Week2-5

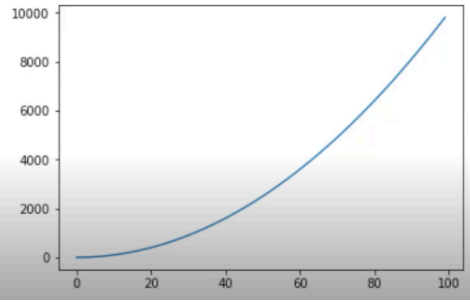
1. 학습정리
    1. 시각화도구
        1. **matplitlib**

matplotlib overview

- pyplot 객체를 사용하여 데이터를 표시
- pyplot 객체에 그래프들을 쌓은 다음 flush

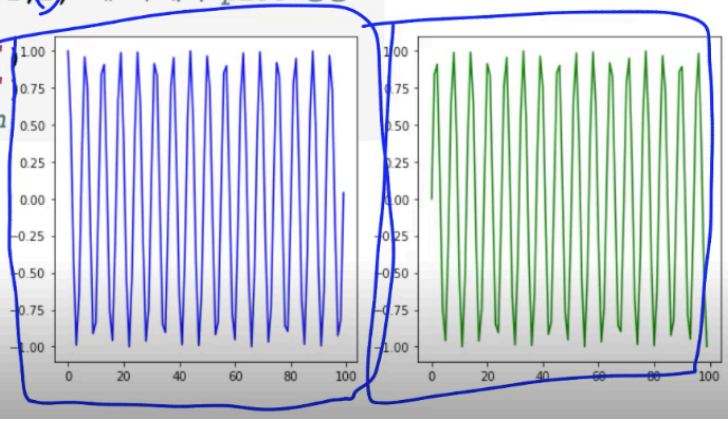
```
import matplotlib.pyplot as plt

X = range(100)
Y = [value**2 for value in X]
plt.plot(X, Y)
plt.show()
```




```
fig = plt.figure() # figure 반환
fig.set_size_inches(10,5) # 크기지정
ax_1 = fig.add_subplot(1,2,1) # 두개의 plot 생성
ax_2 = fig.add_subplot(1,2,2)  # 두개의 plot 생성

ax_1.plot(X_1, Y_1, c="b")
ax_2.plot(X_2, Y_2, c="g")
plt.show() # show & flush
```



set color

matplotlib

- color 속성을 사용
- float : 흑백, rgb color, predefined color 사용

```
plt.plot(X_1, Y_1, color="#eeefff")
plt.plot(X_2, Y_2, color="r")

plt.show()
```

1. 그 외 다양한 거 지정 가능
2. 스타일 적용으로 한 번에
3. show() 하면 flush 되서 메모리에서 사라짐
4. 그래프
    1. 산점도
        1. scatter()
    2. 막대그래프
        1. bar()
    3. 히스토그램 (분포차트)
        1. hist()

```
plt.style.use("ggplot")      # 스타일적용
```

2. **seaborn**
    1. 통계적인 데이터 시각화
    2. matplotlib 를 좀 더 쉽게 다루기 위함
    3. matplotlib 의 wrapper
    4. 간단한 코드 + 예쁜 코드
    5. basic plots

seaborn - basic plots

seaborn

- matplotlib와 같은 기본적인 plot
- 손쉬운 설정으로 데이터 산출
- lineplot, scatterplot, countplot 등

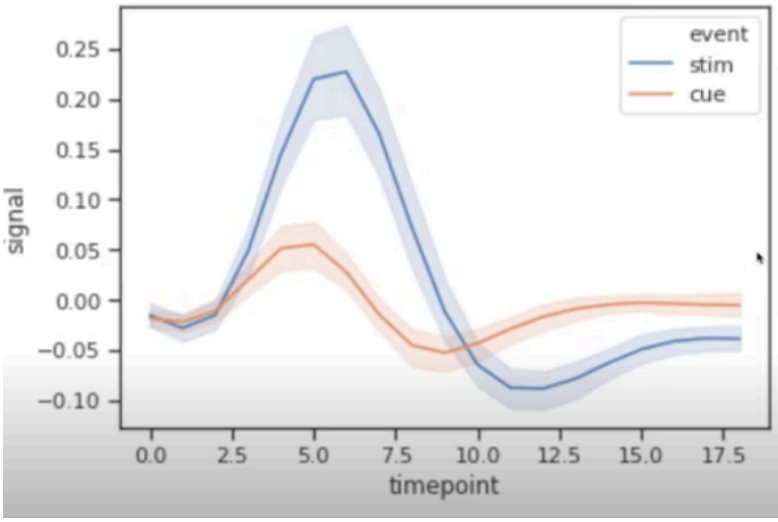
basic / multiplot

```
sns.lineplot(x="total_bill", y="tip", data=tips)
```

1. lineplot
    1. 기본적으로 데이터를 정렬 해 줌
    2. 분포도 그려줌
    3. hue
        1. 카테고리컬 데이터에 따라 데이터가 어떻게 변하는지 볼 수 있음

```
sns.lineplot(x="timepoint", y="signal", hue="event", data=fmri)
```

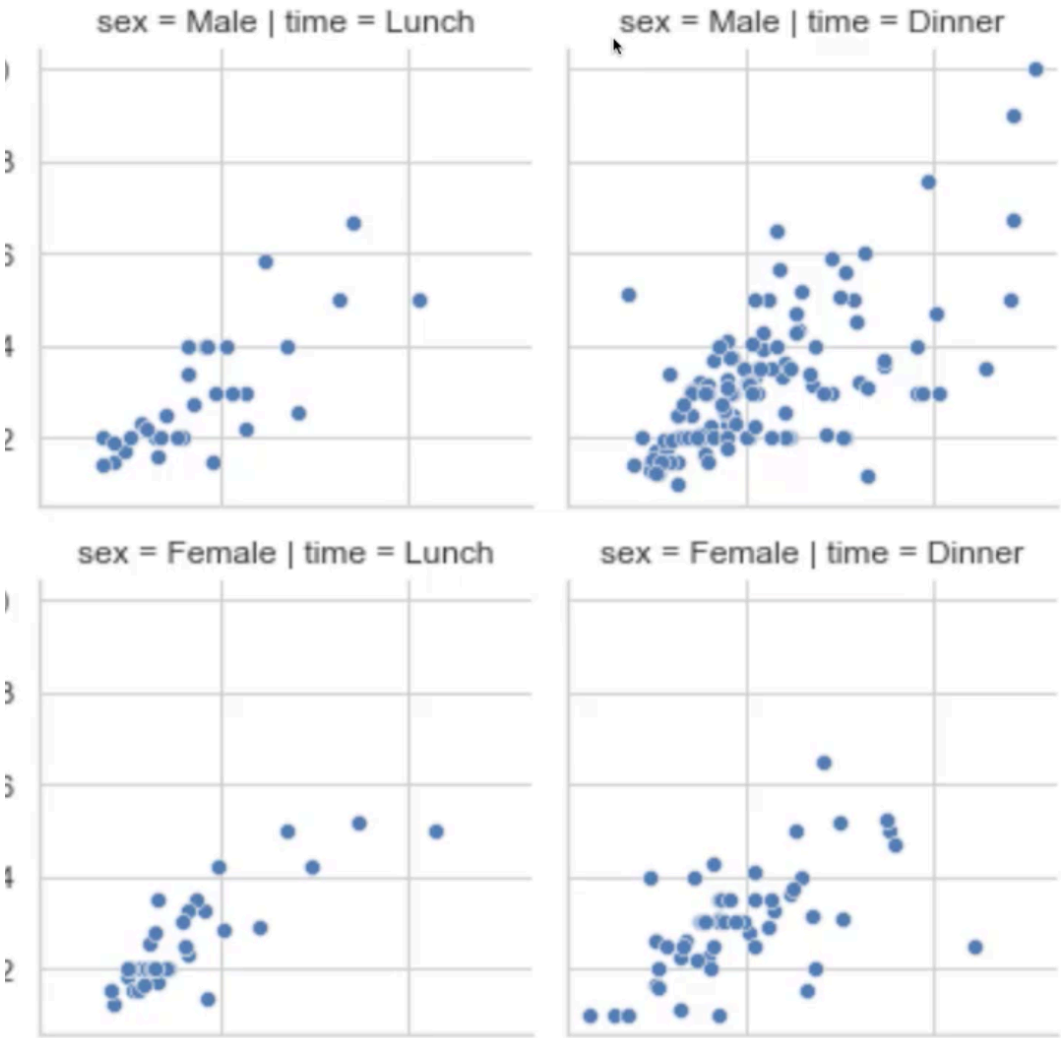
```
matplotlib.axes._subplots.AxesSubplot at 0x1a24a1bac8>
```



2. scatterplot
    1. 산점도
3. regplot
    1. 산점도에 선형회귀 라인을 그려줌
4. countplot
    1. 카테고리 데이터의 개수를 세어줌
5. barplot
    1. 막대그래프
6. distplot
    1. 분포 그래프
7. Facetgrid
    1. 카테고리에 따라 그리드 나눔
    2. map 으로 그림

```
g = sns.FacetGrid(tips, col="time",  row="sex")
g.map(sns.scatterplot, "total_bill", "tip")
```

```
aborn.axisgrid.FacetGrid at 0x7fb720f7e790>
```



6. 스타일 지정

```
import seaborn as sns

sns.set(style="darkgrid")
```

7.

3. **통계학**

모수가 뭐에요?

- 통계적 모델링은 적절한 가정 위에서 확률분포를 추정(inference)하는 것이 목표이며, 기계학습과 통계학이 공통적으로 추구하는 목표입니다
- 그러나 유한한 개수의 데이터만 관찰해서 모집단의 분포를 정확하게 알아낸다는 것은 불가능하므로, 근사적으로 확률분포를 추정할 수 밖에 없습니다
- 데이터가 특정 확률분포를 따른다고 선험적으로(a priori) 가정한 후 그 분포를 결정하는 모수(parameter)를 추정하는 방법을 모수적(parametric) 방법론이라 합니다
- 특정 확률분포를 가정하지 않고 데이터에 따라 모델의 구조 및 모수의 개수가 유연하게 바뀌면 비모수(nonparametric) 방법론이라 부릅니다

기계학습의 많은 방법론은 비모수 방법론에 속합니다

1. 확률분포 가정법

확률분포 가정하기: 예제

- 확률분포를 가정하는 방법: 우선 히스토그램을 통해 모양을 관찰합니다
  - 데이터가 2개의 값(0 또는 1)만 가지는 경우 → 베르누이분포
  - 데이터가 n개의 이산적인 값을 가지는 경우 → 카테고리분포
  - 데이터가 $[0,1]$ 사이에서 값을 가지는 경우 → 베타분포
  - 데이터가 0 이상의 값을 가지는 경우 → 감마분포, 로그정규분포 등
  - 데이터가 $\mathbb{R}$ 전체에서 값을 가지는 경우 → 정규분포, 라플라스분포 등
- 기계적으로 확률분포를 가정해서는 안 되며, 데이터를 생성하는 원리를 먼저 고려하는 것이 원칙입니다

각 분포마다 검정하는 방법들이 있으므로
모수를 추정한 후에는 반드시 검정을 해야 한다

1. 데이터를 먼저 관찰한 후 어떤 확률 분포가 적절한 지 선택해야 함

데이터로 모수를 추정해보자!

- 데이터의 확률분포를 가정했다면 모수를 추정해볼 수 있습니다
- 정규분포의 모수는 평균 μ 과 분산 σ^2 으로 이를 추정하는 통계량(statistic)은 다음과 같다:

$$\underbrace{\bar{X} = \frac{1}{N}\sum_{i=1}^{N} X_i}_{\text{표본평균}} \qquad \underbrace{S^2 = \frac{1}{N-1}\sum_{i=1}^{N}(X_i - \bar{X})^2}_{\text{표본분산}}$$

- 통계량의 확률분포를 표집분포(sampling distribution)라 부르며, 특히 표본평균의 표집분포는 N 이 커질수록 정규분포 $\mathcal{N}(\mu, \sigma^2/N)$ 를 따릅니다

이를 **중심극한정리(Central Limit Theorem)**이라 부르며, 모집단의 분포가 정규분포를 따르지 않아도 성립합니다

1. 표본분포는 데이터를 많이 모아도 정규분포가 되지는 않음
2. 표집분포 : 통계량의 확률분포
    1. 표집분포는 데이터가 커지면 정규분포를 따름 (중심극한정리)
2. 최대 가능도 추정법

최대가능도 추정법

- 표본평균이나 표본분산은 중요한 통계량이지만 확률분포마다 사용하는 모수가 다르므로 적절한 통계량이 달라지게 됩니다
- 이론적으로 가장 가능성이 높은 모수를 추정하는 방법 중 하나는 최대가능도 추정법(maximum likelihood estimation, MLE)입니다

$$\hat{\theta}_{\text{MLE}} = \underset{\theta}{\operatorname{argmax}}\ L(\theta; \mathbf{x}) = \underset{\theta}{\operatorname{argmax}}\ P(\mathbf{x}|\theta)$$

가능도(likelihood) 함수는 모수 θ 를 따르는 분포가 $\mathbf{x}$ 를 관찰할 가능성을 뜻하지만 확률로 해석하면 안됩니다

- 데이터 집합 $\mathbf{X}$ 가 독립적으로 추출되었을 경우 로그가능도를 최적화합니다

$$L(\theta; \mathbf{X}) = \prod_{i=1}^{n} P(\mathbf{x}_i|\theta) \quad \Rightarrow \quad \log L(\theta; \mathbf{X}) = \sum_{i=1}^{n} \log P(\mathbf{x}_i|\theta)$$

1. 확률분포마다 사용하는 모수가 다르므로 적절한 통계량이 달라짐
2. 이론적으로 가장 가능성이 높은 모수를 추정하는 방법

왜 로그가능도를 사용하나요?

- 로그가능도를 최적화하는 모수 θ 는 가능도를 최적화하는 MLE 가 됩니다
- 데이터의 숫자가 적으면 상관없지만 만일 데이터의 숫자가 수억 단위가 된다면 컴퓨터의 정확도로는 가능도를 계산하는 것은 불가능합니다
- 데이터가 독립일 경우, 로그를 사용하면 가능도의 곱셈을 로그가능도의 덧셈으로 바꿀 수 있기 때문에 컴퓨터로 연산이 가능해집니다
- 경사하강법으로 가능도를 최적화할 때 미분 연산을 사용하게 되는데, 로그가능도를 사용하면 연산량을 $O(n^2)$ 에서 $O(n)$ 으로 줄여줍니다
- 대게의 손실함수의 경우 경사하강법을 사용하므로 음의 로그가능도 (negative log-likelihood)를 최적화하게 됩니다

3. 확률분포의 거리
    1. 기계학습은 확률분포간의 거리를 줄이는 방향으로 학습

확률분포의 거리를 구해보자

- 기계학습에서 사용되는 손실함수들은 모델이 학습하는 확률분포와 데이터에서 관찰되는 확률분포의 거리를 통해 유도합니다
- 데이터공간에 두 개의 확률분포 $P(\mathbf{x})$, $Q(\mathbf{x})$ 가 있을 경우 두 확률분포 사이의 거리(distance)를 계산할 때 다음과 같은 함수들을 이용합니다
  - 총변동 거리 (Total Variation Distance, TV)
  - 쿨백-라이블러 발산 (Kullback-Leibler Divergence, KL)
  - 바슈타인 거리 (Wasserstein Distance)

2. 쿨백-라이블러 발산

쿨백-라이블러 발산

- 쿨백-라이블러 발산(KL Divergence)은 다음과 같이 정의합니다

$$\mathbb{KL}(P\|Q) = \overset{\text{이산확률변수}}{\sum_{\mathbf{x}\in\mathcal{X}}} P(\mathbf{x}) \log\left(\frac{P(\mathbf{x})}{Q(\mathbf{x})}\right) \qquad \mathbb{KL}(P\|Q) = \overset{\text{연속확률변수}}{\int_{\mathcal{X}}} P(\mathbf{x}) \log\left(\frac{P(\mathbf{x})}{Q(\mathbf{x})}\right) \mathrm{d}\mathbf{x}$$

- 쿨백 라이블러는 다음과 같이 분해할 수 있습니다

$$\mathbb{KL}(P\|Q) = \underbrace{-\mathbb{E}_{\mathbf{x}\sim P(\mathbf{x})}[\log Q(\mathbf{x})]}_{\text{크로스 엔트로피}} + \underbrace{\mathbb{E}_{\mathbf{x}\sim P(\mathbf{x})}[\log P(\mathbf{x})]}_{\text{엔트로피}}$$

- 분류 문제에서 정답레이블을 P, 모델 예측을 Q 라 두면 최대가능도 추정법은 쿨백-라이블러 발산을 최소화하는 것과 같습니다

            1. 확률분포사이의 거리를 최소화 하는 것 = 크로스 엔트로피 최대화
            2. = 로그가능도를 최대화 하는 것과 밀접한 관련이 있음
            3. 거리를 최소화 한다 = 최적화된 모수를 구한다
1. **피어세션**
    1. 강의 리뷰
        1. matplotlib 만 써봤는데 다른 걸 써봐서 좋았다
    2. week2-4 퍼더퀘스쳔 리뷰
        1. 연속형 확률 변수를 사용한 풀이
        2. 이산형 확률 변수를 사용한 풀이
    3. week2-5 퍼더퀘스쳔 풀이 공유
    4. 알고리즘 문제 리뷰
        1. https://leetcode.com/problems/h-index/submissions/
            1. 이분탐색을 이용한 풀이
            2. 파이써닉한 풀이
        2. https://leetcode.com/problems/delete-operation-for-two-strings/
            1. LCS 알고리즘
            2. 메모리절약법
        3.