

5회차 파이통! 스터디 내용 블로그 게재

우리 동네 인구 분석

```
result = []

for row in data :
    if '진잠동' in row[0] :
        for i in row[3:104] :
            result.append(int(i))
```

result = []

➡ append문 수행시 넣어줄 리스트를 만들기

for row in data:

➡ csv안에 있는 data를 row로 지정하기

if '진잠동' in row[0]

➡ 0번째 열인 row[0]에는 행정구역명이 기록되어 있습니다. 행정구역명이랑 지정한 문장이 맞으면 다음 문장 실행하기

Learning & Sharing



result에 빈 리스트를 생성해줌으로써 append문을 수행했을때 리스트의 요소를 채워넣을 수 있도록 해주었다.

행정안전부에서 데이터를 다운로드를 받을때는 2021년 3월에 해당하는 데이터만 지정해서 받았다.

for 문을 통해 csv 내부의 데이터를 지정하는 과정을 가졌는데,

row는 csv 파일의 열을 의미한다. 행정안전부에서 받아낸 공공 데이터에서 1열은 행정구역을 의미했었고 2열은 2021년 3월의 구역별 총 인구, 3열은 연령구간 인구수의 계가 있었다.

우리가 알고 싶은 데이터는 각 나이에 해당하는 인구수였기 때문에, 4열에 해당하는 리스트부터 넣어주었다.

그 코드는 밑의 이미지를 통해 확인 할 수 있는

스터디 내용 카드뉴스

Learning & Sharing

파이썬 코드 읽기

2021. 04. 13



우리동네 인구분석

```
import csv
```

```
f = open('age.csv', 'r', encoding='cp949')  
data = csv.reader(f)
```

import csv

➡ csv 파일 읽는 모듈 불러오기

f= open('age.csv', 'r', encoding='cp949')

➡ age.csv파일을 읽기모드로 열고, 한글이 들어
가도 읽을 수 있게 해주는 변환해주기

data = csv.reader(f)

➡ data값을 csv.reader(f)로 지정하기



우리동네 인구분석

```
result = []

for row in data :
    if '진잠동' in row[0] :
        for i in row[3:104] :
            result.append(int(i))
```

result = []

➡ **append문 수행시 넣어줄 리스트를 만들기**

for row in data:

➡ **csv안에 있는 data를 row로 지정하기**

if '진잠동' in row[0]

➡ **0번째 열인 row[0]에는 행정구역명이 기록되어 있습니다. 행정구역명이랑 지정한 문장이 맞으면 다음문장 실행하기**



우리동네 인구분석

```
result = []  
  
for row in data :  
    if '진잠동' in row[0] :  
        for i in row[3:104] :  
            result.append(int(i))
```

for i in row[3:104] :

➡ 나이를 지정하는 3~103열까지 슬라이딩해서
i에 지정하기

result.append(int(i))

➡ i의 데이터를 int형으로 변환 후에 result에
값을 하나씩 추가하기



우리동네 인구분석

```
import matplotlib.pyplot as plt

plt.style.use('ggplot')
plt.plot(result)
plt.show()
```

import matplotlib.pyplot as plt

➡ **그래프그리기 위한 함수 불러오기**

plt.style.use('ggplot')

➡ **ggplot을 스타일로 지정해 격자를 제공하기**

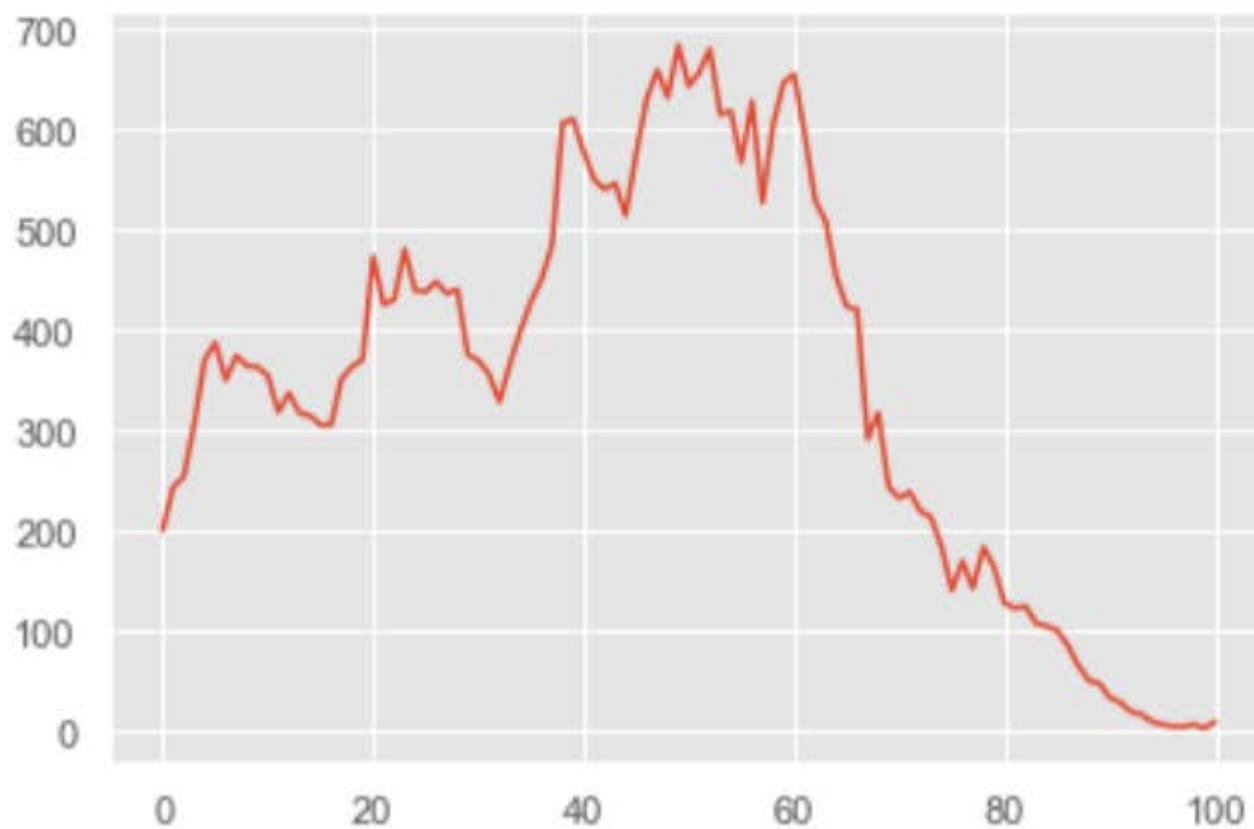
plt.plot(result)

plt.show()

➡ **result를 선형 그래프로 그리고, 보여주기**



우리 동네 인구분석



Learning & Sharing



우리동네 인구분석

나이별 바그래프로
바꿀수 있을까?



Learning & Sharing

바 그래프로 바꾸기

```
result = []

for row in data :
    if '진잠동' in row[0] :
        for i in row[3:104] :
            result.append(int(i))
```



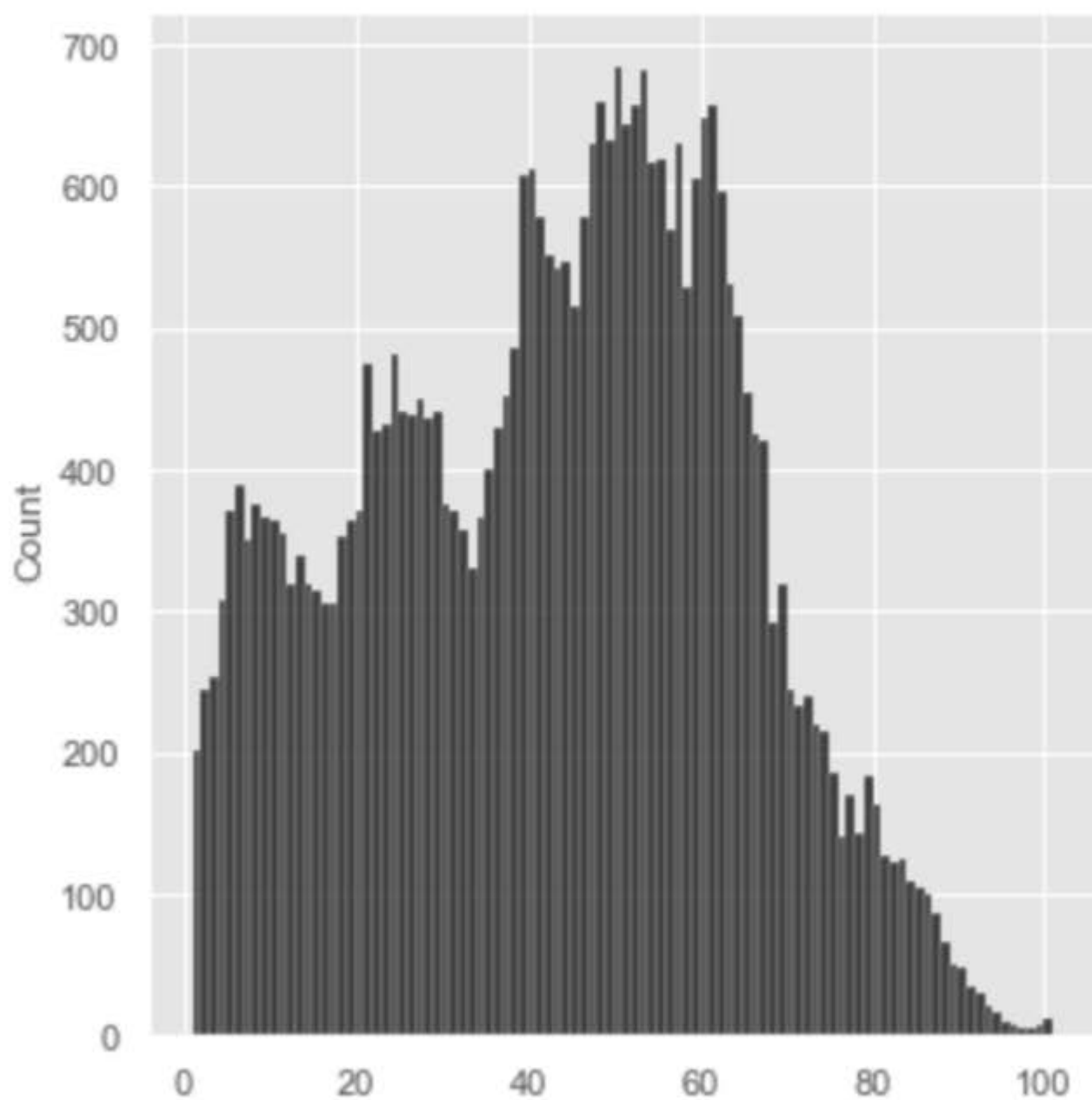
```
result = []
a=1

for row in data:
    if '진잠동' in row[0]:
        for i in row[3:104]:
            for k in range(0,int(i)):
                result.append(int(a))
            a = a+1
```



바 그래프로 바꾸기

```
sns.displot(result, bins = 100, color='black')
```



Learning & Sharing



매년 5월 19일

최고, 최저 온도

```
import csv
import matplotlib.pyplot as plt

f = open('###.csv', 'r', encoding='cp949')
data = csv.reader(f)

next(data)
high = []
low = []
year = []
```

next(data)

➡ **반복자를 입력받아 다음 출력값을 반환하기**

high=[],low=[],year=[]

➡ **데이터를 넣을 곳 미리 만들기**



매년 5월 19일 최고, 최저 온도

```
for row in data :  
    if row[-1] != '' and row[-2] != '':  
        if 1981 <= int(row[0].split('-')[0]) :  
            if row[0].split('-')[1] == '05' and row[0].split('-')[2] == '19':  
                high.append(float(row[-1]))  
                low.append(float(row[-2]))  
                year.append(int(row[0].split('-')[0]))
```

if row[-1] != '' and row[-2] != '':

➡ row[-1](최고기온), row[-2](최저기온)이
NULL 함수와 같지 아니하면 다음문장 실행

if 1981 <= int(row[0].split('-')[0]) :

➡ -로 구분했을 때, row[0](년도)가 1981년도
보다 크면 다음문장 실행

if row[0].split('-')[1] == '05' and row[0].split('-')[2] == '19':

➡ -로 구분했을 때, row[1](월)이 5이고,
row[2](일)이 19일 때 밑의 값을 저장하는 문장 실행한다.



매년 5월 19일

최고, 최저 온도

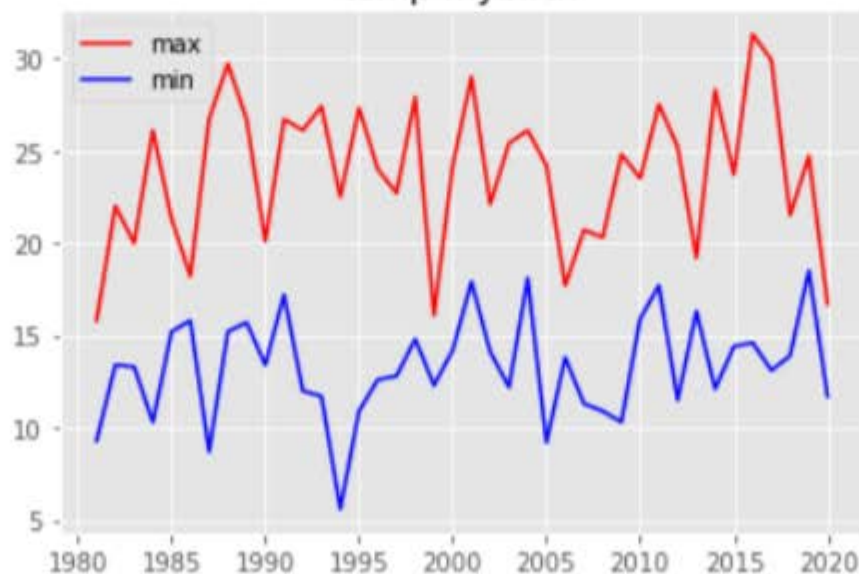
```
plt.rc('font', family='NanumBarunGothic')
plt.title('temp-mybirth')

plt.plot(year, high, 'r', label='max')
plt.plot(year, low, 'b', label='min')
plt.legend()

plt.show()
```



temp-mybirth



Learning & Sharing

