K-Digital Training

codingOn

Seaborn

Seaborn

- Seaborn은 데이터 시각화를 위한 Python 라이브러리
- 통계적 그래프를 생성하는 데 유용
- Matplotlib 위에 구축되어 있으며, 데이터의 관계, 분포, 카테고리 비교 등을 시각적으로 쉽게 이해할 수 있게 해줌
- Pandas DataFrame과 잘 통합되어 있어 데이터를 간단히 시각화
- Seaborn에는 자체 데이터셋이 포함되어 있어 다양한 테스트를 할 수 있음
- 공식홈페이지

```
import seaborn as sns
print(sns.get_dataset_names())
```

출처: wikidocs

산점도

- scatterplot() : 두 연속형 변수 간의 **관계를 시각화**
  - x, y: x축과 y축에 사용할 데이터 열
  - hue: 데이터 포인트의 색상을 그룹으로 구분
  - size: 데이터 포인트의 크기를 그룹으로 구분
  - style: 데이터 포인트의 스타일(모양)을 그룹으로 구분
  - palette: 색상 팔레트 지정
  - alpha: 투명도 설정 (0에서 1 사이 값)
  - legend: 범례 표시 여부 ('brief', 'full', False)
  - data: 데이터셋(Pandas DataFrame)

```
sns.scatterplot(
    x="total_bill",
    y="tip",
    hue="sex",
    style="time",
    size="size",
    alpha=0.8,
    data=tips
)
plt.show()
```

stripplot

codingOn

- stripplot() : 단순 데이터 분포를 빠르게 보고 싶을 때
- 점들을 한 줄로 나열: 데이터 포인트를 카테고리별로 한 줄에 나열
- 데이터 포인트들이 겹칠 수 있음
- 단순히 데이터의 분포를 보여주기 때문에 시각적으로 간단함
- 주요 옵션
  - jitter: 점들을 수평으로 약간 퍼뜨려서 겹침을 줄임 (기본값: False)
  - hue: 색상을 사용해 그룹 구분
  - dodge: 여러 카테고리를 옆으로 나란히 배치 (색상 구분 시 유용)

```
sns.stripplot(x="day", y="total_bill", data=tips, jitter=True, hue="sex", dodge=True)
plt.show()
```

swarmplot

codingOn

- swarmplot() : 정확한 데이터 분포를 확인하고 싶을 때
- 점들이 겹치지 않도록 정렬: 데이터를 적절히 분산시켜 서로 겹치지 않게 배치
- 데이터 포인트의 정확한 분포를 더 명확히 보여줌
- stripplot보다 시각적으로 더 읽기 쉬움
- 주요 옵션
  - hue: 색상을 사용해 그룹 구분
  - dodge: 색상 구분 시 카테고리를 옆으로 나란히 배치

```
sns.swarmplot(x="day", y="total_bill", data=tips, hue="sex", dodge=True)
plt.show()
```

relplot

codingOn

- relplot() : 관계형 플롯(Relational Plot)을 생성
- 주요 옵션:
  - x, y: x축과 y축에 사용할 데이터 열
  - hue: 색상으로 그룹을 구분
  - style: 마커 스타일로 그룹을 구분
  - size: 마커 크기로 그룹을 구분
  - kind: 플롯 종류 (scatter 또는 line)

```
sns.relplot(x="total_bill", y="tip", hue="sex", style="time", data=tips)
plt.show()
```

catplot

codingOn

- catplot() : 카테고리형 데이터의 분포를 시각화.
- 주요 옵션:
  - kind: 플롯 종류 (strip, swarm, box, violin, bar, count)
  - hue: 색상으로 그룹을 구분
  - col, row: 데이터의 서브셋을 생성하여 서브플롯으로 분할

```
sns.catplot(x="day", y="total_bill", hue="sex", kind="box", data=tips)
plt.show()
```

displot

codingOn

- displot() : 데이터의 분포를 시각화.
- 주요 옵션:
  - kind: 분포 그래프 종류 (hist, kde, ecdf)
  - bins: 히스토그램의 빈 수
  - hue: 색상으로 그룹을 구분

```
sns.displot(tips['total_bill'], bins=30, kde=True)
plt.show()
```

heatmap

- heatmap() : 2차원 데이터(매트릭스)를 히트맵 형태로 시각화.
- 주요 옵션:
  - annot: 셀에 값을 표시 여부 (True / False)
  - fmt: 셀에 표시될 값의 포맷
  - cmap: 컬러맵 지정

```
import numpy as np
data = np.random.rand(10, 10)
sns.heatmap(data, annot=True, fmt=".2f", cmap="coolwarm")
plt.show()
```

pairplot

codingOn

- pairplot() : 여러 변수 간의 관계를 한 번에 시각화
- 주요 옵션:
  - hue: 색상으로 그룹을 구분
  - diag_kind: 대각선에 표시할 그래프 종류 (kde, hist)

```
sns.pairplot(tips, hue="sex")
plt.show()
```

regplot

codingOn

- regplot() : 회귀선이 포함된 산점도를 생성
- 주요 옵션:
  - x, y: x축과 y축에 사용할 데이터 열
  - hue: 색상으로 그룹을 구분
  - order: 다항 회귀의 차수

```
sns.regplot(x="total_bill", y="tip", data=tips)
plt.show()
```

실습1. 데이터 분석 후 그래프 그리기

codingOn

- penguins 데이터 셋 이용하여 아래 조건의 그래프 생성하세요
- 조건
  - 펭귄의 종(species)별 평균 몸무게(body_mass_g)를 막대 그래프로 나타내세요.
  - 부리 길이(bill_length_mm)와 부리 깊이(bill_depth_mm)의 관계를 산점도로 시각화하고, 종(species)별로 색상을 다르게 표시하세요.
  - 펭귄의 섬(island)에 따라 몸무게의 분포를 violinplot으로 시각화하세요.

실습2. 데이터 분석 후 그래프 그리기

codingOn

- flights 데이터 셋 이용하여 아래 조건의 그래프 생성하세요
- 조건
  - 연도(year)별 승객 수(passengers)의 평균을 꺾은선 그래프로 나타내세요.
  - 연도와 월별(month) 승객 수를 히트맵으로 시각화하세요.
  - 특정 연도(예: 1958년)의 월별 승객 수를 막대 그래프로 나타내세요.

실습3. 데이터 분석 후 그래프 그리기

codingOn

- titanic 데이터 셋 이용하여 아래 조건의 그래프 생성하세요
- 조건
  - 탑승 클래스(class)와 생존 여부(survived) 간의 관계를 catplot으로 시각화하세요.
  - 나이(age)의 분포를 생존 여부(survived)에 따라 kdeplot으로 시각화하세요.

codingOn

복.습.철.저

수고하셨습니다