**bar graph – 사라지는 막대들**

본 글의 코드 중 캡쳐한 것은 코랩, 코드 블록은 vscode에서 실행한 코드입니다.

코드는 모두 [여기](https://github.com/keizikang/missing_bars_in_matplotlib)에서 보실 수 있습니다.

**세줄요약**

증상: matplotlib.pyplot.bar()로 막대 그래프를 그릴 때 막대가 사라지거나 막대 사이의 틈이 사라지는 현상이 있음

원인: bar()는 막대를 무조건 1픽셀 단위로만 그리는데, 막대 또는 틈이 너무 얇으면 bar()가 무시함

해결책: bar()의 파라미터 중 width를 조절하며, 필요에 따라 figsize, edgecolor를 설정하거나 Figure 창의 크기를 조절

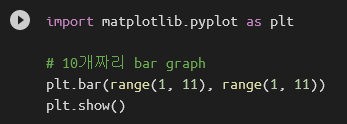
**감사의 글**

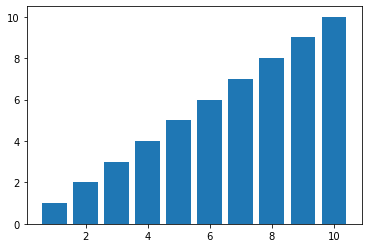
본 글의 아이디어를 주신 [페가님](https://jehyunlee.github.io/)께 감사 말씀 드립니다.

==

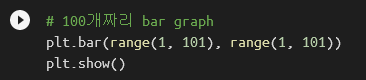
**이상한 막대 그래프**

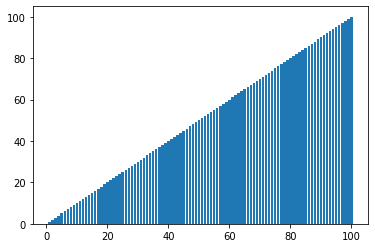
matplotlib의 bar()를 이용하여 간단히 막대 그래프를 그려보겠습니다.



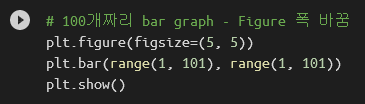


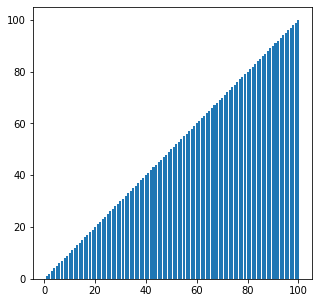
딱히 특별할 건 없네요. 이번엔 1부터 100까지 그려보겠습니다.



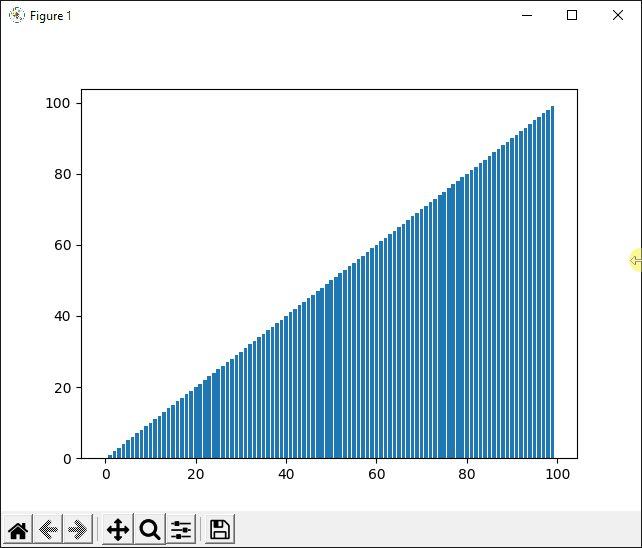


엇… 이게 뭐죠? 이상한 현상이 발생했습니다. 어디는 막대들이 붙어있고 어디는 떨어져 있네요. 자세히 보니 주기성이 있군요. 막대들이 서로 겹쳐서 그런 걸까요? figsize를 바꿔보죠.

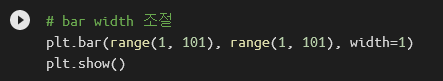




전혀 다른 패턴으로 이상해졌군요. vscode나 idle에서는 Figure 창을 띄워서 창의 폭을 동적으로 조절할 수 있습니다.



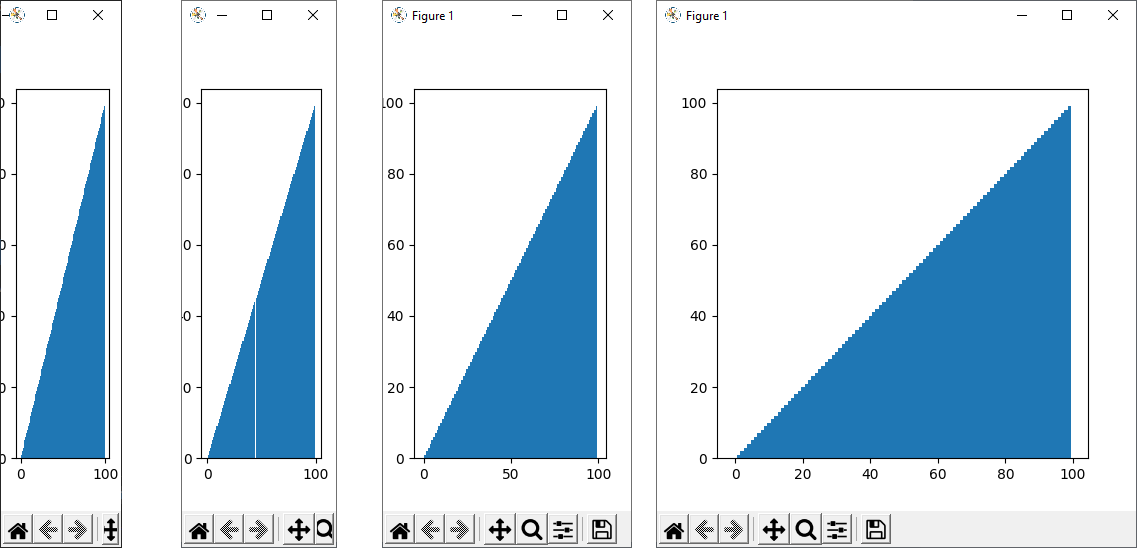
흥미롭군요. 어쨌든 가로 방향으로 무언가 문제가 생기는 것 같습니다. 우선 위 Figure 창의 폭은 100픽셀보다 훨씬 크므로 공간 부족의 문제는 아닙니다. 막대의 폭을 조절해보겠습니다. bar()의 width 파라미터를 조절하면 됩니다. width의 기본값은 0.8입니다. 자세한 내용은 [공식 문서](https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.bar.html)를 참고해주세요.



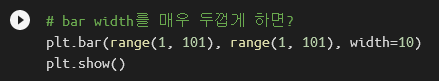
텍스트, 톱이(가) 표시된 사진

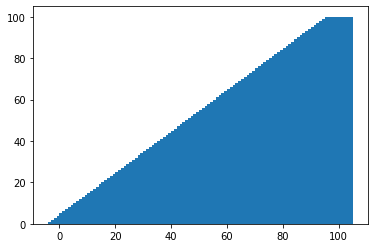
자동 생성된 설명

아하? 한방에 해결됐군요. Figure 창의 폭을 다양하게 만들어보겠습니다.



드물게 빈 공간이 생기기는 하지만 대체적으로 빈틈없이 빼곡하게 채워지는군요. 장난을 좀 쳐보겠습니다.



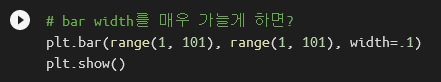


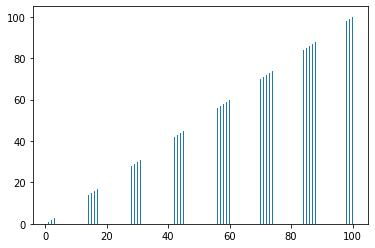
width가 어떻게 동작하는지 알 것 같군요. 비슷한 이슈는 이미 [stackoverflow에 올라온 적](https://stackoverflow.com/questions/56170270/matplotlib-pyplot-bar-seems-to-produce-incorrect-results)이 있기도 합니다.

==

**사라지는 막대들**

다른 방향으로 장난을 쳐보겠습니다.



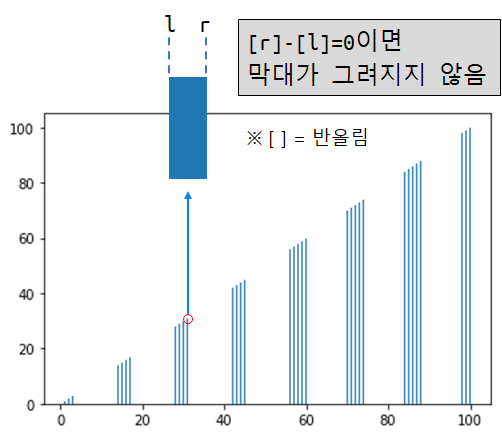


막대의 폭을 0.1로 엄청 가늘게 만들어봤습니다. 그 결과 100개의 막대가 생겨야 하는데 겨우 33개만 생겼습니다. 중간중간빈 곳이 많군요. 이 빈 곳은 실제로 비어있지는 않습니다. 확대해보면 분명 막대가 있습니다. 단지 표시만 되지 않을 뿐입니다. 그리고 역시 주기성이 있네요. 막대의 폭을 줄였더니 표시되는 막대의 개수가 줄어들고 빈 곳이 생긴다? 그리고 주기적이다? 가설을 하나 세워봤습니다.

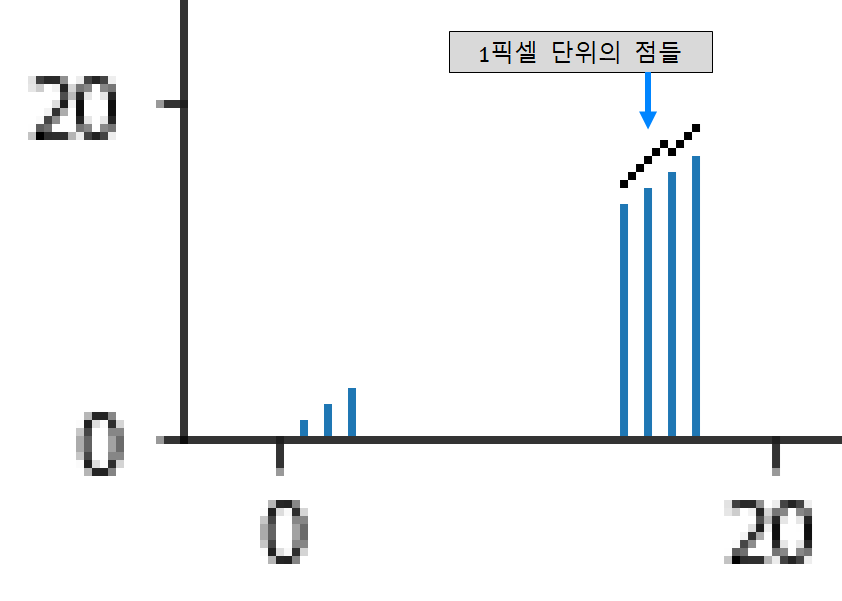
1. 각 막대 좌/우 경계edge의 **모니터 상** 위치를 막대 중앙과 width로부터 계산한다. (가설)

2. 그런데 bar()로 그려지는 막대는 무조건 폭이 1픽셀 단위로만 그려진다. (사실)

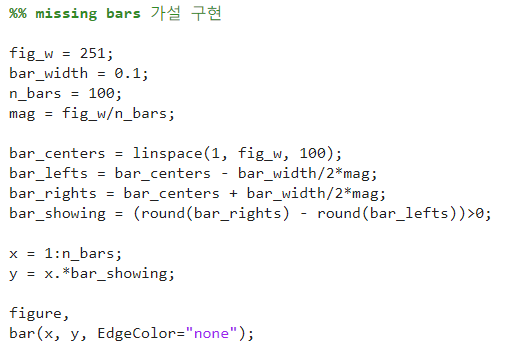
3. 막대의 좌/우 경계의 반올림 결과가 같으면 막대가 그려지지 않을 것이다. (가설)

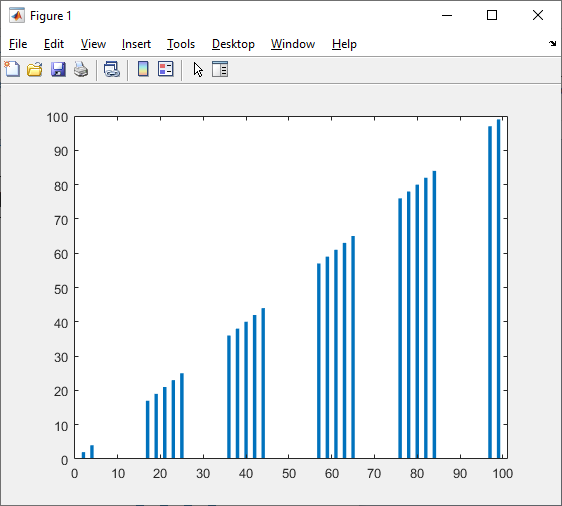


우선 2번은 가설이 아니라 사실입니다. Figure를 캡쳐해서 그림판에 붙여보면 바로 알 수 있습니다.



가설을 검증하기 위해 매트랩으로 가설을 구현해봤습니다.





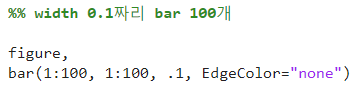
두둥-! 비슷한 그림이 만들어졌습니다. 위 매트랩 코드에서 fig\_w 값 251은 비슷한 그림이 나올 때까지 시행착오로 얻은 값입니다. matplotlib이 axes의 폭을 어떻게 정하는지 모르니까요. (아는 분 계시면 알려주세요.)

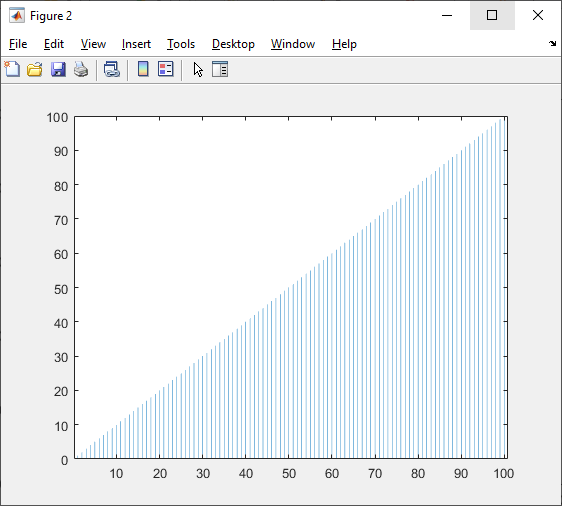
매트랩에서는 bar()의 EdgeColor 기본값이 black입니다. 그래서 비슷한 그림을 그리기 위해 EdgeColor를 none으로 설정했습니다. matplotlib의 bar()는 edgecolor의 기본값이 none입니다. 원래 white였는데 언젠가부터 바뀌었다고 하네요. 매트랩의 bar()도 막대의 폭 기본값은 0.8입니다.

==

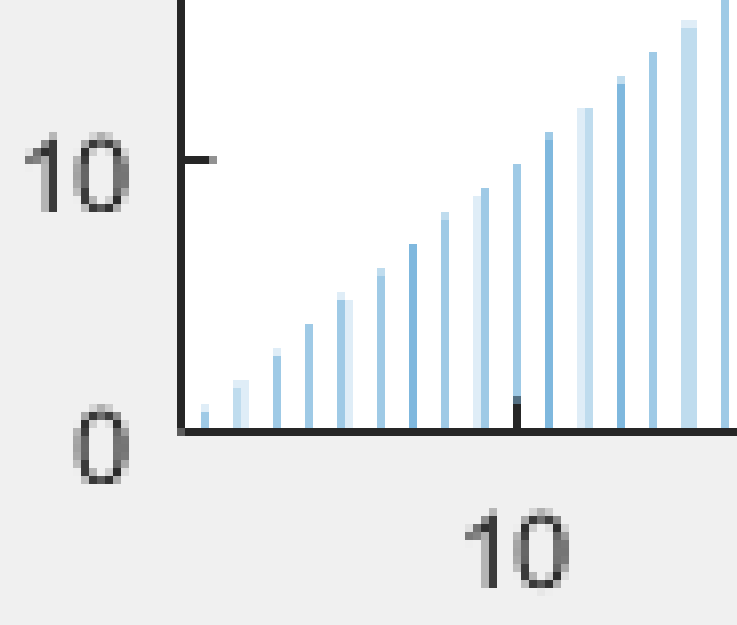
**matplotlib vs MATLAB: 막대를 표현하는 방법**

그럼 매트랩에서 width가 0.1인 bar 100개를 그리면 어떻게 될까요? matplotlib과 비슷한 현상이 생길까요?

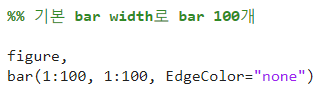


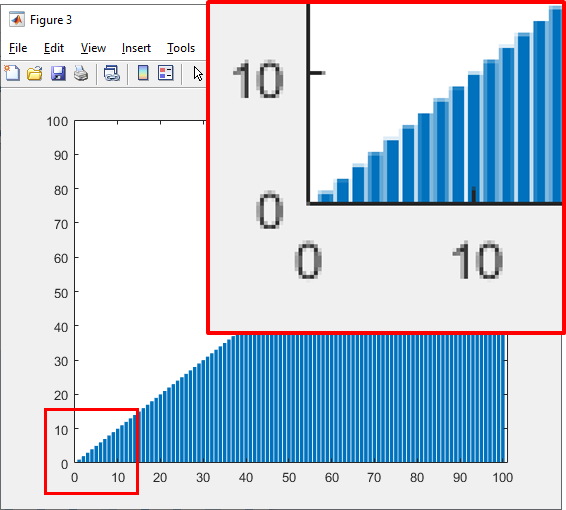


안 생기는군요. 막대 100개가 빠짐없이 모두 표시됐습니다. matplotlib과는 결정적인 차이가 있습니다. **매트랩은 막대를 1픽셀 단위로 그리지 않습니다.** 위 그래프를 캡쳐해서 확대해보면



막대마다 색깔이 다르고, 어떤 막대는 폭이 2픽셀입니다. 아마 2픽셀짜리 막대는 폭 0.1이 픽셀 경계에 걸쳤을 겁니다. 매트랩에서는 이렇게 **색깔을 이용해서 막대의 폭을 표현**합니다. 폭을 두껍게 해보면 바로 알 수 있습니다.





막대 폭이 0.1일 때에 비해 폭뿐만 아니라 색깔도 다릅니다. 이 색깔이 원래 막대의 FaceColor에 해당됩니다. 폭이 0.1일 때에는 FaceColor를 제대로 보여줄 공간이 없습니다. 대신 색을 연하게 함으로써 폭이 작음을 간접적으로 보여줍니다. 정리하자면, 매트랩과 matplotlib은 아래의 차이 때문에 막대 그래프가 다르게 표현됩니다.

\* matplotlib

\* bar()로 그리는 막대의 폭은 무조건 1픽셀 단위이다. (사실)

\* 막대의 색은 facecolor를 그대로 따른다. (사실)

\* 막대의 양쪽 edge와 가까운 정수(픽셀 위치)가 동일하면 막대를 그리지 않는다. (가설)

\* 매트랩

\* bar()로 그리는 막대의 폭은 1픽셀 단위가 아니다.

\* 폭이 좁은 것은 색깔을 연하게 함으로써 대신하며, 이 경우 막대의 색은 facecolor를 그대로 따르지 않을 수 있다.

\* 막대가 픽셀 경계에 걸치면 2픽셀 폭으로 그리고 막대의 색깔을 조절한다.

이왕 하는 김에 막대 폭을 다양하게 바꿔가며 그려봤습니다. 사실 그린 게 아까워서 붙이는 겁니다. 아래는 코드를 vscode와 매트랩에서 실행한 후 비슷한 그림이 나오도록 Figure 창의 폭을 조절한 것입니다.

# bar width를 다양하게 바꿔가며 테스트

# colab과 vscode에서 그래프가 다르게 나올 수 있음

x = range(100)

y = range(100)

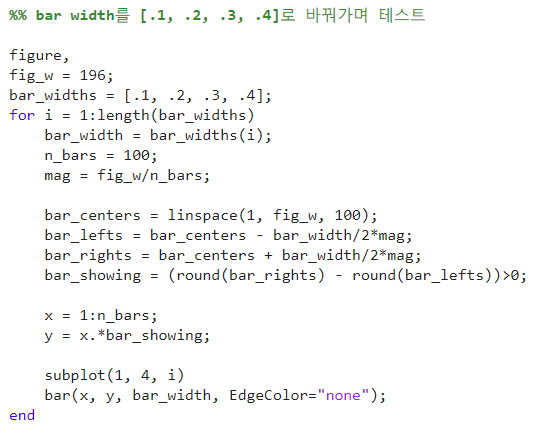
fig, ax = plt.subplots(1, 4, figsize=(15, 5))

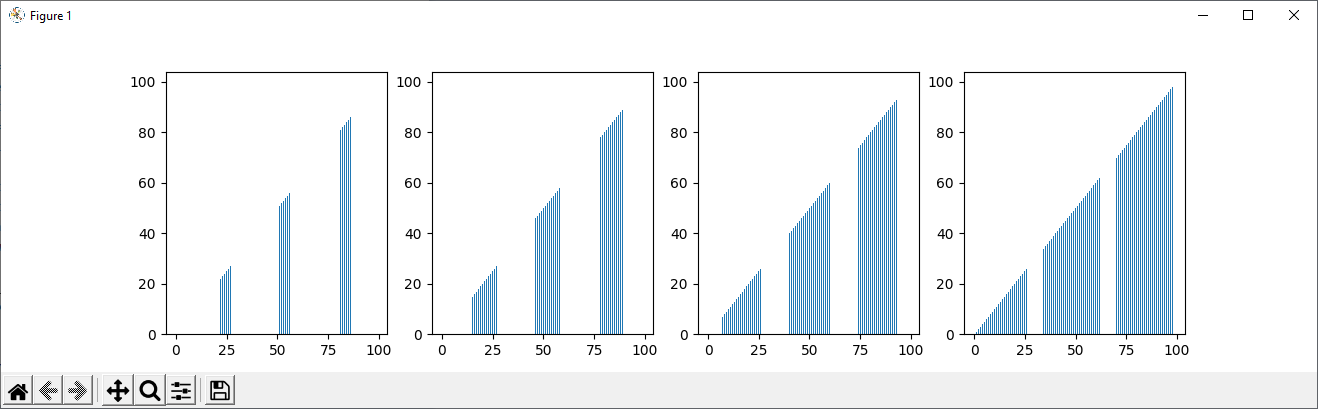
widths = [.1, .2, .3, .4]

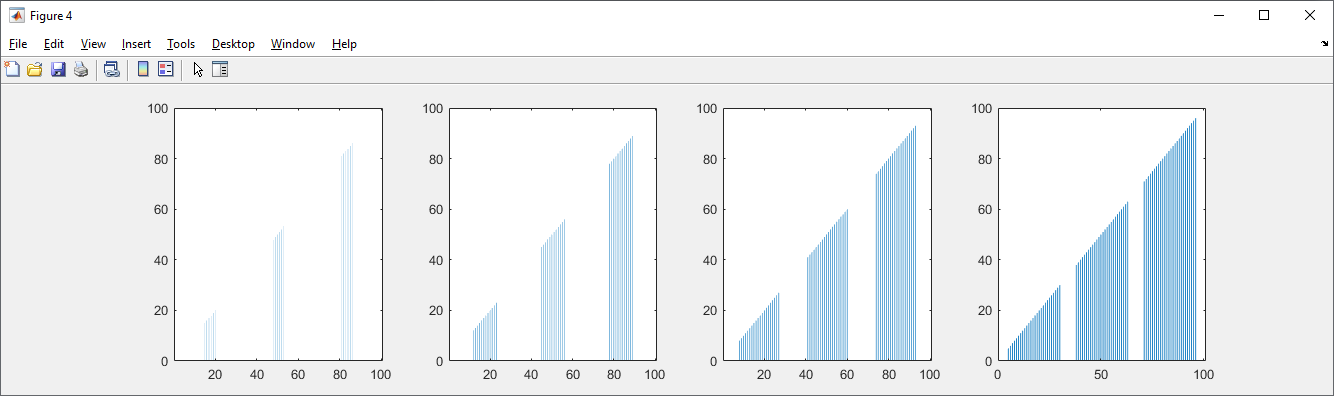
for i, w in enumerate(widths):

    ax[i].bar(x, y, width=w)

plt.show()







아름답네요. 가설은 어느 정도 검증이 된 것 같습니다. 역시 그래프는 매트랩이 더 잘 그려주는군요. 매트랩 흥해라!

==

**두 현상은 사실 하나의 현상이었다**

지금까지 matplotlib.pyplot.bar()로 막대 그래프를 그릴 때 나타나는 두 가지 현상을 말씀드렸습니다.

현상1: 막대가 많을 때 막대 사이의 간격이 불균일하게 표시됨

현상2: 막대의 폭width이 아주 작을 때 막대가 표시되지 않음

공통의 원인은 matplotlib이 막대를 무조건 1픽셀 단위로만 그린다는 것입니다. 그리고 위의 두 현상은 사실 **한 가지 현상**입니다. 왜냐하면 **현상1은 다르게 표현하자면 막대 사이의 틈이 사라지는 현상**이거든요. 즉,

현상1: 막대가 많을 때 막대 사이의 틈이 1픽셀이 되지 않아 틈이 사라짐

현상2: 막대가 아주 얇을 때 막대가 1픽셀이 되지 않아 막대가 사라짐

입니다. 존재存在와 부재不在는 동전의 양면이랄까요? 관점을 바꾸면 존재가 부재가 되고 부재가 존대가 된다라… 뭔가 철학적이군요. (멋있는 척)

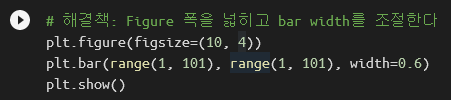
각 현상에 대해 해결책은 공통점과 차이점이 있습니다.

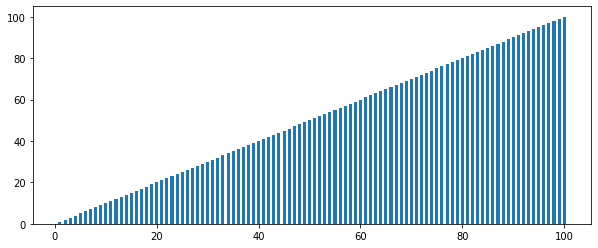
1. 막대가 너무 촘촘해서 틈이 사라지는 경우

막대 사이의 틈을 보이게 하고 싶다면 막대 폭을 줄여야 합니다. 그런데 무작정 줄이다 보면 현상2가 나타날 수 있습니다. 그래서 Figure 폭 또는 figsize와 함께 조절해야 합니다.

2. 막대가 너무 얇아서 막대가 사라지는 경우

막대 폭을 늘리면 됩니다. 마찬가지로 Figure 폭이 충분하지 않다면 현상1이 생길 수 있으므로 Figure 폭 또는 figsize 조절이 필요할 수 있습니다. edgecolor 설정도 도움이 될 수 있습니다.





==

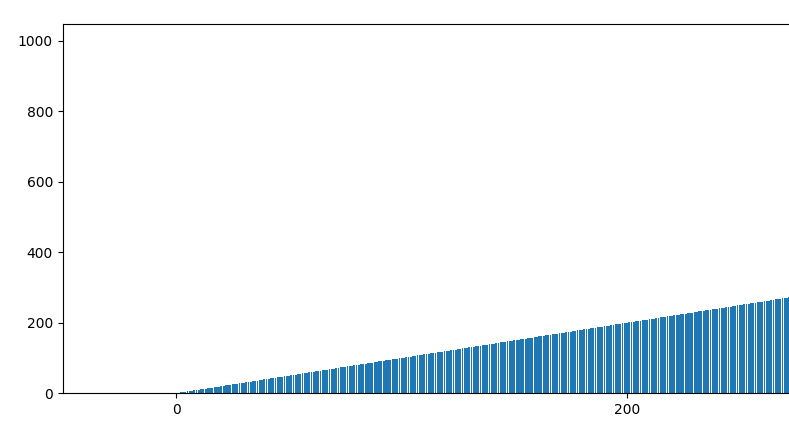
**여전히 해결되지 않는 경우**

위 방법으로도 해결되지 않는 경우가 있습니다. bar가 너무 많은 경우입니다. 이번엔 막대 1,000개를 그려보겠습니다.

# bar가 너무 많으면 Figure 폭으로도 해결되지 않을 수 있음

plt.bar(range(1000), range(1000))

plt.show()

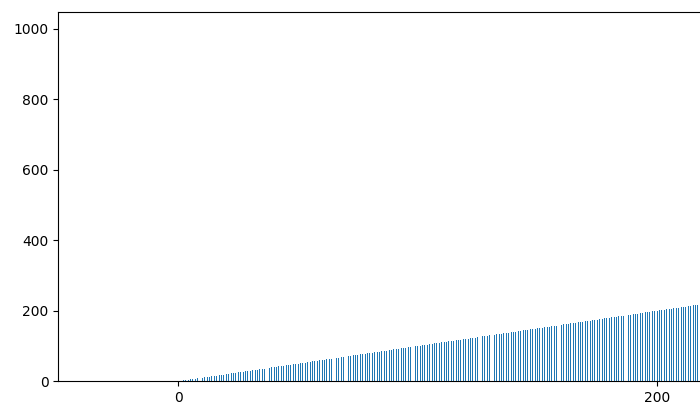


vscode로 얻은 그래프의 일부입니다. Figure 창을 모니터 2개 폭만큼 늘렸는데도 많은 막대가 붙어있습니다. 막대 폭을 0.1로 줄이면 붙은 막대는 없어지는 대신 사라지는 막대가 생깁니다. 몇 번의 시행착오 후에

# bar가 너무 많으면 bar width로도 해결되지 않을 수 있음

plt.bar(range(1000), range(1000), width=0.4)

plt.show()

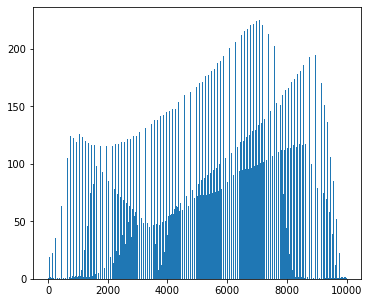


이 정도가 그나마 최선임을 알아냈습니다. 하지만 적절한 width를 찾는 데에 시간 들이느니 차라리 다른 유형의 그래프를 그리는 것이 낫지 않을까 싶습니다. 사실 이 개인 연구를 하게 된 계기도 비슷한 현상의 발견이었습니다.

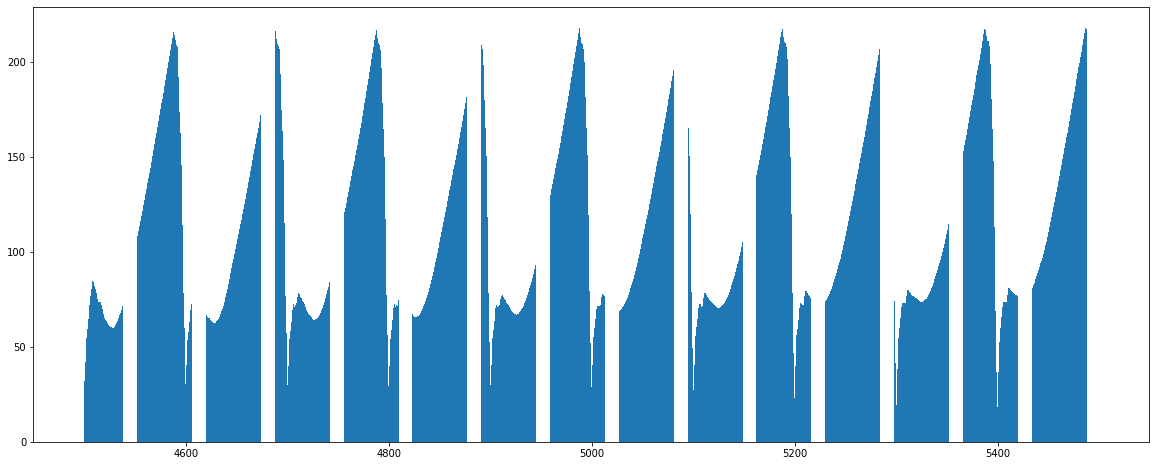
==

**왜 이 글을 쓰게 되었나?**

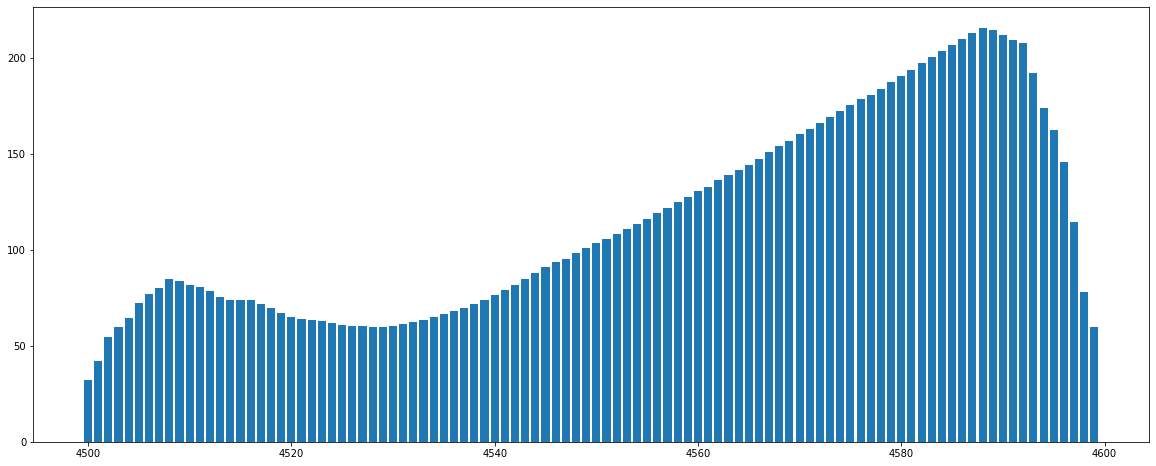
[혼공머신](https://www.hanbit.co.kr/store/books/look.php?p_code=B2002963743) 책을 공부하던 중 이상한 그래프를 발견했습니다. 그래프의 정체는 뒤에서 간단히 설명하겠습니다. 코드와 자세한 설명은 [저의 다른 글](https://lazymatlab.tistory.com/143) 링크로 대신하겠습니다.



무려 10,000개의 막대로 구성된 그래프입니다. 빈틈없이 촘촘하게 그려져야 할텐데… 생긴 게 영 이상하다 싶어 x축의 일부만 그려봤지만



더 이상해졌군요. 빈 공간에도 분명히 데이터가 있습니다. 왜냐면 x축 범위를 더 줄이면

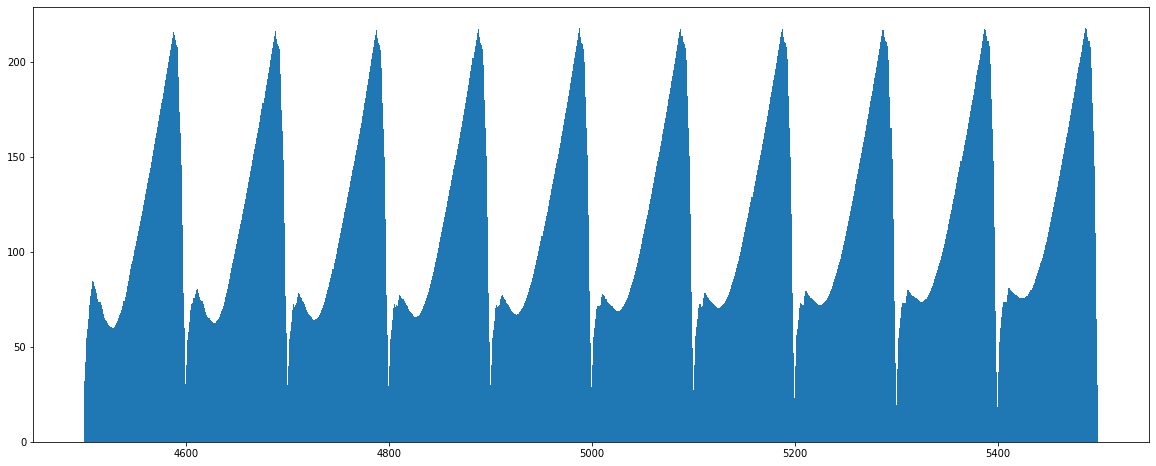


이렇게 잘 나오거든요. 안되겠다 싶어 시각화의 거장 [페가님](https://jehyunlee.github.io/)께 도움을 요청했습니다. 역시! 아래와 같이 힌트를 주셨습니다.

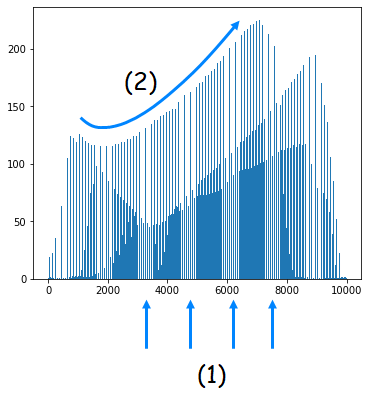
bar()로 그리는 각 막대의 width는 기본값이 0.8이다.

x축 공간이 막대 개수에 비해 충분치 않아서 일부 막대가 표시되지 않는 것 같다.

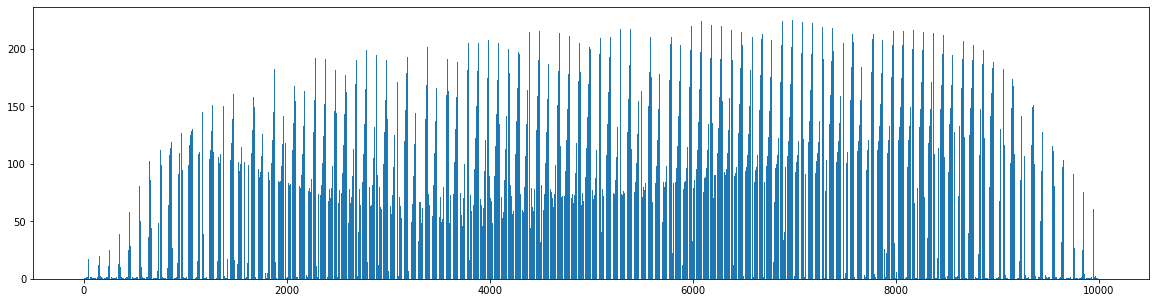
width=1로 바꾸면 어떨까?



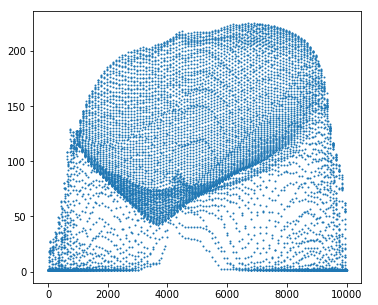
역시는 역시군요! 사랑합니다 페가님. 페가님의 힌트로부터 검색 아이디어를 얻어 비슷한 증상을 호소하는 글을 몇 개 찾았고 [[1]](https://stackoverflow.com/questions/49269927/missing-bars-in-matplotlib-bar-chart) [[2]](https://github.com/matplotlib/matplotlib/issues/8808) [[3]](https://stackoverflow.com/questions/47740478/matplotlib-bar-chart-some-bars-are-not-visible-and-seem-to-be-of-different-widt) [[4]](https://stackoverflow.com/questions/56170270/matplotlib-pyplot-bar-seems-to-produce-incorrect-results) 지금까지 보여드린 실험을 통해 본 글이 나왔습니다. 다시 맨 처음의 그래프로 되돌아가보면



저 좁은 곳에 무려 10,000개의 막대가 들어갔습니다. Figure의 폭이 기껏해야 수백 픽셀일 테니 당연히 대부분의 막대는 사라졌을 겁니다. 그 과정에서 (1)과 같이 빈 공간도 생겼구요. 게다가 (2)와 같은 이상한 패턴도 생겼습니다. 이런 패턴이 있을리가 없는 데이터였거든요. figsize를 대폭 늘렸더니



이런 그래프가 나왔습니다. 이게 제대로 된 그래프입니다. (2)의 패턴은 원래 데이터에는 존재하지 않으나, Figure의 폭이 좁아 막대가 사라지면서 나타난 훼이크 패턴인 셈이죠. 이런 경우엔 차라리 scatter plot이 낫다고 생각합니다.

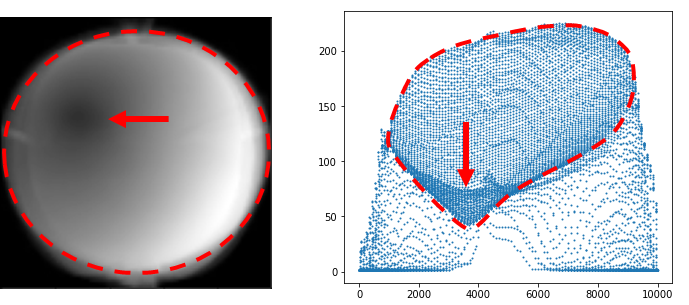


훨씬 해석하기 편합니다. 이 그래프가 무엇이냐면

실내, 어두운, 닫기이(가) 표시된 사진

자동 생성된 설명

100 x 100 크기의 이 이미지 각 행을 잘라서 한 행으로 이어붙인 (10000,) 행벡터를 그린 그래프입니다. 그래프와 이미지를 함께 보면 어디가 어디에 해당하는지 직관적으로 알 수 있습니다.



==

긴 글 읽어주셔서 감사합니다.

게으른 파이썬