03/23/2019

파이썬 라이브러리 활용 기초

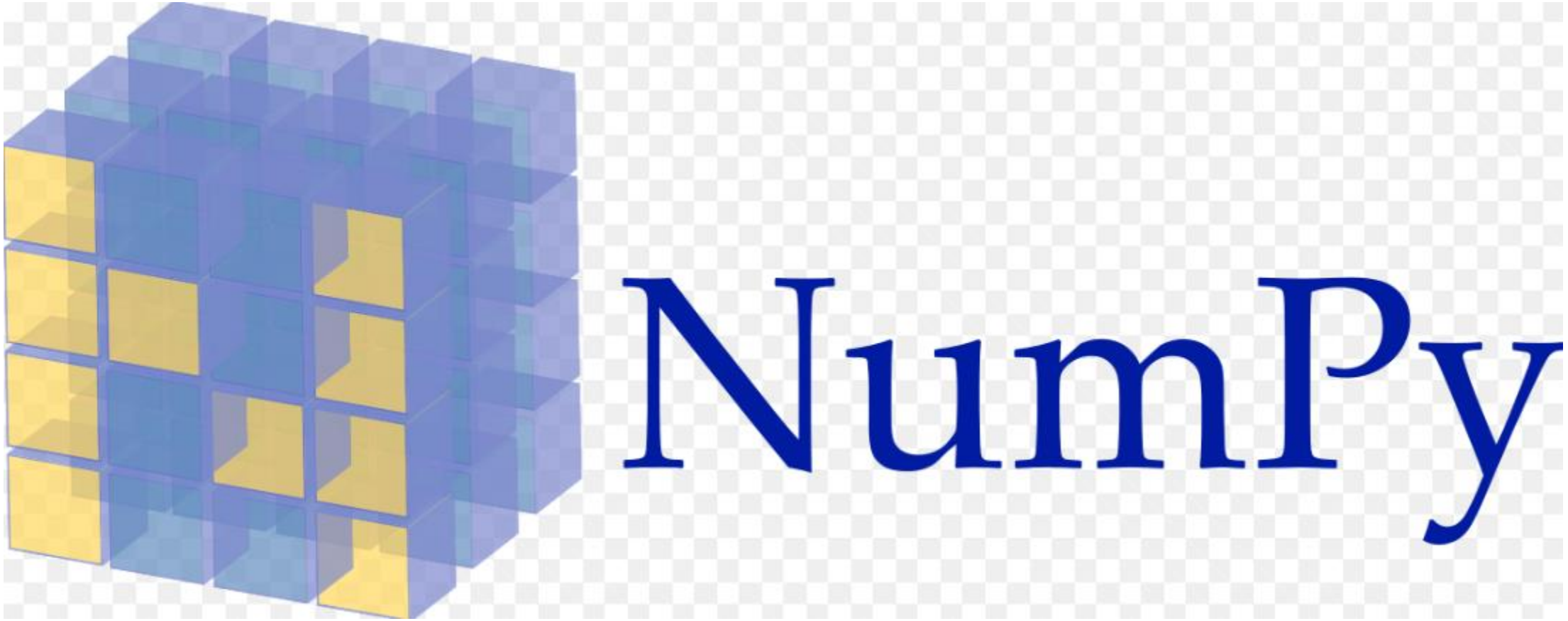
-Numpy, Pandas, Matplotlib-

4기 김신

Growth Hackers

CONTENTS

Numpy



Numpy

NumPy 소개

NumPy(보통 "넘파이"라고 발음한다)는 2005년에 Travis Oliphant가 발표한 수치해석용 Python 패키지이다. 다차원의 행렬 자료구조인 ndarray 를 지원하여 벡터와 행렬을 사용하는 선형대수 계산에 주로 사용된다. 내부적으로는 BLAS 라이브러리와 LAPACK 라이브러리에 기반하고 있어서 C로 구현된 CPython에서만 사용할 수 있으며 Jython, IronPython, PyPy 등의 Python 구현에서는 사용할 수 없다. NumPy의 행렬 연산은 C로 구현된 내부 반복문을 사용하기 때문에 Python 반복문에 비해 속도가 빠르다. 행렬 인덱싱(array indexing)을 사용한 질의(Query) 기능을 이용하여 짧고 간단한 코드로 복잡한 수식을 계산할 수 있다.

- NumPy
  - 수치해석용 Python 라이브러리
  - CPython에서만 사용 가능
  - BLAS/LAPACK 기반
  - ndarray 다차원 행렬 자료 구조 제공
  - 내부 반복문 사용으로 빠른 행렬 연산 가능
  - 행렬 인덱싱(array indexing) 기능

Numpy

내장 함수

np.array(iterable): iterable과 같은 원소를 담은 array를 반환

np.arange(): 기본 내장 함수 range()와 동일하나, array 형태로 반환

np.zeros(shape), np.ones(shape), np.full(shape, value): 각각 0, 1, value로 채워진 array를 반환

np.eye(n): n*n 형태의 Identity Matirx array를 반환

np.random.randint(n, m, size = (x, y,..)): n 이상 m 미만의 임의의 수로 채워진 array를 반환

Array의 매서드

reshape(shape): array의 형태를 변환

ndim, shape, size: array의 차원, 형태, 원소의 개수를 반환

numpy_basic.ipynb 파일 참고!

Pandas

Pandas

Pandas

Pandas("판다스"라고 읽는다) 패키지는 테이블 형태의 데이터를 다루기 위한 데이터프레임(DataFrame) 자료형을 제공한다. 자료의 탐색이나 정리에 아주 유용하여 데이터 분석에 빠질 수 없는 필수 패키지이다.

2008년도에 Wes McKinney에 의해 프로젝트가 시작되었다. 원래는 R 언어에서 제공하는 데이터프레임 자료형을 파이썬에서 제공할 수 있도록 하는 목적이었으나 다양한 기능이 추가되어 원래의 R 데이터프레임보다 능가하는 대규모 프로젝트가 되었다.

- Pandas
  - 데이터 분석 라이브러리. R의 data.frame 자료구조 구현
  - http://pandas.pydata.org/
  - 2008, Wes McKinney (AQR Capital Management)

Growth Hackers

Pandas

csv 파일 입출력

입력: pd.read_csv('파일 경로', engine = x, index_col = y)

출력: DataFrame.to_csv('파일 경로', encoding = x)

DataFrame의 주요 매서드

평균 – mean(), 분산 – var(), 표준편차 – std(), 최소값 – min(), 최대값 – max()

apply(function): DataFrame의 특정 부분에 function을 적용한 것을 반환

head(): DataFrame의 앞 5줄만을 보여줌

groupby([col_name]): 특정 column의 값을 기준으로 데이터를 묶어줌

pandas_basic.ipynb 파일 참고!

Growth Hackers

Matplotlib

Matplotlib

Matplotlib

Matplotlib("맷플롯리브"라고 읽는다) 패키지는 파이썬에서 각종 그래프나 챠트 등을 그리기 위한 시각화 기능을 제공한다. Tkinter, wxPython, Qt, GTK+ 등의 다양한 그래픽 엔진을 사용할 수 있다. 또한, MATLAB의 그래프 기능을 거의 동일하게 사용할 수 있는 pylab이라는 서브패키지를 제공하므로 MATLAB에 익숙한 사람들은 바로 Matplotlib을 사용할 수 있다.

- Matplotlib
  - 시각화 라이브러리, MATLAB 플롯 기능 구현
  - http://matplotlib.org/
  - 2002, John D. Hunter

Matplotlib

차트

Line Chart: plt.plot(x, y, ...) 색: color, 점 모양: marker, 선 모양: linestyle, 라벨: label ...

Bar Chart: plt.bar(x, y, ...) 색: color, 너비: width, 라벨: label ...

Histogram: plt.hist(x, ...) 구간: range, 구간 개수: bins, 스타일: histtype, 색: color ...

Scatter Plot: plt.scatter(x, y, ...) 색: color, 점 크기: size, 점 모양: marker, alpha: 투명도 ...

주요 함수

plt.show(): 설정된 차트를 display하는 함수
plt.figure(): 차트에 대한 각종 설정을 넣는 함수 차트 크기: figsize
plt.subplot(n, m, x): n*m등분한 칸 중 x번 째 칸에 차트를 그리겠다는 선언
plt.title(x): x를 타이틀로 넣음 / plt.xlabel(x): x를 횡축 라벨로 넣음 / plt.ylabel(x): x를 종축 라벨로 넣음
plt.grid(True): 그리드를 넣음 / plt.legend(): 범례를 넣음
plt.xtics(xs, labels): xs가 나타내는 x축 위치들에 labels의 원소들을 라벨로 넣음
plt.annotate(label, (x, y)): (x, y)가 나타내는 위치에 label을 라벨로 넣음

Exercise

In [1]:
```
import pandas as pd
```

In [6]:
```
CCTV_Seoul = pd.read_csv('C:₩₩Users₩₩shin₩₩Desktop₩₩study₩₩data₩₩CCTV_in_Seoul.csv', encoding = 'utf-8')
CCTV_Seoul.head()
```

Out[6]:

	기관명	소계	2013년도 이전	2014년	2015년	2016년
0	강남구	3238	1292	430	584	932
1	강동구	1010	379	99	155	377
2	강북구	831	369	120	138	204
3	강서구	911	388	258	184	81
4	관악구	2109	846	260	390	613

- 한글이 포함된 자료일 경우 인코딩을 해야 깨지지 않고 열리는 경우가 많다.
- head(number) - 지정한 row 만큼만 데이터를 간추려서 보여주는 것. 아무것도 넣지 않으면 첫 5행이 출력된다.

Exercise

```
In [12]: CCTV_Seoul.columns.values[0] = '지역구별'
```

```
In [13]: CCTV_Seoul.head()
```

Out[13]:

	지역구별	소계	2013년도 이전	2014년	2015년	2016년
0	강남구	3238	1292	430	584	932
1	강동구	1010	379	99	155	377
2	강북구	831	369	120	138	204
3	강서구	911	388	258	184	81
4	관악구	2109	846	260	390	613

또는, CCTV_Seoul.rename(columns={CCTV_Seoul.columns[0] : '지역구별'}, inplace = True)

Growth Hackers

Exercise

자치구	세대	인구								
		합계			한국인			등록외국인		
		계	남자	여자	계	남자	여자	계	남자	여자
합계	4,263,868	10,049,607	4,910,849	5,138,758	9,765,623	4,773,899	4,991,724	283,984	136,950	147,034
종로구	73,735	163,026	79,156	83,870	153,065	74,825	78,240	9,961	4,331	5,630
중구	61,502	135,633	66,674	68,959	125,725	61,947	63,778	9,908	4,727	5,181
용산구	108,974	245,090	119,766	125,324	228,999	110,640	118,359	16,091	9,126	6,965
성동구	137,209	316,463	155,091	161,372	308,221	151,359	156,862	8,242	3,732	4,510
광진구	162,606	371,063	179,527	191,536	355,559	172,794	182,765	15,504	6,733	8,771
동대문구	161,820	364,338	179,774	184,564	348,052	173,567	174,485	16,286	6,207	10,079
중랑구	180,511	408,147	202,448	205,699	403,209	200,419	202,790	4,938	2,029	2,909
성북구	186,601	447,687	216,495	231,192	435,868	211,904	223,964	11,819	4,591	7,228
강북구	143,395	322,915	157,522	165,393	319,164	156,071	163,093	3,751	1,451	2,300
도봉구	138,087	341,649	167,043	174,606	339,413	166,160	173,253	2,236	883	1,353
노원구	217,655	548,160	265,870	282,290	543,752	263,919	279,833	4,408	1,951	2,457
은평구	205,001	487,666	235,600	252,066	483,197	233,702	249,495	4,469	1,898	2,571
서대문구	138,549	323,080	153,816	169,264	310,313	149,569	160,744	12,767	4,247	8,520

(population_in_Seoul.excel)

Growth Hackers

Exercise

자치구	세대	인구								
		합계			한국인			등록외국인		
		계	남자	여자	계	남자	여자	계	남자	여자
합계	4,263,868	10,049,607	4,910,849	5,138,758	9,765,623	4,773,899	4,991,724	283,984	136,950	147,034
종로구	73,735	163,026	79,156	83,870	153,065	74,825	78,240	9,961	4,331	5,630
중구	61,502	135,633	66,674	68,959	125,725	61,947	63,778	9,908	4,727	5,181
용산구	108,974	245,090	119,766	125,324	228,999	110,640	118,359	16,091	9,126	6,965
성동구	137,209	316,463	155,091	161,372	308,221	151,359	156,862	8,242	3,732	4,510
광진구	162,606	371,063	179,527	191,536	355,559	172,794	182,765	15,504	6,733	8,771
동대문구	161,820	364,338	179,774	184,564	348,052	173,567	174,485	16,286	6,207	10,079
중랑구	180,511	408,147	202,448	205,699	403,209	200,419	202,790	4,938	2,029	2,909
성북구	186,601	447,687	216,495	231,192	435,868	211,904	223,964	11,819	4,591	7,228
강북구	143,395	322,915	157,522	165,393	319,164	156,071	163,093	3,751	1,451	2,300
도봉구	138,087	341,649	167,043	174,606	339,413	166,160	173,253	2,236	883	1,353
노원구	217,655	548,160	265,870	282,290	543,752	263,919	279,833	4,408	1,951	2,457
은평구	205,001	487,666	235,600	252,066	483,197	233,702	249,495	4,469	1,898	2,571
서대문구	138,549	323,080	153,816	169,264	310,313	149,569	160,744	12,767	4,247	8,520

(population_in_Seoul.excel)

Growth Hackers

Exercise

	자치구	계	계.1	계.2	65세이상고령자
0	합계	10049607	9765623	283984	1416131
1	종로구	163026	153065	9961	26742
2	중구	135633	125725	9908	22005
3	용산구	245090	228999	16091	37640
4	성동구	316463	308221	8242	42767

population_in_Seoul.xls 파일을 다음과 같이 불러와 보세요!
(Hint: 1. read_csv (x) read_excel (o) 2. 일부 칼럼만 가져올 때는 parse_col 옵션 3. header 지정)

Exercise

```
pop_Seoul = pd.read_excel('C:₩₩Users₩₩shin₩₩Desktop₩₩study₩₩data₩₩population_in_Seoul.xls', header = 2, parse_cols = 'B, D, G, J, N'
pop_Seoul.head()
```

	자치구	계	계.1	계.2	65세이상고령자
0	합계	10049607	9765623	283984	1416131
1	종로구	163026	153065	9961	26742
2	중구	135633	125725	9908	22005
3	용산구	245090	228999	16091	37640
4	성동구	316463	308221	8242	42767

- read_csv가 아닌 read_excel
- parse_cols : 불러오고자 하는 칼럼 지정
- header = 2: 세 번째 row부터 불러올 것

Growth Hackers

Exercise

Out[19]:

	구별	인구수	한국인	외국인	고령자
0	합계	10049607	9765623	283984	1416131
1	종로구	163026	153065	9961	26742
2	중구	135633	125725	9908	22005
3	용산구	245090	228999	16091	37640
4	성동구	316463	308221	8242	42767

방금 생성한 pop_Seoul 객체의 칼럼 이름을 위와 같이 만들어 보세요!

Exercise

In [19]:
```
pop_Seoul.rename(columns = {pop_Seoul.columns[0] : '구별',
                            pop_Seoul.columns[1] : '인구수',
                            pop_Seoul.columns[2] : '한국인',
                            pop_Seoul.columns[3] : '외국인',
                            pop_Seoul.columns[4] : '고령자'}, inplace = True)
pop_Seoul.head()
```

Out[19]:

	구별	인구수	한국인	외국인	고령자
0	합계	10049607	9765623	283984	1416131
1	종로구	163026	153065	9961	26742
2	중구	135633	125725	9908	22005
3	용산구	245090	228999	16091	37640
4	성동구	316463	308221	8242	42767

- 이름이 비슷한 여러 칼럼의 이름을 한 번에 바꾸고 싶을 때는 .rename을 이용해 복붙하는 것이 빠를 수도 있어요!

Growth Hackers

Exercise

Out [9] :

	구별	인구수	한국인	외국인	고령자
1	종로구	163026	153065	9961	26742
2	중구	135633	125725	9908	22005
3	용산구	245090	228999	16091	37640
4	성동구	316463	308221	8242	42767
5	광진구	371063	355559	15504	45619

맨 위의 '합계' 행을 삭제해 보세요! (drop 이용)

Growth Hackers

Exercise

In [9]:
```
pop_Seoul.drop([0], inplace = True)
pop_Seoul.head()
```

Out[9]:

	구별	인구수	한국인	외국인	고령자
1	종로구	163026	153065	9961	26742
2	중구	135633	125725	9908	22005
3	용산구	245090	228999	16091	37640
4	성동구	316463	308221	8242	42767
5	광진구	371063	355559	15504	45619

Exercise

```
In [21]: CCTV_Seoul.sort_values(by = '소계', ascending = True).head(5)
```

Out[21]:

	구별	소계	2013년도 이전	2014년	2015년	2016년
9	도봉구	825	238	159	42	386
2	강북구	831	369	120	138	204
5	광진구	878	573	78	53	174
3	강서구	911	388	258	184	81
24	중랑구	916	509	121	177	109

Exercise

```
In [22]: CCTV_Seoul.sort_values(by = '소계', ascending = False).head(5)
```

Out[22]:

	구별	소계	2013년도 이전	2014년	2015년	2016년
0	강남구	3238	1292	430	584	932
18	양천구	2482	1843	142	30	467
14	서초구	2297	1406	157	336	398
4	관악구	2109	846	260	390	613
21	은평구	2108	1138	224	278	468

Exercise

```
In [23]: CCTV_Seoul['최근증가율'] = (CCTV_Seoul['2016년'] + CCTV_Seoul['2015년'] + CCTV_Seoul['2014년']) / CCTV_Seoul['2013년도 이전'] * 100
```

```
In [24]: CCTV_Seoul.head(5)
```

Out[24]:

	구별	소계	2013년도 이전	2014년	2015년	2016년	최근증가율
0	강남구	3238	1292	430	584	932	150.619195
1	강동구	1010	379	99	155	377	166.490765
2	강북구	831	369	120	138	204	125.203252
3	강서구	911	388	258	184	81	134.793814
4	관악구	2109	846	260	390	613	149.290780

Exercise

	구별	소계	2013년도 이전	2014년	2015년	2016년	최근증가율
22	종로구	1619	464	314	211	630	248.922414
9	도봉구	825	238	159	42	386	246.638655
12	마포구	980	314	118	169	379	212.101911
8	노원구	1566	542	57	451	516	188.929889
1	강동구	1010	379	99	155	377	166.490765

‘최근증가율’ 칼럼을 기준으로, 위와 같이 내림차순 정렬해보세요!

Exercise

```
In [25]: CCTV_Seoul.sort_values(by='최근증가율', ascending = False).head(5)
```

Out[25]:

	구별	소계	2013년도 이전	2014년	2015년	2016년	최근증가율
22	종로구	1619	464	314	211	630	248.922414
9	도봉구	825	238	159	42	386	246.638655
12	마포구	980	314	118	169	379	212.101911
8	노원구	1566	542	57	451	516	188.929889
1	강동구	1010	379	99	155	377	166.490765

Exercise

	구별	소계	2013년도 이전	2014년	2015년	2016년	최근증가율	인구수	한국인	외국인	고령자	외국인비율	고령자비율
0	강남구	3238	1292	430	584	932	150.619195	547453	542364	5089	67085	0.929578	12.254020
1	강동구	1010	379	99	155	377	166.490765	431920	427573	4347	58770	1.006436	13.606686
2	강북구	831	369	120	138	204	125.203252	322915	319164	3751	58196	1.161606	18.022080
3	강서구	911	388	258	184	81	134.793814	603611	596949	6662	79660	1.103691	13.197241
4	관악구	2109	846	260	390	613	149.290780	520040	501957	18083	72249	3.477233	13.892970

CCTV_Seoul 객체와 pop_Seoul 객체를 다음과 같이 통합해 보세요! (merge 함수 이용)

Growth Hackers

Exercise

```
data_result = pd.merge(CCTV_Seoul, pop_Seoul, on = '구별')
data_result.head()
```

	구별	소계	2013년도 이전	2014년	2015년	2016년	최근증가율	인구수	한국인	외국인	고령자	외국인비율	고령자비율
0	강남구	3238	1292	430	584	932	150.619195	547453	542364	5089	67085	0.929578	12.254020
1	강동구	1010	379	99	155	377	166.490765	431920	427573	4347	58770	1.006436	13.606686
2	강북구	831	369	120	138	204	125.203252	322915	319164	3751	58196	1.161606	18.022080
3	강서구	911	388	258	184	81	134.793814	603611	596949	6662	79660	1.103691	13.197241
4	관악구	2109	846	260	390	613	149.290780	520040	501957	18083	72249	3.477233	13.892970

Exercise

```
In [42]: import numpy as np
         np.corrcoef(data_result['고령자비율'], data_result['소계'])

Out[42]: array([[ 1.        , -0.27474224],
                [-0.27474224,  1.        ]])
```

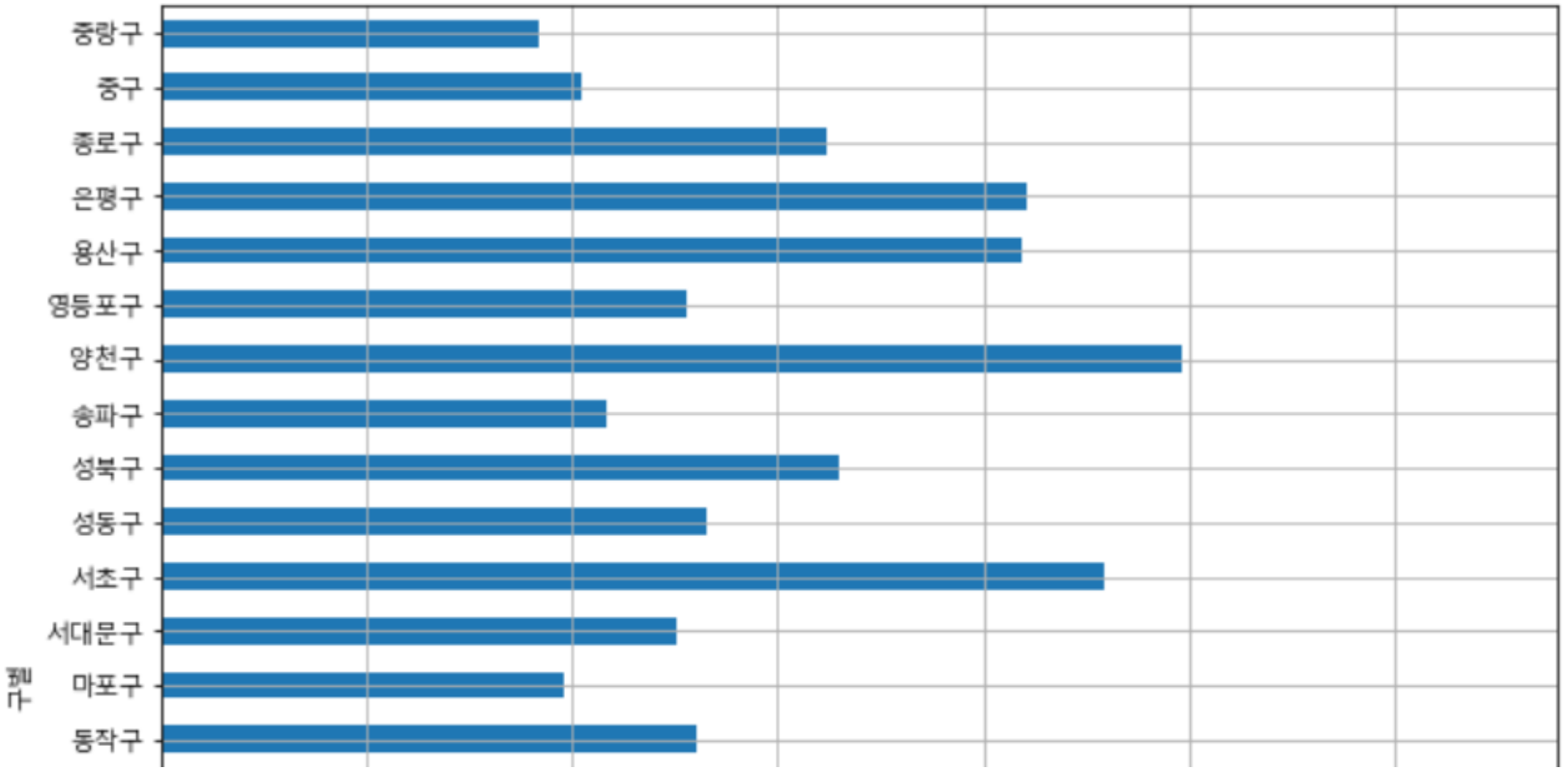
np.corrcoef를 사용하면 두 변수의 상관계수를 파악할 수 있습니다
(소계와 다른 칼럼들이 각각 어느 정도 상관관계가 있는지 파악해보면 좋겠죠?)

Exercise

```
import matplotlib.pyplot as plt #matplotlib가 한글폰트를 지원 안해서 미리 처리해주는 과정
import platform                 #아마 요즘 matplotlib는 이런거 안해도 잘 될수도 있어요!

from matplotlib import font_manager, rc
plt.rcParams['axes.unicode_minus']= False

if platform.system() == 'Darwin': #맥os 사용자
    rc('font', family = 'AppleGothic')
elif platform.system() == 'Windows':#윈도우 사용자
    path = 'c:/Windows/Fonts/malgun.ttf'
    font_name = font_manager.FontProperties(fname=path).get_name()
    rc('font', family=font_name)
```

Exercise

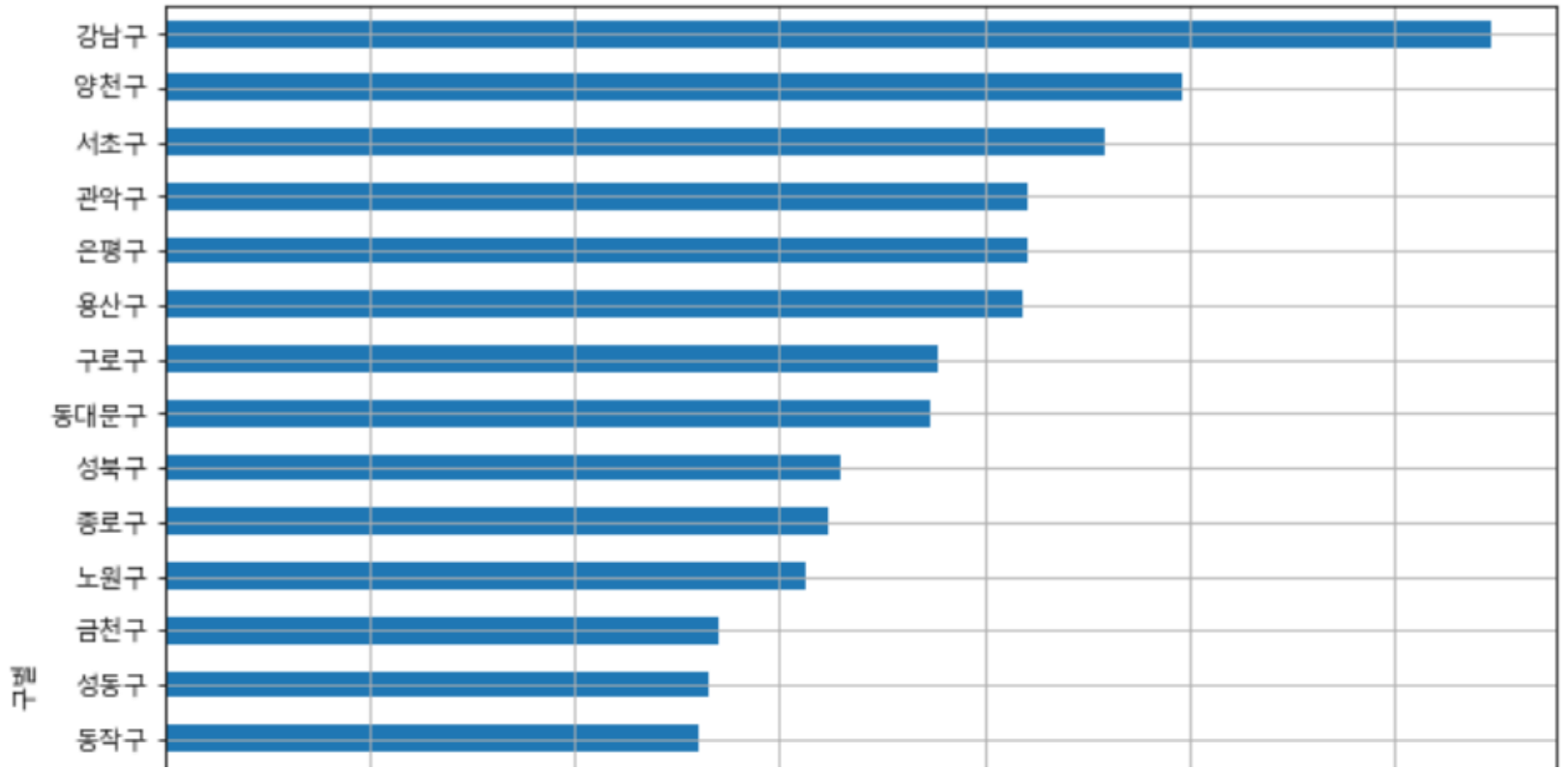
In [54]:
```
data_result['소계'].plot(kind='barh',grid=True, figsize=(10,10)) #bar는 수직바, barh는 수평바(horizontal)
plt.show()
```



Growth Hackers

Exercise

In [55]:
```
data_result['소계'].sort_values().plot(kind='barh', grid=True, figsize=(10,10))
plt.show()
```



Growth Hackers

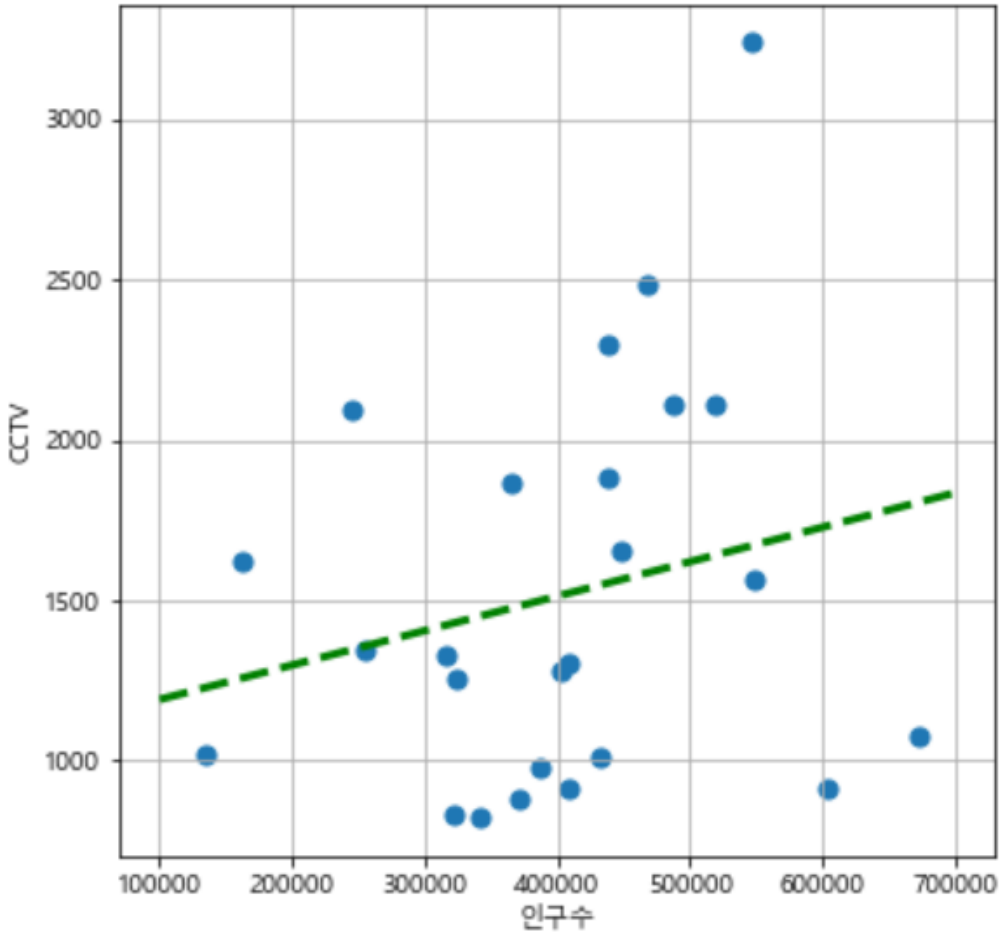
Application

```
In [27]: fp1 = np.polyfit(data_result['인구수'], data_result['소계'], 1)
         fp1

Out[27]: array([  1.07336946e-03,   1.08384235e+03])

In [28]: f1 = np.poly1d(fp1)
         fx = np.linspace(100000, 700000, 100)
```

√ poly1d를 통해 f1을 1차함수로 만들었고, 거기에 fx를 대입한 것.
√ 더 궁금하신 분들은 https://pinkwink.kr/1127, https://webnautes.tistory.com/117 를 참고하세요!

Growth Hackers

Application




```
plt.figure(figsize=(6,6))
plt.scatter(data_result['인구수'], data_result['소계'], s=50)
plt.plot(fx, f1(fx), ls='dashed', lw=3, color='g')
plt.xlabel('인구수')
plt.ylabel('CCTV')
plt.grid()
plt.show()
```

√ 먼저 인구수, 소계를 각각 X축, Y축으로 하는 산점도를 그림
√ 이후 앞서 도출한 1차식에 X값(f(x))을 대입한 직선을 그림
√ X,Y축 각각 라벨링
√ 격자 생성

Growth Hackers

Quest

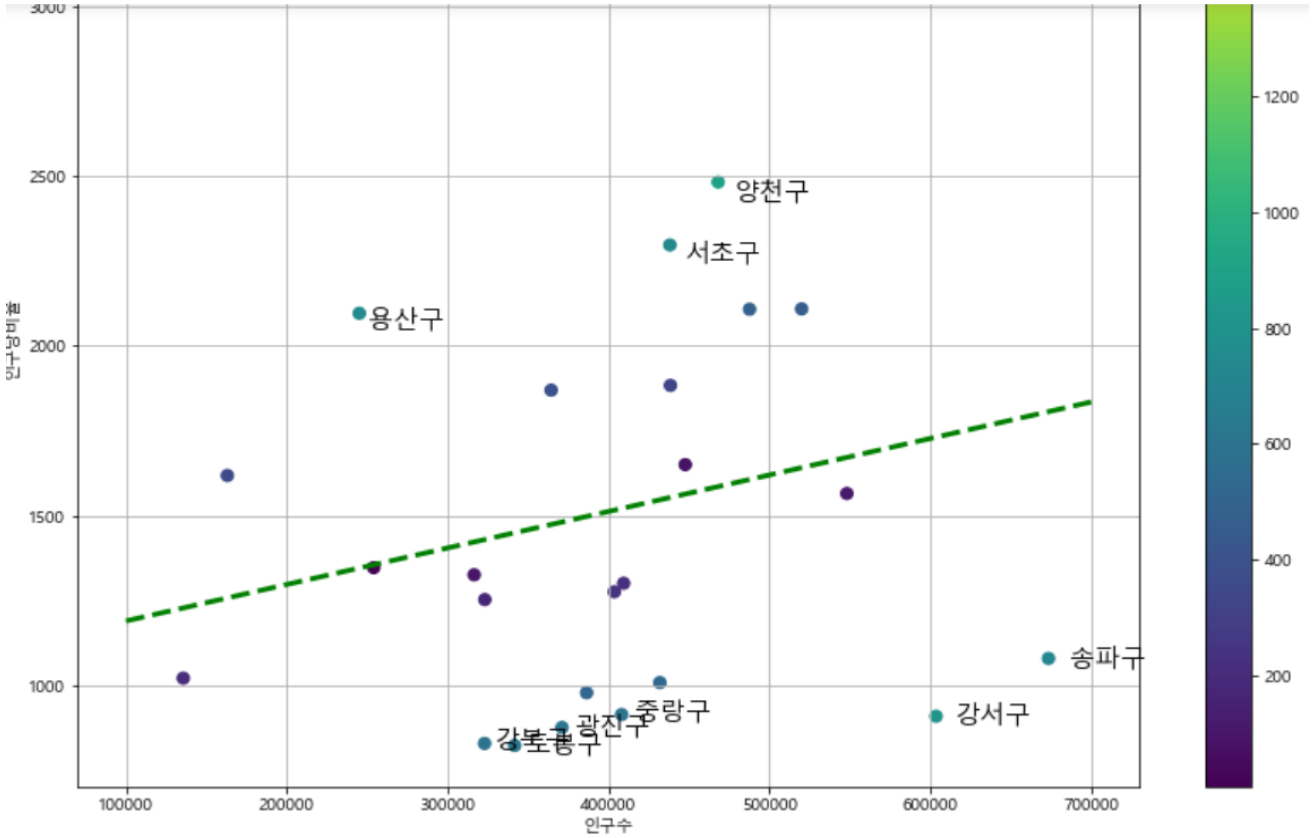
1. ‘데이터셋’ 폴더의 CCTV_in_Seoul.csv, population_in_Seoul.csv를 이용, pandas / numpy를 통해 아래와 같은 형태로 데이터를 정리해 볼 것. (‘외국인비율’ = ‘외국인’ / ‘인구수’ * 100, ‘고령자비율’ = ‘고령자’ / ‘인구수’ * 100, ‘오차’ - |소계 - 회귀직선(인구수)| , 오차는 np.abs를 활용)

구별	소계	최근증가율	인구수	한국인	외국인	고령자	외국인비율	고령자비율	CCTV비율	오차
강남구	3238	150.619195	547453	542364	5089	67085	0.929578	12.254020	0.591466	1566.538319
양천구	2482	34.671731	468145	464185	3960	58045	0.845892	12.398936	0.530178	895.665104
강서구	911	134.793814	603611	596949	6662	79660	1.103691	13.197241	0.150925	820.739963
용산구	2096	53.216374	245090	228999	16091	37640	6.565343	15.357624	0.855196	749.085528
서초구	2297	63.371266	438163	433951	4212	54751	0.961286	12.495578	0.524234	742.846867

Growth Hackers

Quest

2. (1)에서 정리한 데이터를 '오차' 컬럼을 기준으로 색깔을 입혀서 시각화 해보고, 이를 통해 무엇을 파악할 수 있는지 주석으로 간략하게 1-2줄로 서술해보기. (plt.colormap, plt.text 로 컬러맵과 이름 추가)



Growth Hackers

References

- 〈Python for Data Analysis & Visualization〉, 3기 김현세
- 〈누구나 따라 하는 금융 데이터 분석 – 기초 선형대수와 NumPy 활용하기〉, 알파스퀘어
- 〈파이썬으로 데이터 주무르기〉, 비제이퍼블릭, 민형기
- https://pinkwink.kr/ (위 도서 저자가 운영하는 블로그)
- http://aikorea.org/cs231n/python-numpy-tutorial/
- http://taewan.kim/post/numpy_cheat_sheet/
- https://webnautes.tistory.com/1177

감사합니다

Growth Hackers