3장. 데이터 시각화

데이터 시각화의 목적

- 데이터 탐색(exploration)
- 데이터 전달(communication)

목표: 좋은 시각화와 그렇지 않은 것에 대한 분별력

1. 막대 그래프
2. 히스토그램
3. 선 그래프
4. 산점도

> matplotlib

```
# 기본 제공하는 라이브러리가 아니기 때문에 설치 필요
python -m pip install matplotlib
```

In [ ]:
```
from matplotlib import pyplot as plt
```

- matplotlib은 간단한 막대 그래프, 선 그래프, 또는 산점도를 그릴 때 널리 사용되는 라이브러리
- 인터랙티브한 시각화를 만들고 싶다면 플롯리(Plotly) 추천, 사용자 마우스에 반응하는 인터랙티브한 가능을 제공

> 막대 그래프

막대 그래프(bar chart)는 이산적인(discrete) 항목에 대한 변화를 나타낼 때 사용하면 좋음

예제. 영화별 아카데미 수상 횟수

In [ ]:
```
movies = ["Annie Hall", "Ben-Hur", "Casablance", "Candhi", "West Side Story"]
num_oscars = [5, 11, 3, 8, 10]
```

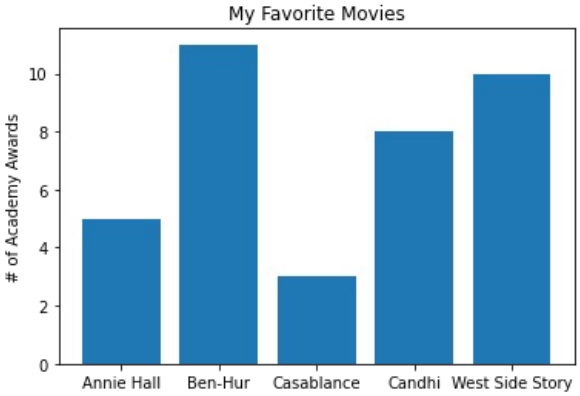
- x 축: 범주(영화 제목)
- y 축: 도수(수상 횟수)

In [ ]:
```
# 막대의 x 좌표는 [0, 1, 2, 3, 4], y 좌표는 [num_oscars]로 설정
plt.bar(range(len(movies)), num_oscars)

plt.title("My Favorite Movies")   # 제목을 추가
plt.ylabel("# of Academy Awards")  # y축에 레이블을 추가

# x축 각 막대의 중앙에 영화 제목을 레이블로 추가하자
plt.xticks(range(len(movies)), movies)

plt.show()
```



막대그래프를 이용해 히스토그램(histogram)을 그려보자

히스토그램이란 정해진 구간에 해당하는 항목의 개수를 보여줌으로서 값의 분포를 관찰할 수 있는 그래프 형태

양적 데이터들의 연속성을 반영하기 위해 막대 그래프들 사이의 간극이 없도록 그린다

In [ ]:
```
from collections import Counter
grades = [83, 95, 91, 87, 70, 0, 85, 82, 100, 67, 73,77, 0]

# 점수는 10점 단위로 그룹화한다. 100점은 90점대에 속한다
histogram = Counter(min(grade // 10 * 10, 90) for grade in grades)
histogram
```

Out[ ]:
```
Counter({80: 4, 90: 3, 70: 3, 0: 2, 60: 1})
```

In [ ]:
```
[min(grade // 10 * 10, 90) for grade in grades]
```

Out[ ]:
```
[80, 90, 90, 80, 70, 0, 80, 80, 90, 60, 70, 70, 0]
```

In [ ]:
```
83 / 10
```

Out[ ]:
```
8.3
```

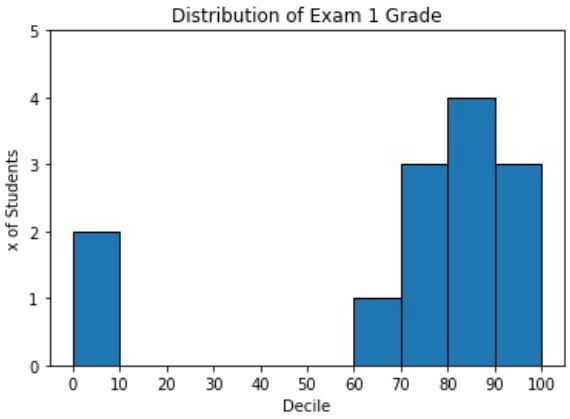
In [ ]:
```
83 // 10
```

Out[ ]:
```
8
```

In [ ]:
```
plt.bar([x+5 for x in histogram.keys()],   # 각 막대의 오른쪽으로 5 만큼 옮기고
        histogram.values(),                # 각 막대의 높이를 정해 주고
        10,                                # 너비는 10으로 하자.
        edgecolor = (0, 0, 0))             # 각 막대의테두리는 검은색으로 설정하자

plt.axis([-5, 105, 0, 5])

plt.xticks([10*i for i in range(11)])      # x축 레이블은 0, 10, ...., 100
plt.xlabel("Decile")                       # decile: 10분위수(어떤 집합체를 특정 변수에 따라 10개로 균등하게 나눈 집단의
plt.ylabel("x of Students")
plt.title("Distribution of Exam 1 Grade")
plt.show()
```



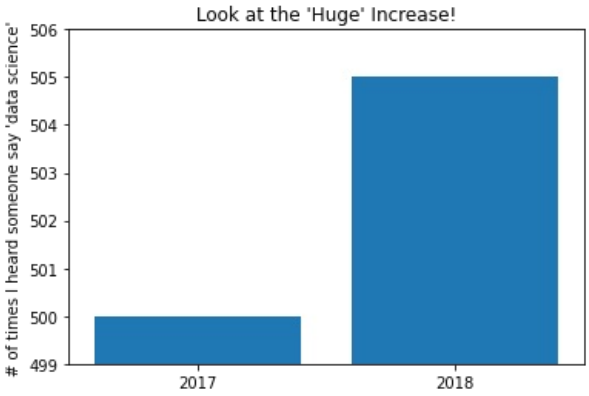
plt.axis 사용의 신중

In [ ]:
```
mentions = [500, 505]
years = [2017, 2018]

plt.bar(years, mentions, 0.8)
plt.xticks(years)
plt.ylabel("# of times I heard someone say 'data science'")

plt.ticklabel_format(useOffset = False)

# y축의 시작점을 4999로 잡으면 오해를 불러일으킬 수 있음
plt.axis([2016.5, 2018.5, 499, 506])
plt.title("Look at the 'Huge' Increase!")
plt.show()
```



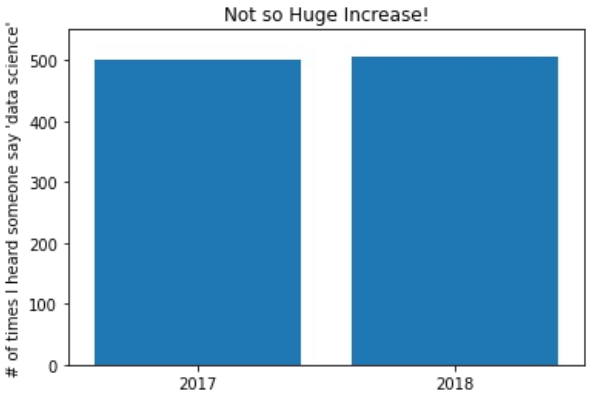

```
<Figure size 432x288 with 0 Axes>
```

In [ ]:
```
mentions = [500, 505]
years = [2017, 2018]

plt.bar(years, mentions, 0.8)
plt.xticks(years)
plt.ylabel("# of times I heard someone say 'data science'")

plt.ticklabel_format(useOffset = False)

# y축의 시작점을 4999로 잡으면 오해를 불러일으킬 수 있음
plt.axis([2016.5, 2018.5, 0, 550])
plt.title("Not so Huge Increase!")
plt.show()
```



> 선 그래프

선 그래프는 막대 그래프의 상단 중심부를 선분으로 연결하여 범주별 변화를 비교하는 그래프

시계열 데이터에 대한 정보 전달 시 유용

(참고) 시계열 데이터: 일정 기간에 대해 시간의 함수로 표현되는 데이터. 과거 5년간의 연도별 매상액, 3시간마다의 태풍 위치 등을 예로 들 수 있는데, 주로 예측 업무에 사용된다.

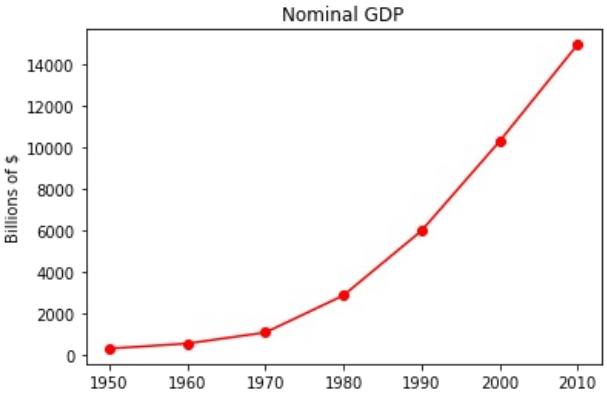
선 그래프 예제 1

1950년부터 10년 주기로 미국 국내총생산(GDP)에 대해 가시화해보자. 단위는 10억 달러이다.

In [ ]:
```
years = [1950, 1960, 1970, 1980, 1990, 2000, 2010]
gdp = [300.2, 543.3, 1075.9, 2862.5, 5979.6, 10289.7, 14958.3]

plt.plot(years, gdp, color = 'red', marker='o', linestyle='solid')

plt.title("Nominal GDP")
plt.ylabel("Billions of $")
plt.show()
```



선 그래프 예제 2

머신러닝에서 알고리즘을 학습시킬 때 학습시킨 알고리즘의 예측이 정답으로부터 떨어져 있는 오차를 재는 기준으로 편향(bias)과 분산(variance)를 사용한다.

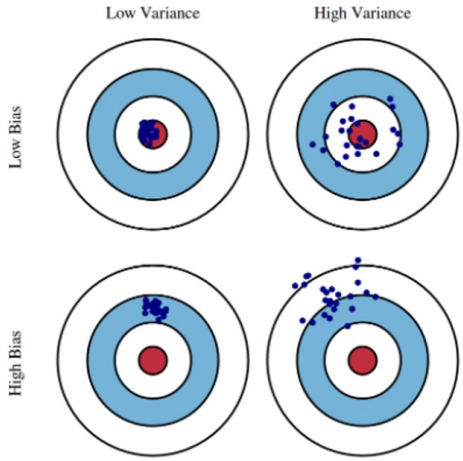


Fig 1: Graphical illustration of bias and variance.
Source: http://scott.fortmann-roe.com/docs/BiasVariance.html

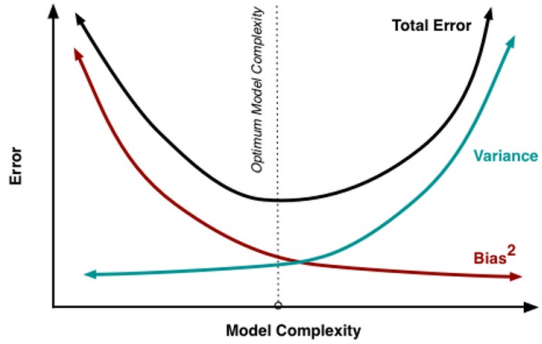
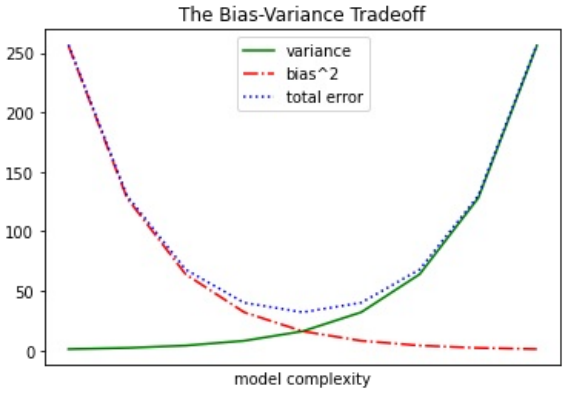


Fig 2: The variation of Bias and Variance with the model complexity. This is similar to the concept of overfitting and underfitting. More complex models overfit while the simplest models underfit.
Source: http://scott.fortmann-roe.com/docs/BiasVariance.html

In [ ]:
```
variance = [1, 2, 4, 8, 16, 32, 64, 128, 256]
bias_squared = [256, 128, 64, 32, 16, 8, 4, 2, 1]
total_error = [x + y for x, y in zip(variance, bias_squared)]
xs = [i for i, _ in enumerate(variance)]

plt.plot(xs, variance,     'g-',  label = 'variance')
plt.plot(xs, bias_squared, 'r-.', label = 'bias^2')
plt.plot(xs, total_error,  'b:',  label = 'total error')

plt.legend(loc = 9)
plt.xlabel("model complexity")
plt.xticks([])
plt.title("The Bias-Variance Tradeoff")
plt.show()
```



> 산점도

산점도(scatterplot)는 두 변수 사이의 연관관계를 2차원 평면 상에서 점으로 찍어 보여주는 그래프

예제

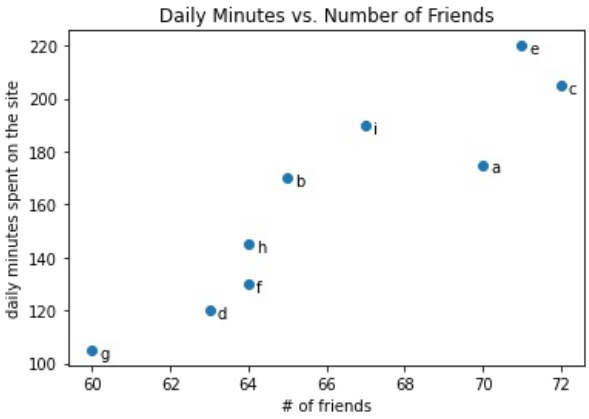
어느 웹사이트 회원의 친구수와 하룻동안의 사이트 이용시간 사이의 연관성

In [ ]:
```
friends = [70, 65, 72, 63, 71, 64, 60, 64, 67]
minutes = [175, 170, 205, 120, 220, 130, 105, 145, 190]
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i']

plt.scatter(friends, minutes)

for label, friend_count, minute_count in zip(labels, friends, minutes):
  plt.annotate(label,                          # annotate: 주석을 달다
         xy=(friend_count, minute_count),      #레이블을 데이터 포인트 근처에 두되
         xytext = (5, -5),                     # 약간 떨여져 있게 하자
         textcoords='offset points')

plt.title("Daily Minutes vs. Number of Friends")
plt.xlabel("# of friends")
plt.ylabel("daily minutes spent on the site")
plt.show()
```

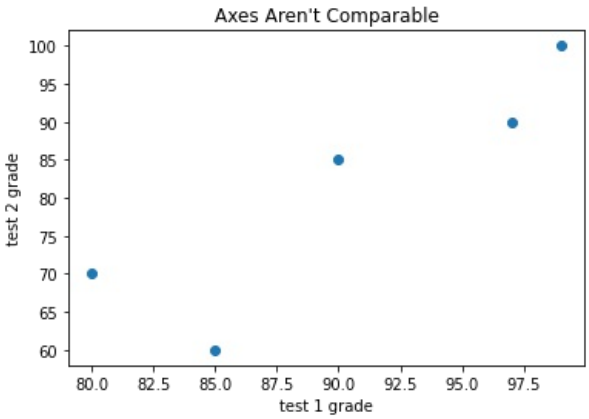


산점도 유의사항

matplotlib이 자동으로 축의 점위를 설정하게 되면 공정한 비교가 불가능할 수 잇음

In [ ]:
```
test_1_grades = [ 99, 90, 85, 97, 80]
test_2_grades = [100, 85, 60, 90, 70]

plt.scatter(test_1_grades, test_2_grades)
plt.title("Axes Aren't Comparable")
plt.xlabel("test 1 grade")
plt.ylabel("test 2 grade")
plt.show()
```



`plt.axis("equal")` 이라는 명령을 추가해 공정한 비교를 하자

In [ ]:
```
test_1_grades = [ 99, 90, 85, 97, 80]
test_2_grades = [100, 85, 60, 90, 70]

plt.scatter(test_1_grades, test_2_grades)
plt.title("Axes Aren't Comparable")
plt.axis("equal")
plt.xlabel("test 1 grade")
plt.ylabel("test 2 grade")
plt.show()
```