

Jin-Soo Kim  
(jinsoo.kim@snu.ac.kr)

Systems Software &  
Architecture Lab.

Seoul National University

Jan. 6 – 17, 2020

*Python for Data Analytics*

# Matplotlib

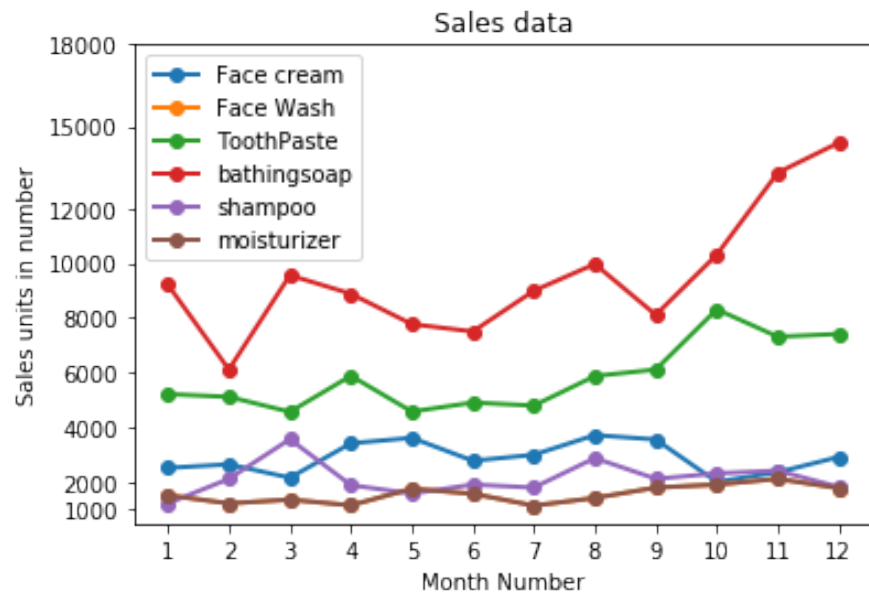


# Matplotlib lab

# Lab 1. 기본 그래프 응용

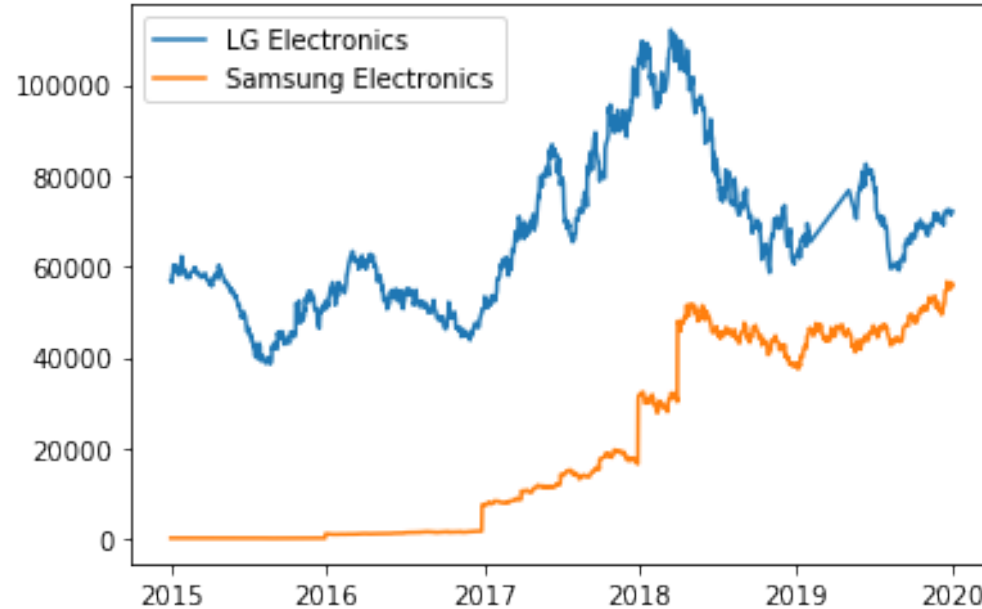
- 주어진 csv파일의 내용으로 아래 그림과 같은 결과를 얻기

- 결과



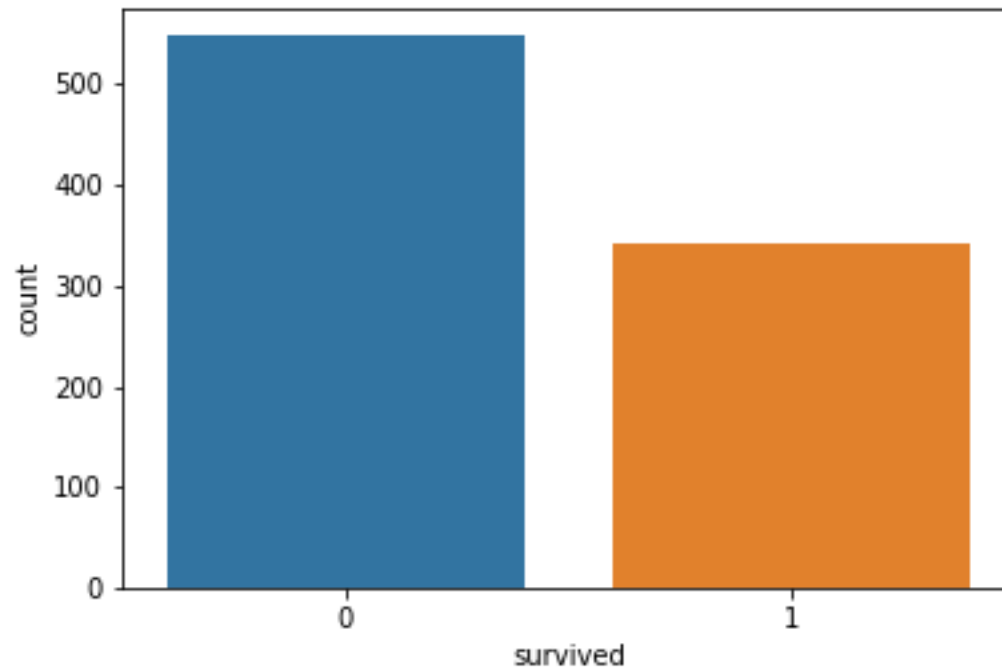
# Lab 2. 주식 그래프 그리기

- Web에 있는 주식 정보를 가져와서 주식 정보를 그래프로 그리기



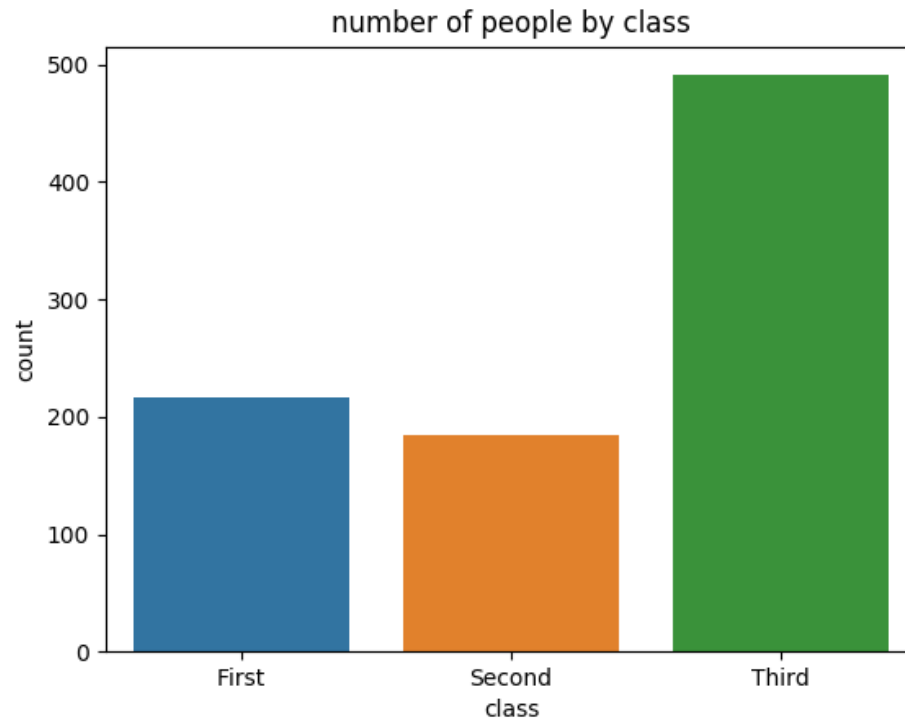
# Lab3 생존자와 사망자 수

- Titanic dataset의 정보를 활용해서 다양한 그래프를 그려본다.
- Seaborn의 countplot을 이용해서, class 별 탑승자의 수를 구한다.



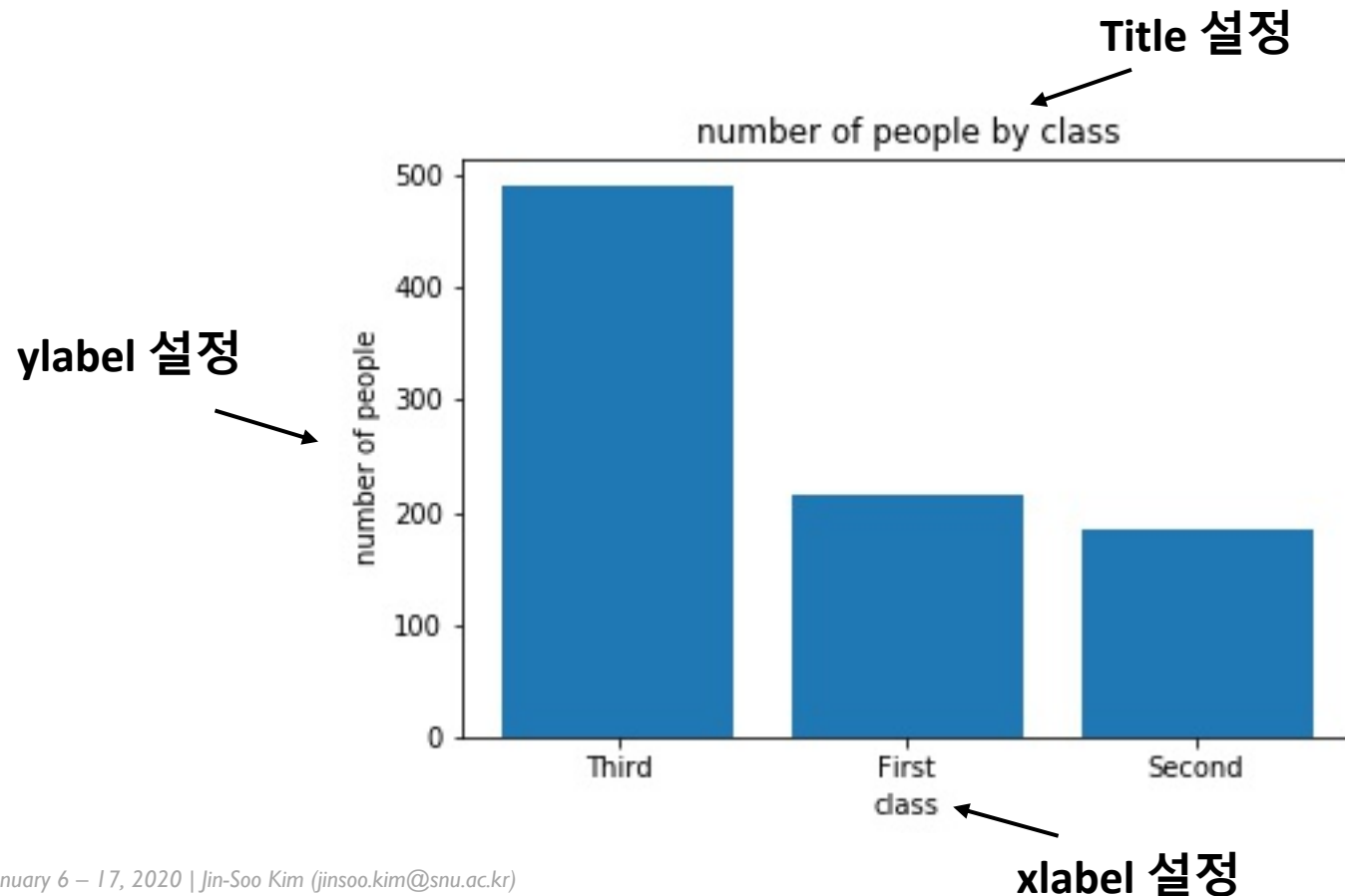
# Lab 4. class 별 탑승자 수 – seaborn

- Titanic dataset의 정보를 활용해서 다양한 그래프를 그려본다.
- Seaborn의 countplot을 이용해서, class 별 탑승자의 수를 구한다.



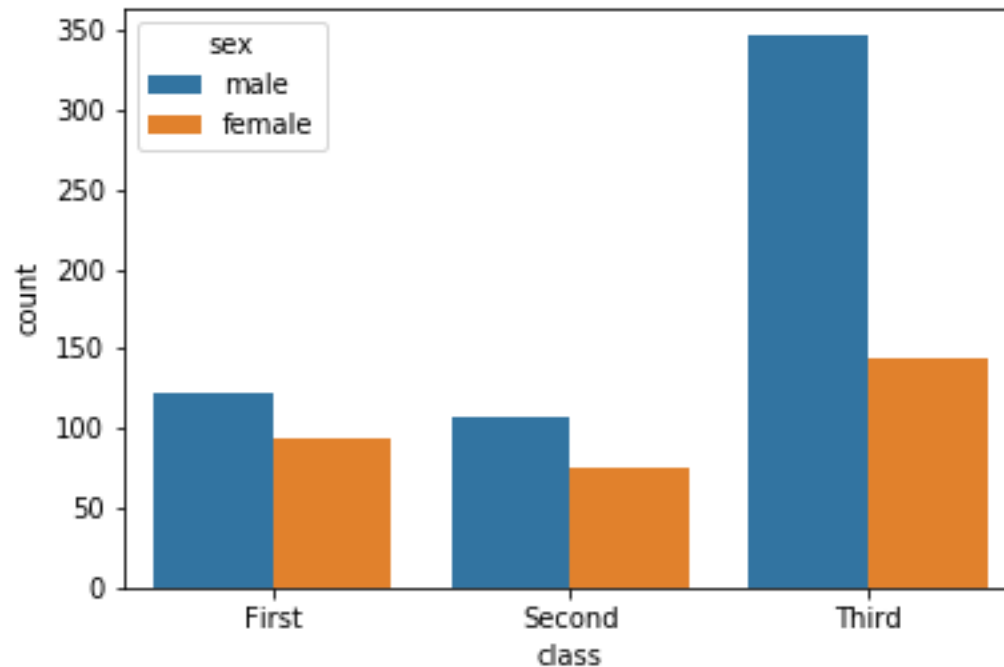
# Lab 5. class 별 탑승자 수 - matplotlib

- Titanic dataset의 정보를 활용해서 다양한 그래프를 그려본다.
- 이전 문제와 동일한 그림을 matplotlib을 사용해서 그리기



# Lab 6. class 별 탑승자 수 - 성별로 구분

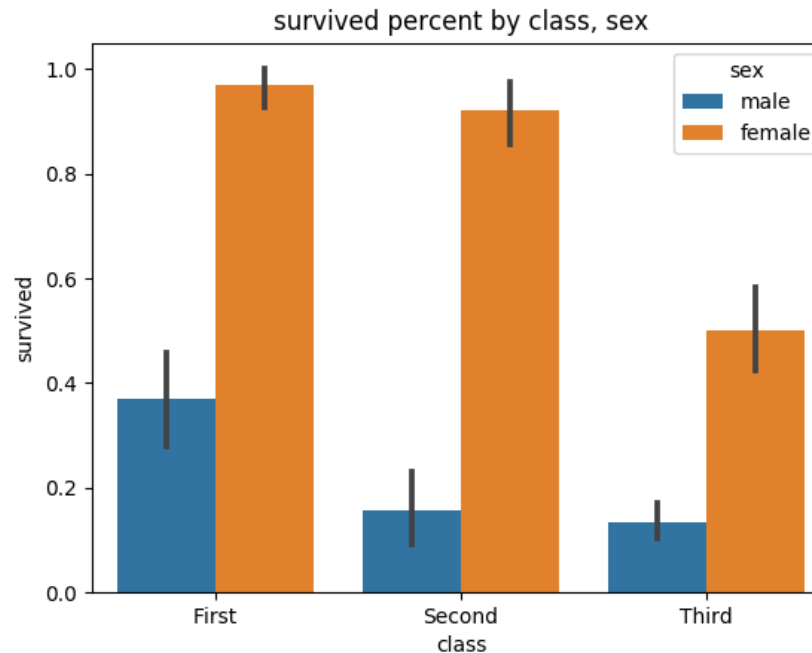
- Titanic dataset의 정보를 활용해서 다양한 그래프를 그려본다.
- Class 별 탑승자 수를 성별로 구분해서 그리기
  - sns.countplot, sns.barplot 두가지를 이용해서 같은 그래프를 그리기





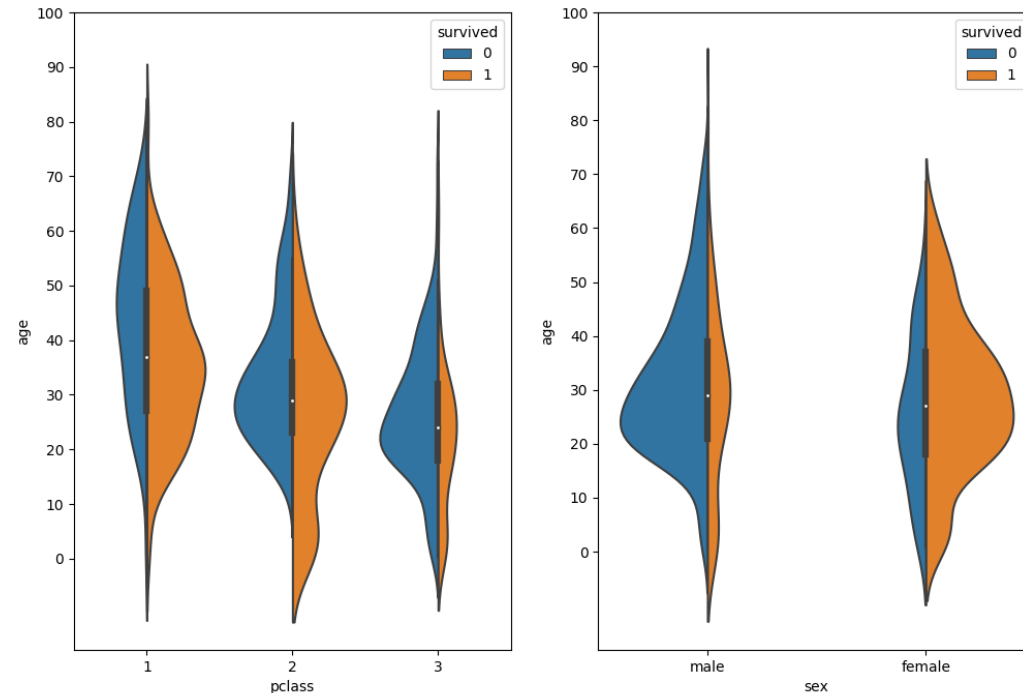
# Lab 7. class 별 성별에 따른 생존률

- Titanic dataset의 정보를 활용해서 다양한 그래프를 그려본다.
- Seaborn의 barplot을 이용해서, 탑승 class 별 성별에 따른 생존률을 구한다.



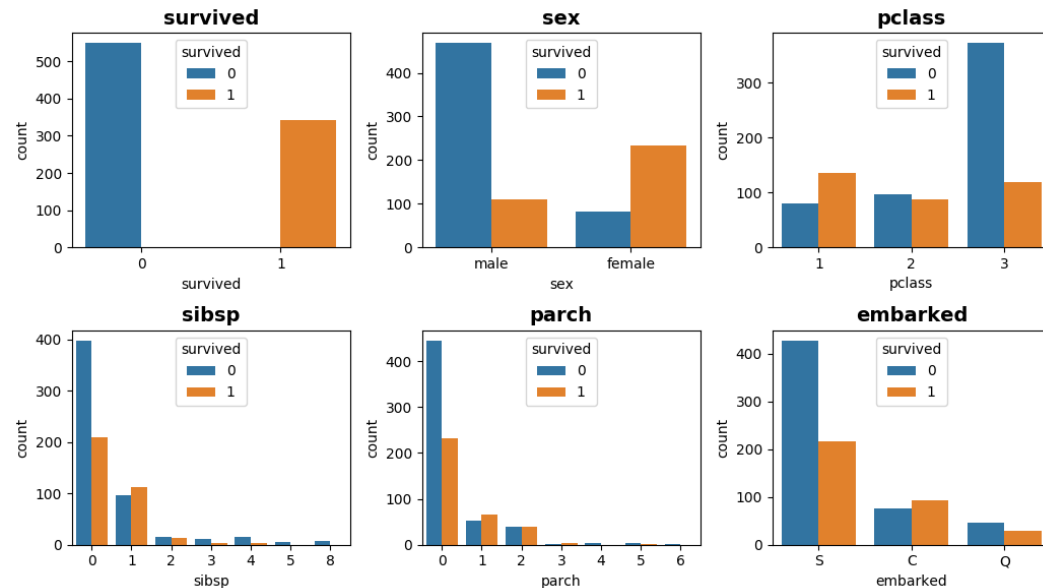
# Lab 8. titanic dataset

- Titanic dataset의 정보를 활용해서 다양한 그래프를 그려본다.
- Seaborn의 violinplot을 이용해서, 탑승 class 별 나이에 따라서 생존률을 구해본다.



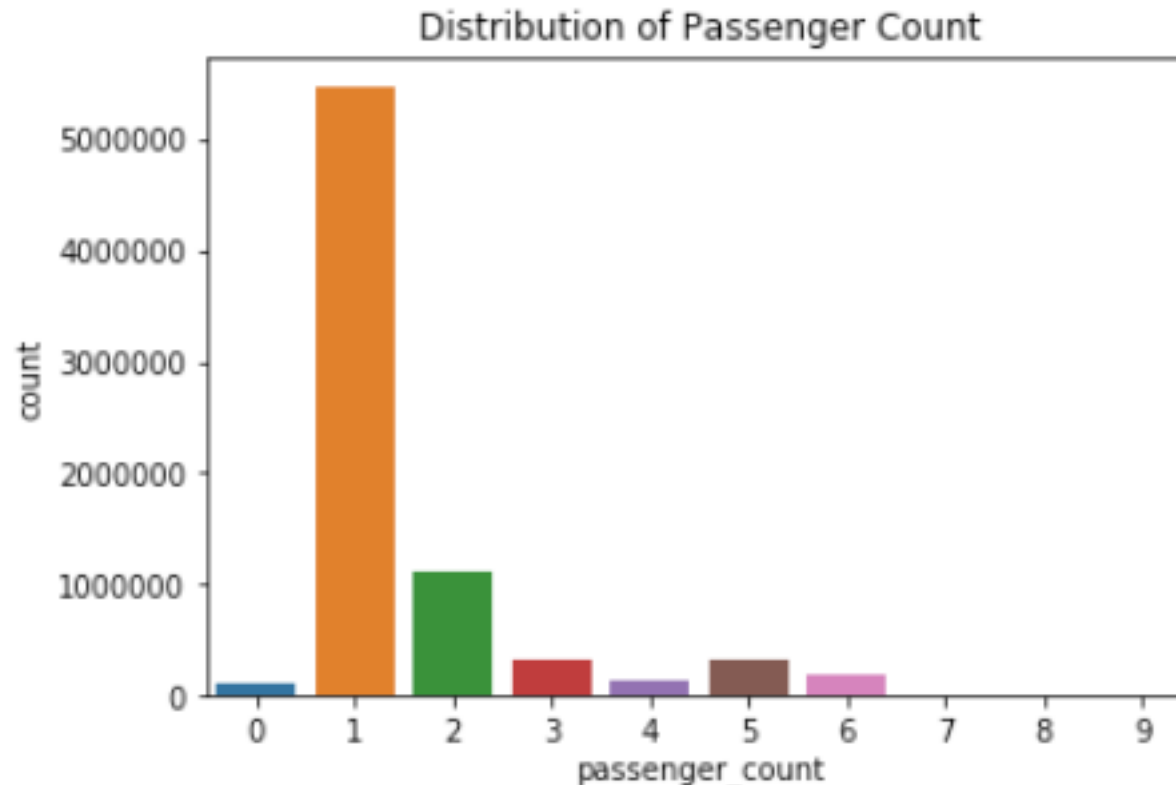
# Lab 9. titanic dataset

- Titanic dataset의 정보를 활용해서 다양한 그래프를 그려본다.
- Seaborn의 countplot을 이용해서 하나의 그래프에 아래 출력물과 같은 결과물을 만들어본다.
  - Hint : plt.subplots를 사용하면 아래와 같이 하나의 그림에 여러 그래프를 그릴 수 있다.



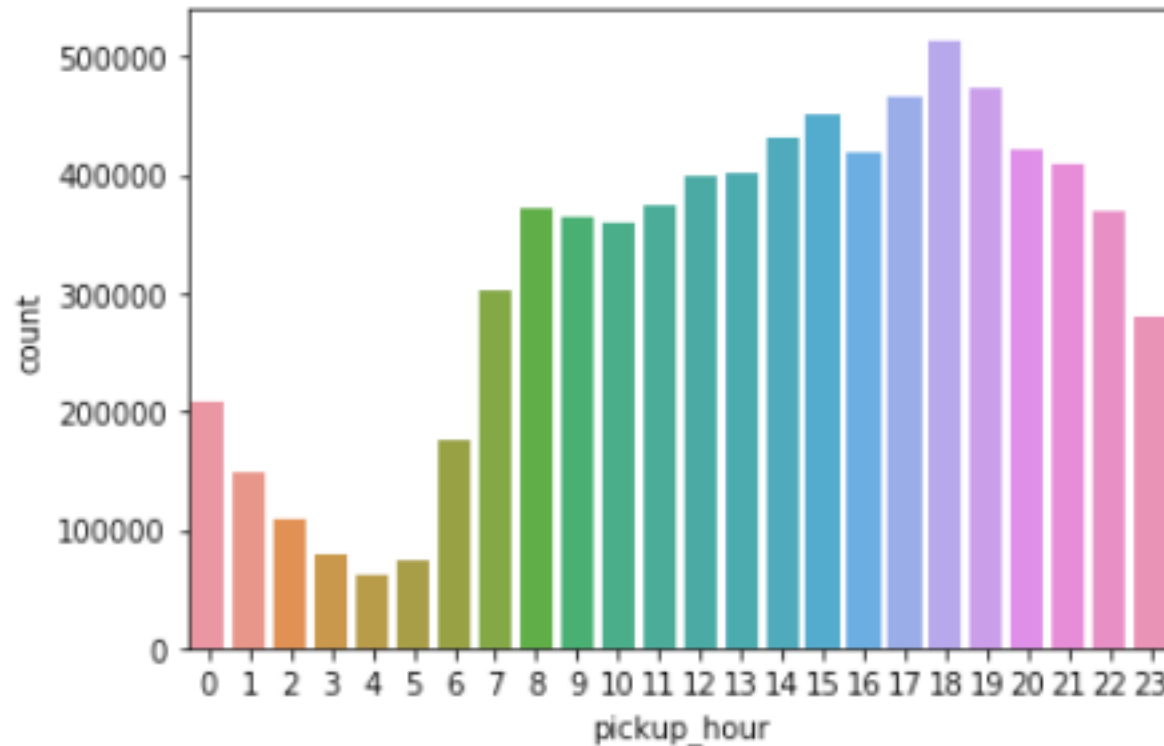
# Lab 10. passenger count

- 어제 사용했던 Newyork taxidataset를 분석한 데이터를 가지고 그림을 그려본다.
- Passenger count별 수를 그려본다.



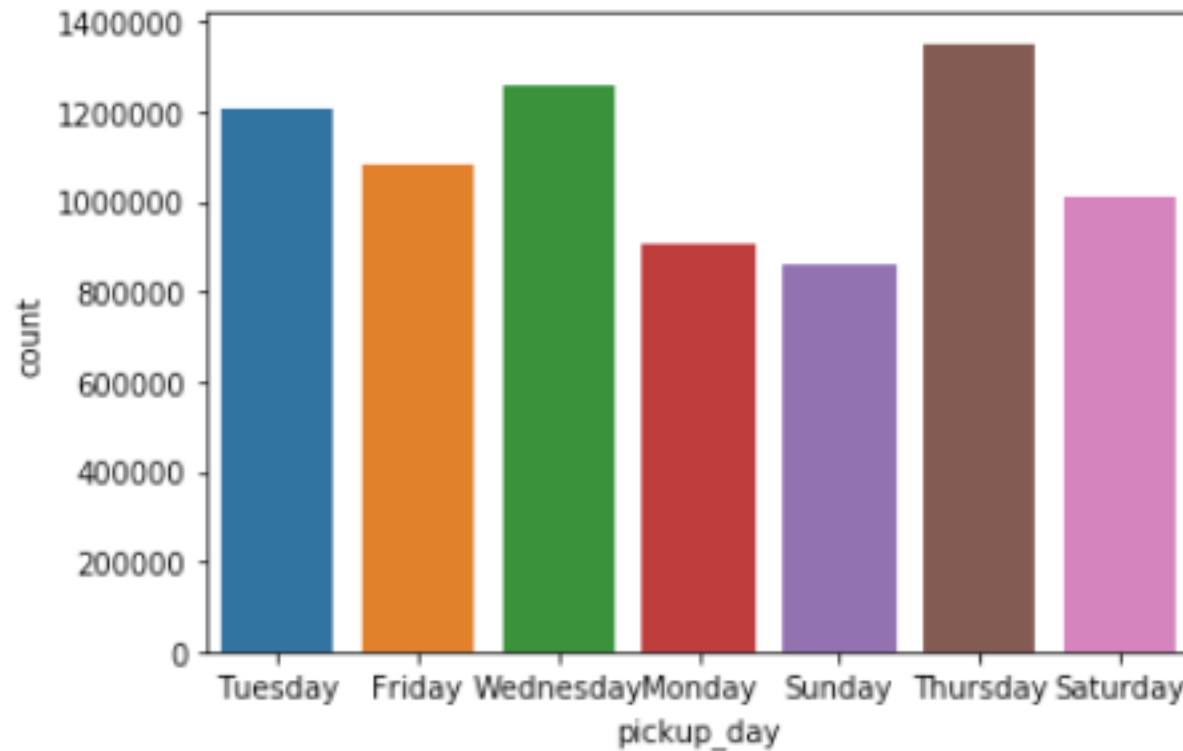
# Lab 11. pickup\_hour

- Pickup\_hour 데이터를 그려본다.



# Lab 12. pickup\_day

- Pickup\_day 데이터를 그려본다.



Additional lab

# Lab Numpy. 기계학습의 성능 평가 지표

- 기계학습에서 모델이나 패턴의 분류 성능 평가에 사용되는 지표인 Accuracy, Precision, Recall, F-measure(F1 score)를 구현해보기
- lab\_numpy\_1\_Answer.py 에 있는 함수의 EDIT HERE 부분을 구현
- 각 함수에서 넘겨주는 변수 label은 정답이고, pred는 예측값
- Label, pred에서 1은 positive이고, 0은 negative이다.
- numpy를 이용해서 함수를 구현해본다.



# Lab Numpy. 기계학습의 성능 평가 지표

■ **Accuracy** = 
$$\frac{\text{True Negative} + \text{True Positive}}{\text{Total}}$$

- Accuracy(정확도)는 전체에서 실제 positive를 positive라 예측한 것과 실제 negative를 negative라 예측한 것의 비율
- Hint :

		Predicted(pred)	
		Negative(0)	Positive(1)
Actual (label)	Negative(0)	True Negative	False Positive
	Positive(1)	False Negative	True Positive

# Lab Numpy. 기계학습의 성능 평가 지표

■ Precision = 
$$\frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$
$$= \frac{\text{True Positive}}{\text{Total Predicted Positive}}$$

- Precision은 예측이 positive라고 했을 때, 실제 positive의 비율

- Hint :  &&

**Predicted(pred)**

**Actual  
(label)**

	<b>Negative(0)</b>	<b>Positive(1)</b>
<b>Negative(0)</b>	True Negative	False Positive
<b>Positive(1)</b>	False Negative	True Positive

# Lab Numpy. 기계학습의 성능 평가 지표

■ Recall = 
$$\frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$
$$= \frac{\text{True Positive}}{\text{Total Actual Positive}}$$

- Recall은 실제 positive 중에 모델이 positive라 예측한 것의 비율

		Predicted(pred)	
		Negative(0)	Positive(1)
Actual (label)	Negative(0)	True Negative	False Positive
	Positive(1)	False Negative	True Positive

# Lab Numpy. 기계학습의 성능 평가 지표

- $F\_measure = \frac{2 * precision * recall}{precision + recall}$ 
  - 앞서 구현한 Precision, Recall의 조화평균
  - 앞서 구현한 precision과 recall을 이용해서 구현하면 편리하게 구현할 수 있음

Actual (label)	Predicted(pred)		
	Negative(0)	Positive(1)	
	Negative(0)	True Negative	False Positive
	Positive(1)	False Negative	True Positive