CLASS 8. 파이썬 – 시각화 및 분석(2)

수업 목차

복습

#P1. 프로배구와 프로농구의
관중현황 추이 비교 보고서 작성

야구의 데이터

: 트래킹데이터란?

미니 프로젝트

#P2. 야구 데이터 분석 보고서 작성

부록 : 파이썬 문법

#3. 데이터 필터링

복습 #P1. 프로배구와 프로농구의 관중현황 추이 비교 분석 보고서 작성

```
#1. 환경설정 : 한글폰트 설치
import matplotlib as mpl
import matplotlib.pyplot as plt

mpl.rcParams['axes.unicode_minus'] = False
plt.rcParams['font.family'] = 'NanumGothic'
```

```
#2. 데이터 불러오기
import pandas as pd

#df1 : 프로배구
df1 = pd.read_excel('sample_2.xlsx', engine='openpyxl')

#df2 : 프로농구
df2 = pd.read_excel('sample_3.xlsx', engine='openpyxl')
```

복습 #P1. 프로배구와 프로농구의 관중현황 추이 비교 분석 보고서 작성

```
#3. 데이터 전처리
#(1) 문자('-','미개최','미진행') -> 숫자(0) 변환

df1 = df1.replace('-',0)
df1 = df1.replace('미개최',0)
df1 = df1.replace('미진행',0)
```

```
#3. 데이터 전처리
#(2) 프로배구 '합계'열 생성

df1['합계'] = df1['컵대회'] + df1['정규리그'] + df1['올스타전'] + df1['포스트시즌'] + df1['기타대회']
```

```
#3. 데이터 전처리
#(3) 구분(시즌)별 내림차순 정렬

df1 = df1.sort_values(by = ['구분'], ascending=True)
df2 = df2.sort_values(by = ['구분'], ascending=True)
```

복습 #P1. 프로배구와 프로농구의 관중현황 추이 비교 분석 보고서 작성

```
#3. 데이터 전처리
#(4) 인덱스 재설정

df1 = df1.reset_index(drop=True)
df2 = df2.reset_index(drop=True)
```

```
#4. 데이터 시각화 : 관중현황 추이 꺾은선 그래프

plt.figure(figsize=(20,10))
plt.plot(df1['구분'], df1['합계'], color="green", linewidth=5)
plt.plot(df2['구분'], df2['합계'], color="orange", linewidth=5)

plt.title('프로배구, 프로농구 관중 현황 추이 비교', fontsize=20)
plt.xlabel('시즌', fontsize=15)
plt.ylabel('관중 수(명)', fontsize=15)
plt.xticks(df1['구분'], rotation=45)
plt.legend(['프로배구','프로농구'], fontsize=20)
plt.show()
```

복습 #P1. 프로배구와 프로농구의 관중현황 추이 비교 분석 보고서 작성

□ 프로배구의 관중 증가 요인

1) 스타플레이어의 탄생 : 김연경 선수 外

- 세계적으로 인정받는 실력
- 방송 출연 등으로 인지도 상승

2) 국제 대회의 성적 : 여자배구

- 2012년 런던 올림픽 4위
- 2014년 인천 아시안게임 금메달
- 2020년 도쿄 올림픽 4위

□ 프로농구의 관중 감소 요인

1) 스타플레이어의 부재

2) 저조한 경기력

- 과도한 규제 (예)외국인 선수 신장 제한(2019-20시즌 폐지)

야구의 데이터 : 트래킹데이터(Tracking Data)란?

□ **트래킹데이터(Tracking Data) : 공을 추적해서 얻는 데이터**

→ **선수의 특성을 분석하는데 주로 사용됨**

※트래킹데이터 분석시 일반화의 오류에 빠지지 않게 조심! (예)'회전수가 높을수록 좋은 투구' 는 아니다!

① **구속(Pitch Speed) (km/h)** : 투수가 던진 공의 빠르기

② **회전수(Pitch Spin) (rpm)** : 투수가 던진 공의 분당 회전수

③ **구종(Pitch Type)** : 던지는 방법에 따라 구분되는 공의 종류

(예) FF : 패스트볼, SL : 슬라이더, CH : 체인지업, CU : 커브

④ **투구결과(Pitch Result)** : 투수가 던진 공의 결과 기록

(예) SE(Single) : 1루타, DE(Double) : 2루타, HR(Homerun) : 홈런, B(Ball) : 볼, CS(CallStrike) : 스트라이크

미니 프로젝트 #P2. 야구 데이터 분석 보고서 작성

〈사전준비〉

가. 환경설정

(1) 터미널 : 라이브러리 설치, 한글폰트 설치

(2) 주피터노트북 : 한글폰트 설치

미니 프로젝트 #P2. 야구 데이터 분석 보고서 작성

〈타자 기록데이터〉

나. 데이터 가져오기 : sample_12.xlsx (kusf 대학야구 U-리그 왕중왕전 결승전 타자 기록 데이터)

다. 데이터 확인

(1) head()

(2) info()

(3) unique()

미니 프로젝트 #P2. 야구 데이터 분석 보고서 작성

〈시각화〉

라. 팀별 안타수 막대그래프(Bar chart)

마. 안타 종류 원형그래프(Pie chart)

바. 안타수, 도루수, 사사구수(4구+사구), 삼진수 막대그래프(Bar chart)

미니 프로젝트 #P2. 야구 데이터 분석 보고서 작성

〈트래킹데이터〉

나. 데이터 가져오기 : sample_10.xlsx (kusf 대학야구 U-리그 왕중왕전 결승전 트래킹데이터)

다. 데이터 확인

(1) head()

(2) info()

(3) unique()

미니 프로젝트 #P2. 야구 데이터 분석 보고서 작성

〈시각화〉

라. 투수별 구종구사율 원형그래프(Pie chart)

마. 투수별 패스트볼(구종==FF) 구속 추이 비교 꺾은선 그래프(Line chart)

바. 투수별 구종별 최소, 평균, 최대 구속 표(Table)

사. 투수별 패스트볼(구종==FF) 특성 비교 산점도(Scatter Plot)

아. 투수별 구종별 구속과 회전수 특성 비교 산점도(Scatter Plot)

미니 프로젝트 #P2. 야구 데이터 분석 보고서 작성

가. 환경설정 - 터미널

```
#1. 라이브러리 설치

pip install openpyxl
pip install xlrd
```

```
#2. 한글폰트 설치

sudo apt-get install  -y fonts-nanum font-nanum-coding fonts-nanum-extra
rm  -rf ~/.cache/matplotlib
fc-cache -fv
```

미니 프로젝트 #P2. 야구 데이터 분석 보고서 작성

가. 환경설정 – 주피터노트북

```
#3. 한글폰트 설치

import matplotlib as mpl
import matplotlib.pyplot as plt

mpl.rcParams['axes.unicode_minus'] = False
plt.rcParams['font.family'] = 'NanumGothic'
```

미니 프로젝트 #P2. 야구 데이터 분석 보고서 작성

〈타자 기록 데이터〉

나. 데이터 가져오기 : sample_12.xlsx (kusf 대학야구 U-리그 왕중왕전 결승전 타자 기록 데이터)

```
import pandas as pd

df = pd.read_excel('sample_12.xlsx', engine='openpyxl')
```

미니 프로젝트 #P2. 야구 데이터 분석 보고서 작성

라. 팀별 안타수 막대그래프(Bar chart)

```
#1. 데이터 전처리 : '1타' 열 생성
df['1타'] = df['안타'] - (df['2타'] + df['3타'] + df['홈런'])
```

```
#2. 데이터필터링 : 각 팀별 필터 생성
filt1 = df['팀'] == '원광대'
filt2 = df['팀'] == '성균관대'
```

```
#3. 시각화
plt.figure(figsize=(10,5))
plt.barh('원광대',df[filt1]['안타'].sum())
plt.barh('성균관대', df[filt2]['안타'].sum())
plt.title('팀별 안타 수 막대그래프')
plt.show()
```

미니 프로젝트 #P2. 야구 데이터 분석 보고서 작성

마. 안타 종류 원형그래프(Pie chart)

```
plt.figure(figsize=(15,8))

plt.subplot(1,2,1)
x1 = [df[filt1]['1타'].sum(), df[filt1]['2타'].sum(), df[filt1]['3타'].sum(), df[filt1]['홈런'].sum()]
plt.pie(x1, startangle=90, autopct="%.1f%%")
plt.title('원광대 타구 종류 비율')
plt.legend(['1루타','2루타','3루타','홈런'])

plt.subplot(1,2,2)
x2 = [df[filt2]['1타'].sum(), df[filt2]['2타'].sum(), df[filt2]['3타'].sum(), df[filt2]['홈런'].sum()]
plt.pie(x2, startangle=90, autopct="%.1f%%", textprops={'fontsize':15})
plt.title('성균관대 타구 종류 비율')
plt.legend(['1루타','2루타','3루타','홈런'])


plt.show()
```

미니 프로젝트 #P2. 야구 데이터 분석 보고서 작성

바. 안타수, 도루수, 사사구수(4구+사구), 삼진수 막대그래프(Bar chart) (1)

```
plt.figure(figsize=(10,10))

plt.subplot(2,2,1)
plt.bar('원광대', df[filt1]['안타'].sum())
plt.bar('성균관대', df[filt2]['안타'].sum())
plt.ylim(0,10)
plt.title('팀별 안타수 비교')
plt.text(0,df[filt1]['안타'].sum(), df[filt1]['안타'].sum())
plt.text(1,df[filt2]['안타'].sum(), df[filt2]['안타'].sum())

plt.subplot(2,2,2)
plt.bar('원광대', df[filt1]['도루'].sum())
plt.bar('성균관대', df[filt2]['도루'].sum())
plt.yticks([0,1,2])
plt.title('팀별 도루수 비교')
plt.text(0,df[filt1]['도루'].sum(), df[filt1]['도루'].sum())
plt.text(1,df[filt2]['도루'].sum(), df[filt2]['도루'].sum())
```

미니 프로젝트 #P2. 야구 데이터 분석 보고서 작성

바. 안타수, 도루수, 사사구수(4구+사구), 삼진수 막대그래프(Bar chart) (2)

```
plt.subplot(2,2,3)
plt.bar('원광대', df[filt1]['4구'].sum() + df[filt1]['사구'].sum())
plt.bar('성균관대', df[filt2]['4구'].sum() + df[filt2]['사구'].sum())
plt.ylim(0,10)
plt.title('팀별 사사구수 비교')
plt.text(0,df[filt1]['4구'].sum() + df[filt1]['사구'].sum(), df[filt1]['4구'].sum() + df[filt1]['사구'].sum())
plt.text(1,df[filt2]['4구'].sum() + df[filt2]['사구'].sum(), df[filt2]['4구'].sum() + df[filt2]['사구'].sum())

plt.subplot(2,2,4)
plt.bar('원광대', df[filt1]['삼진'].sum())
plt.bar('성균관대', df[filt2]['삼진'].sum())
plt.ylim(0,10)
plt.title('팀별 삼진수 비교')
plt.text(0,df[filt1]['삼진'].sum(), df[filt1]['삼진'].sum())
plt.text(1,df[filt2]['삼진'].sum(), df[filt2]['삼진'].sum())


plt.show()
```

미니 프로젝트 #P2. 야구 데이터 분석 보고서 작성

〈트래킹데이터〉

나. 데이터 가져오기 : sample_10.xlsx (kusf 대학야구 U-리그 왕중왕전 결승전 트래킹데이터)

```
import pandas as pd

df = pd.read_excel('sample_10.xlsx', engine='openpyxl')
```

미니 프로젝트 #P2. 야구 데이터 분석 보고서 작성

다. 데이터 확인 – head()

```
df.head()
```

다. 데이터 확인 – info()

```
df.info()
```

다. 데이터 확인 – unique()

```
df['투수'].unique()
```

미니 프로젝트 #P2. 야구 데이터 분석 보고서 작성

라. 투수별 구종구사율 원형그래프(Pie Chart)

```
#1. 데이터 필터링 : 각 투수별 필터 생성

#(1) 투수 == 이용헌
filt1 = df['투수'] == '이용헌'

#(2) 투수 == 김태원
filt2 = df['투수'] == '김태원'

#(3) 투수 == 조민석
filt3 = df['투수'] == '조민석'

#(4) 투수 == 이준호
filt4 = df['투수'] == '이준호'

#(5) 투수 == 주승우
filt5 = df['투수'] == '주승우'
```

미니 프로젝트 #P2. 야구 데이터 분석 보고서 작성

라. 투수별 구종구사율 원형그래프(Pie Chart)

```
#2. 투수별 구종 확인 - value_counts() : 유니크한 값의 갯수 출력

#(1) 투수 == 이용헌
df[filt1]['구종'].value_counts()
```

미니 프로젝트 #P2. 야구 데이터 분석 보고서 작성

라. 투수별 구종구사율 원형그래프(Pie Chart)

```
#3. 데이터 시각화 - 투수별 구종구사율 원형그래프

plt.figure(figsize=(10,10))
plt.title('이용헌 구종구사율')
plt.pie(df[filt1]['구종'].value_counts(), labels=['SL','FF','CH','CU'], autopct="%.1f%%",
       startangle=90, counterclock=False,
       textprops={'fontsize':10},
       colors=['blue','lightgray','lightgray','lightgray'],
       explode=[0.1,0,0,0])
plt.legend()
plt.show()
```

미니 프로젝트 #P2. 야구 데이터 분석 보고서 작성

라. 투수별 구종구사율 원형그래프(Pie Chart)

```
plt.figure(figsize=(20,20))

#(1) 투수 == 이용헌
plt.subplot(3,2,1) #3행 X 2열 - 첫 번째
plt.title('이용헌 구종구사율')
plt.pie(df[filt1]['구종'].value_counts(), labels=['SL','FF','CH','CU'], autopct="%.1f%%",
       startangle=90, counterclock=False,textprops={'fontsize':10},
       colors=['blue','lightgray','lightgray','lightgray'],explode=[0.1,0,0,0])
plt.legend()

#(2) 투수 == 김태원
plt.subplot(3,2,2) #3행 X 2열 - 두 번째
plt.title('김태원 구종구사율')
plt.pie(df[filt2]['구종'].value_counts(), labels=['FF','SL','CU','CH'], autopct="%.1f%%",
       startangle=90, counterclock=False,textprops={'fontsize':10},
       colors=['red','lightgray','lightgray','lightgray'],explode=[0.1,0,0,0])
plt.legend()

plt.show()
```

미니 프로젝트 #P2. 야구 데이터 분석 보고서 작성

마. 투수별 패스트볼(구종==FF) 구속 추이 비교 꺾은선 그래프(Line chart)

```
#1. 데이터 필터링 : 각 투수별 + 패스트볼 필터 생성

#(1) 투수 == 이용헌 and 구종 == FF
filt1 = (df['투수'] == '이용헌') & (df['구종'] == 'FF')

#(2) 투수 == 김태원 and 구종 == FF
filt2 = (df['투수'] == '김태원') & (df['구종'] == 'FF')

#(3) 투수 == 조민석 and 구종 == FF
filt3 = (df['투수'] == '조민석') & (df['구종'] == 'FF')

#(4) 투수 == 이준호 and 구종 == FF
filt4 = (df['투수'] == '이준호') & (df['구종'] == 'FF')

#(5) 투수 == 주승우 and 구종 == FF
filt5 = (df['투수'] == '주승우') & (df['구종'] == 'FF')
```

미니 프로젝트 #P2. 야구 데이터 분석 보고서 작성

마. 투수별 패스트볼(구종==FF) 구속 추이 비교 꺾은선 그래프(Line chart)

```
#2. 데이터 시각화 - 투수별 패스트볼 구속 추이 비교 꺾은선 그래프
plt.figure(figsize=(20,10))

plt.title('투수별 패스트볼 구속 추이 비교 그래프')

plt.plot(df[filt1]['구속'], label='이용헌')
plt.plot(df[filt2]['구속'], label='김태원')
plt.plot(df[filt3]['구속'], label='조민석')
plt.plot(df[filt4]['구속'], label='이준호')
plt.plot(df[filt5]['구속'], label='주승우')

plt.ylim(110,150)
plt.grid()
plt.legend()

plt.show()
```

미니 프로젝트 #P2. 야구 데이터 분석 보고서 작성

바. 투수별 구종별 최소, 평균, 최대 구속 표(Table)

```
#1. 투수별 평균 구속

df.groupby(['투수'])[['구속']].mean()
```

```
#2. 투수별 구종별 평균 구속

df.groupby(['투수','구종'])[['구속']].mean()
```

```
#3. 투수별 구종별 최고, 평균, 최소 구속

df.groupby(['투수','구종'])[['구속']].agg(['min','mean','max'])
```

미니 프로젝트 #P2. 야구 데이터 분석 보고서 작성

사. 투수별 패스트볼(구종==FF) 특성 비교 산점도(Scatter Plot)

```
#1. 데이터 필터링 : 투수별 + 패스트볼

#(1) 투수 == 이용헌 and 구종 == FF
filt1 = (df['투수'] == '이용헌') & (df['구종'] == 'FF')

#(2) 투수 == 김태원 and 구종 == FF
filt2 = (df['투수'] == '김태원') & (df['구종'] == 'FF')

#(3) 투수 == 조민석 and 구종 == FF
filt3 = (df['투수'] == '조민석') & (df['구종'] == 'FF')

#(4) 투수 == 이준호 and 구종 == FF
filt4 = (df['투수'] == '이준호') & (df['구종'] == 'FF')

#(5) 투수 == 주승우 and 구종 == FF
filt5 = (df['투수'] == '주승우') & (df['구종'] == 'FF')
```

미니 프로젝트 #P2. 야구 데이터 분석 보고서 작성

사. 투수별 패스트볼(구종==FF) 특성 비교 산점도(Scatter Plot)

```
#2. 투수별 패스트볼 특성 비교(scatter plot)

plt.figure(figsize=(20,10))

#(2) 투수 == 김태원 and 구종 == FF
plt.scatter(df[filt2]['구속'], df[filt2]['회전수'], label='김태원', color="orange", alpha=0.5, s=100)

#(5) 투수 == 주승우 and 구종 == FF
plt.scatter(df[filt5]['구속'], df[filt5]['회전수'], label='주승우', color="purple", alpha=0.5, s=100)

plt.legend(fontsize=15)
plt.xlim(130,150)
plt.ylim(1000,3000)
plt.grid()
plt.show()
```

미니 프로젝트 #P2. 야구 데이터 분석 보고서 작성

아. 투수별 구종별 구속과 회전수 특성 비교 산점도(Scatter Plot)

```
filt = df['투수'] == '이용헌 '
df[filt]['구종'].value_counts()
```

```
#1. 데이터 필터링 : 투수별 구종별

#(1) 투수 == 이용헌 and 구종 == FF
filt1 = (df['투수'] == '이용헌') & (df['구종'] == 'FF')

#(2) 투수 == 이용헌 and 구종 == SL
filt2 = (df['투수'] == '이용헌') & (df['구종'] == 'SL')

#(3) 투수 == 이용헌 and 구종 == CH
filt3 = (df['투수'] == '이용헌') & (df['구종'] == 'CH')

#(4) 투수 == 이용헌 and 구종 == CU
filt4 = (df['투수'] == '이용헌') & (df['구종'] == 'CU')
```

미니 프로젝트 #P2. 야구 데이터 분석 보고서 작성

아. 투수별 구종별 구속과 회전수 특성 비교 산점도(Scatter Plot)

```
#2. 투수별 구종별 구속과 회전수 특성(scatter plot)

plt.figure(figsize=(20,10))

#(1) 투수 == 이용헌 and 구종 == FF
plt.scatter(df[filt1]['구속'], df[filt1]['회전수'], color="red", alpha=0.5, s=100)
#(2) 투수 == 이용헌 and 구종 == SL
plt.scatter(df[filt2]['구속'], df[filt2]['회전수'], color="blue", alpha=0.5, s=100)
#(3) 투수 == 이용헌 and 구종 == CH
plt.scatter(df[filt3]['구속'], df[filt3]['회전수'], color="purple", alpha=0.5, s=100)
#(4) 투수 == 이용헌 and 구종 == CU
plt.scatter(df[filt4]['구속'], df[filt4]['회전수'], color="green", alpha=0.5, s=100)

plt.ylim(1000,3000)
plt.xlim(100,150)
plt.grid()
plt.show()
```

부록. 파이썬 문법 #3. 데이터 필터링

데이터 필터링 : 특정 조건을 만족하는 데이터만 필터링하는것

기본 문법(1) : 열 선택

```
filt = 조건
df[filt]
```

파이썬의 비교 연산자는 == (등호 2개)

예) 데이터프레임 : df

	과일	가격
0	사과	500
1	바나나	100
2	바나나	200
3	딸기	300

1) '과일'이 '바나나 ' 인 데이터만 출력하고 싶을때

```
filt = df[ ' 과일 ' ] == ' 바나나 '
df[filt]
```

[출력결과]

	과일	가격
0	바나나	100
2	바나나	200

2) '가격'이 300원 이상인 과일만 출력하고 싶을때

```
filt = df[ ' 가격 ' ] >= 300
df[filt]
```

[출력결과]

	과일	가격
0	사과	500
3	딸기	300

부록. 파이썬 문법 #3. 데이터 필터링

데이터 필터링 : 특정 조건을 만족하는 데이터만 필터링하는것

기본 문법(1) : 열 선택

```
filt = 조건
df[filt]
```

예) 데이터프레임 : df

	과일	가격
0	사과	500
1	바나나	100
2	바나나	200
3	딸기	300

3) '과일'이 '바나나 ' 이면서 '가격'이 200원 이상인 과일만 출력하고 싶을때

& = and = 그리고

```
filt = (df[ ' 과일 ' ] ==  ' 바나나 ' ) & (df[ ' 가격 ' ] >= 200)
df[filt]
```

[출력결과]

	과일	가격
2	바나나	200

부록. 파이썬 문법 #3. 데이터 필터링

데이터 필터링 : 특정 조건을 만족하는 데이터만 필터링하는것

기본 문법(1) : 열 선택

```
filt = 조건
df[filt]
```

예) 데이터프레임 : df

	과일	가격
0	사과	500
1	바나나	100
2	바나나	200
3	딸기	300

4) '과일'이 '바나나 ' 이거나 '딸기 ' 인 데이터만 출력하고 싶을때

| = or = 또는

```
filt = (df[ ' 과일 ' ] ==  ' 바나나 ' ) | (df[ ' 가격 ' ] >= 200)
df[filt]
```

[출력결과]

	과일	가격
1	바나나	100
2	바나나	200
3	딸기	300

※과제. 야구 데이터 분석 보고서 작성

가. 제출 : wowsjh02@gmail.com

나. 필수 포함 내용

- 표 또는 그래프 3가지 이상 (해당 표 또는 그래프에 대한 내용 해석 필수)