



# [LAB 06] 6. 집단별 요약 시각화



## #01. 준비작업



### 1. 패키지 가져오기

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



### 2. 그래프 초기화

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



### 3. 데이터 가져오기

```
origin = load_data('penguins')
origin
```

```
[94m[data] [0m https://data.hossam.kr/data/lab06/penguins.xlsx
[94m[desc] [0m 남극 팔мер 군도의 펭귄 3종에 대해 신체 치수와 서식지 정보(출처:
seaborn 내장 데이터)
```

field	description
-----	-----
species	펭귄 종
island	서식지

bill_length_mm	부리 길이
bill_depth_mm	부리 두께
flipper_length_mm	날개 길이
body_mass_g	몸무게
sex	성별

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex
0	Adelie	Torgersen	39.1	18.7	181	3750	MALE
1	Adelie	Torgersen	39.5	17.4	186	3800	FEMALE
2	Adelie	Torgersen	40.3	18.0	195	3250	FEMALE
3	Adelie	Torgersen	36.7	19.3	193	3450	FEMALE
4	Adelie	Torgersen	39.3	20.6	190	3650	MALE
...	...	...	...	...	...	...	...
329	Gentoo	Biscoe	47.2	13.7	214	4925	FEMALE
330	Gentoo	Biscoe	46.8	14.3	215	4850	FEMALE
331	Gentoo	Biscoe	50.4	15.7	222	5750	MALE
332	Gentoo	Biscoe	45.2	14.8	212	5200	FEMALE
333	Gentoo	Biscoe	49.9	16.1	213	5400	MALE

334 rows × 7 columns



## #02. barplot

- 집단별 평균/합계/비율 등 요약 통계 시각화의 기본형
- estimator 파라미터로 평균(mean) 외에도 median, sum 등 자유롭게 설정 가능
- 범주형 분석에서 가장 표준적으로 쓰이는 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) barplot 그리기
sb.barplot(
    data=origin, # 사용할 데이터프레임
```

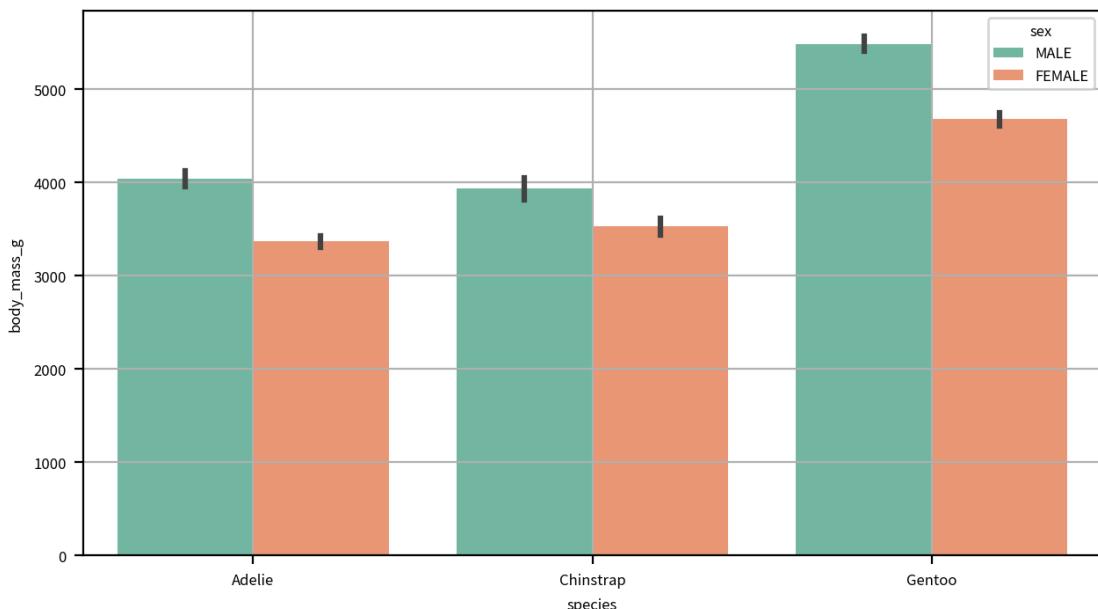
```

x="species",
y="body_mass_g",
hue="sex",
estimator=np.mean,
errorbar=("ci", 95),
준편차, None=없음
palette="Set2"
"bright", ...
)

# 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. countplot (빈도 그래프)

- 범주형 빈도(Count)를 바로 보여주는 가장 단순하고 직관적인 요약 그래프
- 기술통계 보고서·EDA에서 거의 항상 등장

```

# 1) 그래프 초기화
width_px  = 1280           # 그래프 가로 크기
height_px = 720            # 그래프 세로 크기

```

```

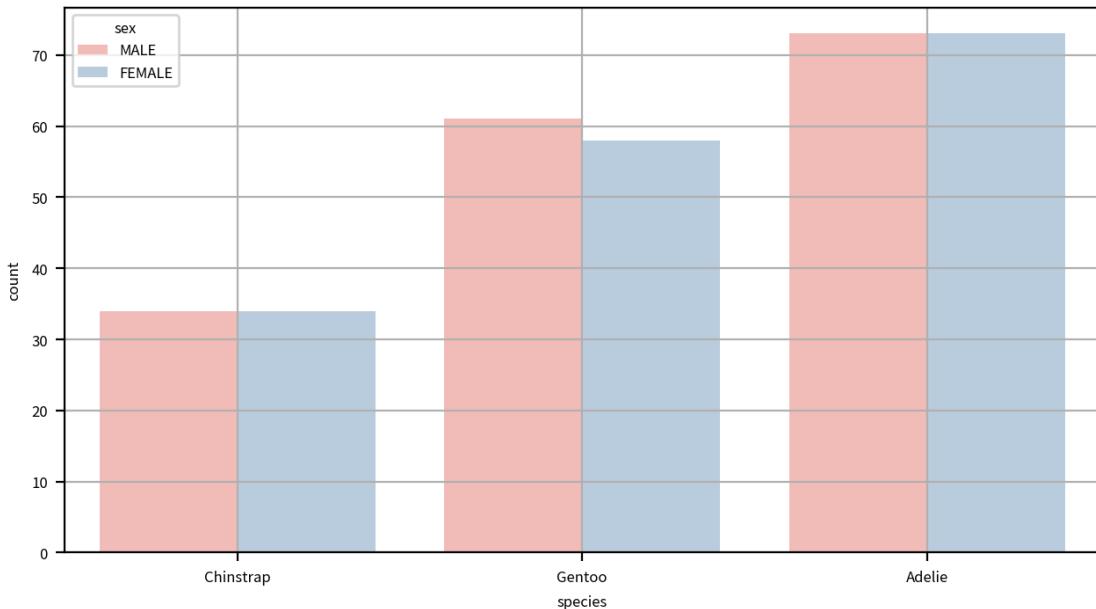
rows = 1 # 그래프 행 수
cols = 1 # 그래프 열 수
figsize = (width_px / my_dpi, height_px / my_dpi)
fig, ax = plt.subplots(rows, cols, figsize=figsize, dpi=my_dpi)

# 2) barplot 그리기
sb.countplot(
    data=origin, # 사용할 데이터프레임
    x="species", # 집계할 범주
    hue="sex", # 그룹 구분: None 가능
    order=['Chinstrap', 'Gentoo', 'Adelie'], # x축 범주 순서
    palette="Pastel1" # 색상 팔레트
)

# 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. pointplot

- 점+오차막대 형태의 집단별 평균 + 신뢰구간 요약

- 여러 그룹(hue) 비교 시 가장 깔끔하고 해석성이 높음
- 회귀분석 전 EDA에서 추세 파악용으로 인기

```

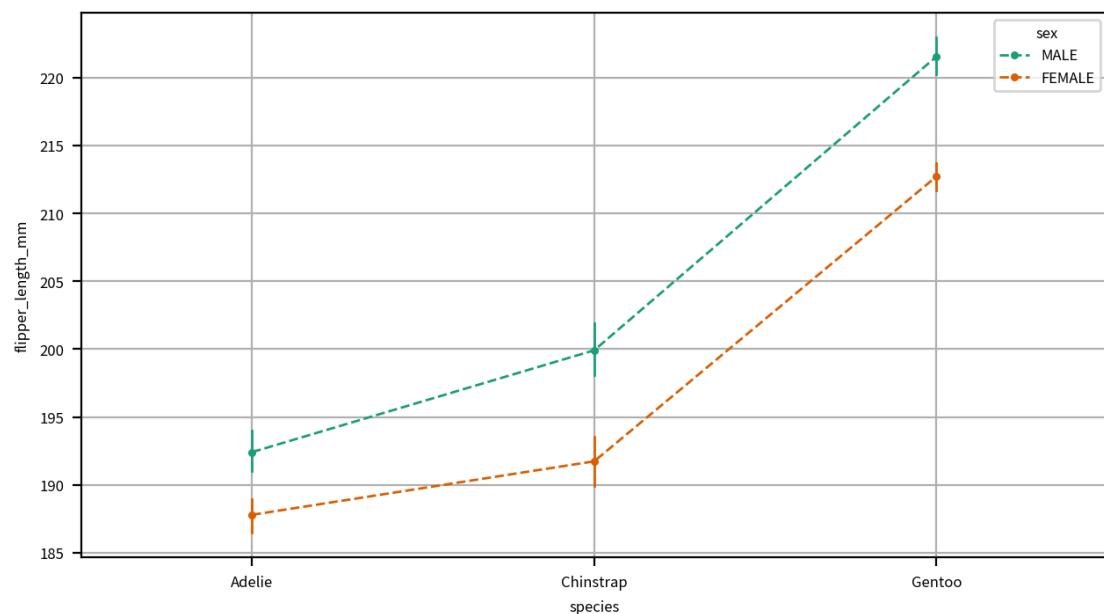
# 1) 그래프 초기화
width_px  = 1280                      # 그래프 가로 크기
height_px = 720                        # 그래프 세로 크기
rows      = 1                           # 그래프 행 수
cols      = 1                           # 그래프 열 수
figsize   = (width_px / my_dpi, height_px / my_dpi)
fig, ax = plt.subplots(rows, cols,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

# 2) barplot 그리기
sb.pointplot(
    data=origin,                      # 사용할 데이터프레임
    x="species",
    y="flipper_length_mm",
    hue="sex",
    estimator=np.mean,                # 요약 방식
    errorbar=("ci", 95),             # 신뢰구간(기본 95%)
    linestyles="--",                 # "-", "--", ":" , "-."
    linewidth=1,
    markers="o",                     # 마커 종류: "o", "s", "D", "^", "v",
    "*",
    palette="Dark2"
)

# 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