st.bar_chart(data=group_month, x='설사', y='출동건수', use_container_width=True)
```



## Day4

## streamlit 함수 사용

## • line\_chart

```
data = pd.read_csv('./119_emergency_dispatch.csv', encoding="cp949")

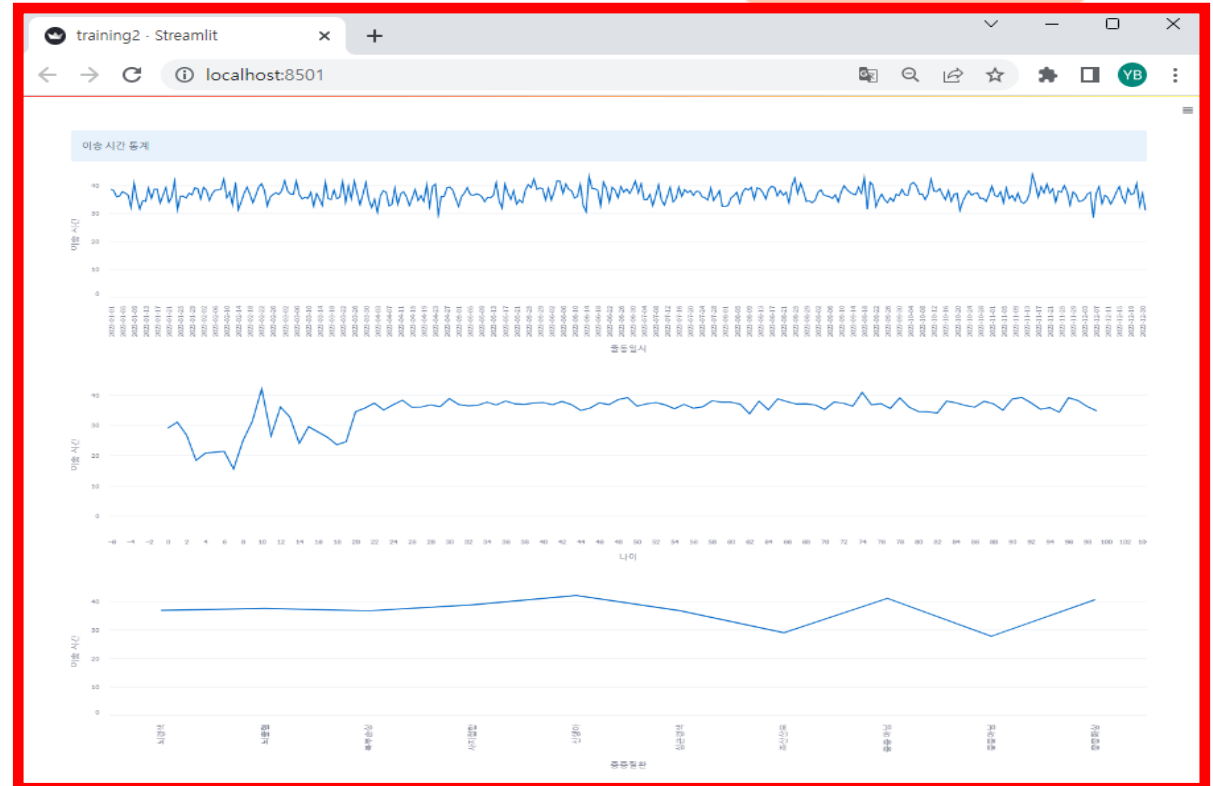
st.set_page_config(layout="wide")

st.info("미송 시간 통계")

group_day_time = data.groupby(by=['출동일시'], as_index=False)['미송 시간'].mean()
group_day_time = group_day_time.rename(columns={"미송 시간": '미송 시간'})
st.line_chart(data=group_day_time, x='출동일시', y='미송 시간', use_container_width=True)

group_week_time = data.groupby(by=['나이'], as_index=False)['미송 시간'].mean()
group_week_time = group_week_time.rename(columns={"미송 시간": '미송 시간'})
st.line_chart(data=group_week_time, x='나이', y='미송 시간', use_container_width=True)

group_month_time = data.groupby(by=['중증질환'], as_index=False)['미송 시간'].mean()
group_month_time = group_month_time.rename(columns={"미송 시간": '미송 시간'})
st.line_chart(data=group_month_time, x='중증질환', y='미송 시간', use_container_width=True)
```



## Day4

# streamlit 함수 사용

- plotly\_chart

```
import pandas as pd
import streamlit as st
import plotly.express as px

data = pd.read_csv('./119_emergency_dispatch.csv', encoding="cp949")

st.set_page_config(layout="wide")

st.info("중증 질환별 통계")

group_day_disease = data.groupby(*****)[*****].*****
group_day_disease = *****.rename(columns={"*****": "*****"})

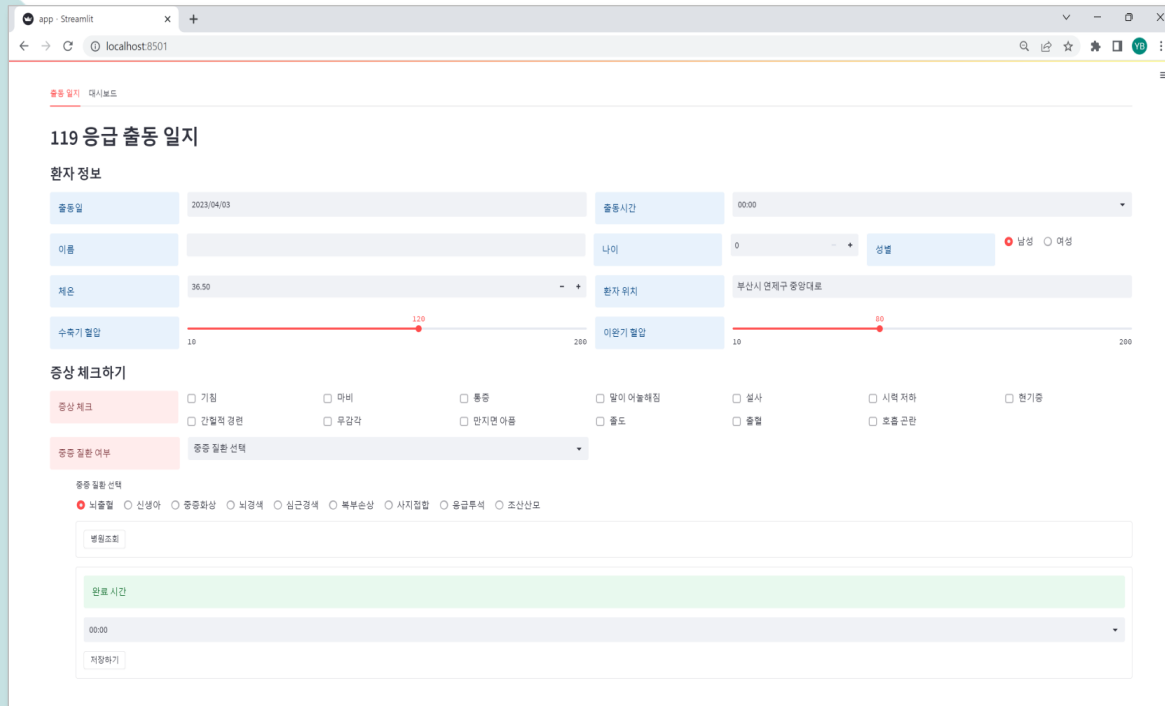
fig = px.pie(*****)
fig.update_traces(textposition='*****', textinfo='*****')
fig.update_layout(*****)
st.plotly_chart(*****)
```



# Day4

## streamlit 함수 사용

- [도전미션] 출동일지 + 대시보드 app.py파일 1개로 통합





## Day4 실습 시간



# 미션5 개별실습

사용자 프로토타입 개발

## Day4

## 실습 가이드

이 시간은?	[개인실습] 미션5 대시보드 화면 만들기
시간	10:30~12:20
장소	실습용 노트북 (강의장은 켜 두세요)
준비사항	AIVLE-EDU 강의실 > 학습자료 > 실습파일 zip다운로드
내용	교재와 ipynb 파일의 가이드를 따라 프로젝트를 수행하여 파일 완성
순서	미션5. 대시보드 개발
결과물	Ipynb 실습파일, app.py 파일 완성본
제출	별도 제출파일 없음
다음 일정	점심시간 → 13:30~15:40 [조별실습]



## Day4

### 점심시간



실습시간 동안 수고 하셨습니다.  
점심시간에는 휴식에 집중해 주세요.

열정은 실습 시간에!

Day4  
중간 점검

Today Mission

## 119 병원선정 솔루션 기획



### [미션5] 대시보드 개발

- Steamlit 활용
- 통계 그래프 출력



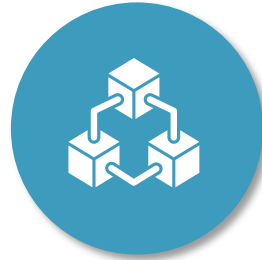
## Day4

# 대시보드 개발 순서



### 정보정의

- 당일 출동이력
- 출동 건수
- 이송시간
- 중증질환 비율



### 통계 데이터 집계

- 일별 집계
- 주별 집계
- 월별 집계



### 화면 구성

- line\_chart
- bar\_chart
- plotly\_chart



## Day4 실습 시간



# 미션5 조별실습

프로토타입 + 대시보드 통합

## Day4

# 실습 가이드

이 시간은?	[조별 실습] 미션5 대시보드 화면 만들기
시간	13:30~16:20
장소	실습용 노트북 (강의장은 켜 두세요)
준비사항	AIVLE-EDU 강의실 > 학습자료 > 실습파일 실zip다운로드
내용	교재와 ipynb 파일의 가이드를 따라 프로젝트를 수행하여 파일 완성
순서	미션5. 대시보드 개발
결과물	Ipynb 실습파일, app.py 파일 완성본
제출	미션별 Ipynb 파일 5개를 파일 한 개로 압축하여 AIVLE Edu 업로드
다음 일정	셀프테스트, 설문조사

## Day4

# Summary

## 4일차 Wrap-up

1. 미션 점검
2. Lesson-learned
3. 프로젝트 의미



데이터 집계



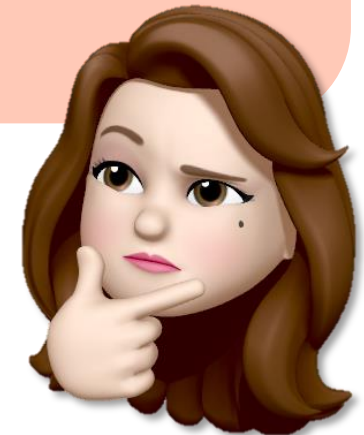
line\_chart



bar\_chart



plotly\_chart



Day4

미션 점검

미션5 Review

강사코드

[https://drive.google.com/drive/folders/  
16sUYgeZx2PaM08u1XU07eEkavyP1VMJz  
?usp=sharing](https://drive.google.com/drive/folders/16sUYgeZx2PaM08u1XU07eEkavyP1VMJz?usp=sharing)

<https://url.kr/aqm6tn>

## Day4

# 미니프로젝트 개요

## 미니프로젝트 5차 미션

**메인목표: 응급환자의 빠른 병원 이송을 위한 119 구급대원의 병원선정 솔루션 프로토타입 제작**

미니프로젝트 모듈	일일 목표	일일 미션	실습 난이도
미니프로젝트 1차 데이터 전처리	데이터 확보 공공데이터 수집 및 가공	Day.1 API 데이터 수집	Level.1- 중급
미니프로젝트 2차 데이터 분석	최적 모델 만들기 Feature 특성 이해 및 최적 모델 찾기	Day.2 데이터분석및 모델링	
미니프로젝트 3차 머신러닝	프로토타입 개발	Day.3 프로토타입 개발	Level.2- 고급
미니프로젝트 4차 딥러닝	사용자 필요 정보 입력 분석 정보 출력	Day.4 대시보드 개발	
미니프로젝트 5차 데이터 분석 종합	대시보드 개발		
미니프로젝트 6차 IT 인프라 설계	구급출동 데이터의 통계 정보 제시		
미니프로젝트 7차 제안전략/PT			

Level.1- 중급

미션 안내  
가이드 코드 제공  
프로토타입 UI 기획

Level.2- 고급

미션 안내  
프로토타입 UI 기획  
도전미션 수행

## Day4

# Lesson Learned

### 오늘의 미션

**bar\_chart**



조건에 따라 원하는 데이터가 출력 되도록 시각화 할 수 있다.

**line\_chart**



원하는 기간의 추세를 보여주고, 항목간 변화를 비교 할 수 있다.

**plotly\_chart**



특정 항목을 그룹으로 묶고, 해당 항목의 이루는 범주별 비율을 표시할 수 있다.

**웹 프레임 워크를 사용하여 데이터 분석 프로토타입을 제작할 수 있다.**

**DX 컨설턴트로서 데이터 분석 결과를 보다 효과적으로 제시할 수 있다.**

Day4

[다시 생각해 보는] 프로젝트 의미



**DX 솔루션 기획 및 컨설팅을 통해**  
**우리 주위의 문제를 해소 시키고**  
**더 나은 세상으로 변화 시킬 수 있다.**





*Thank you!*  
**감사합니다.**

**셀프테스트/설문조사 진행해 주세요.**