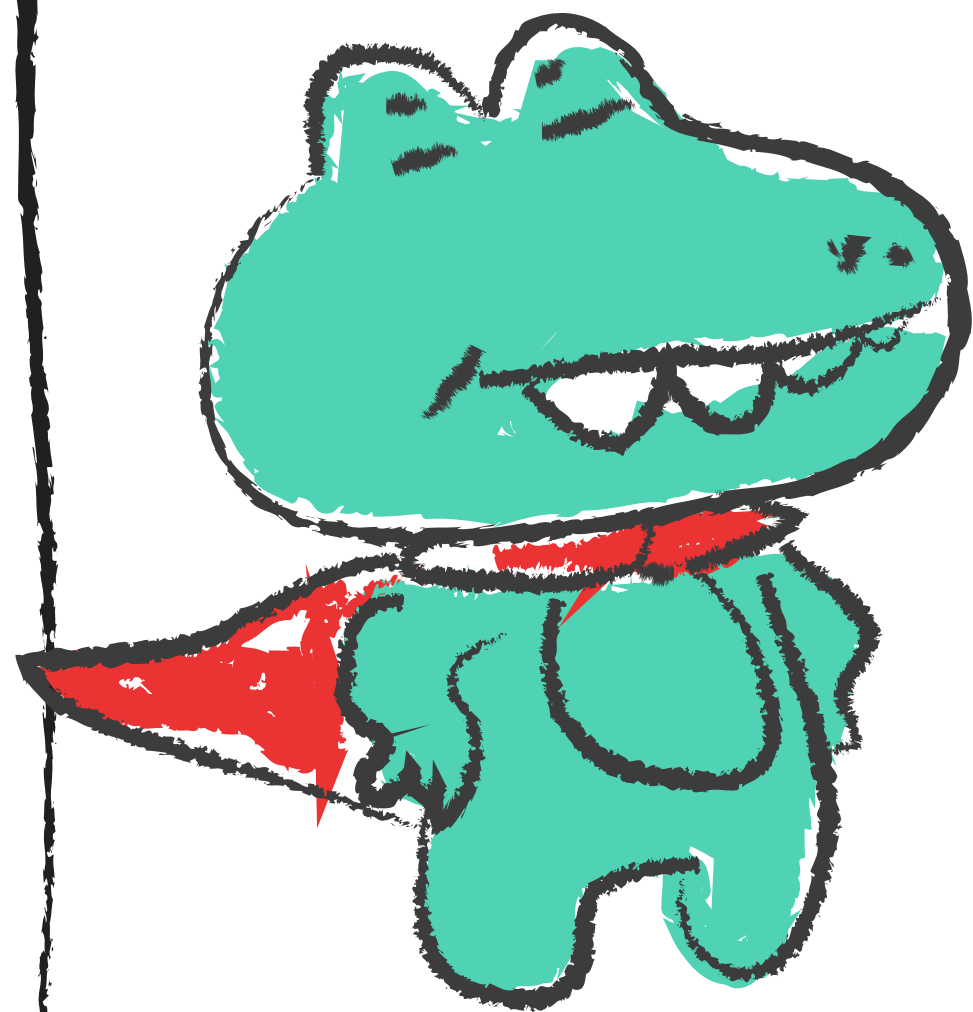


ddh&daeyou EDA

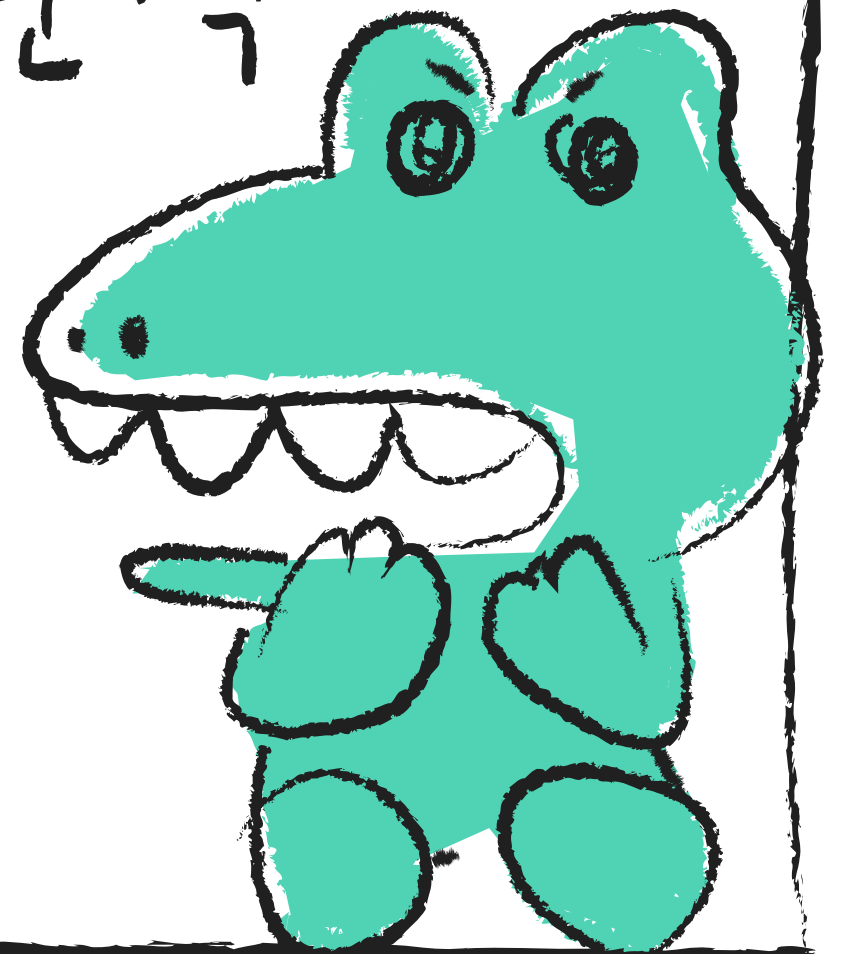
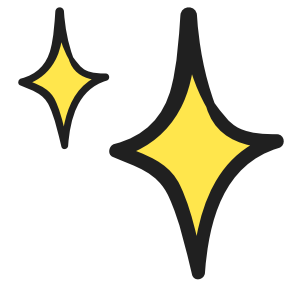
version 2



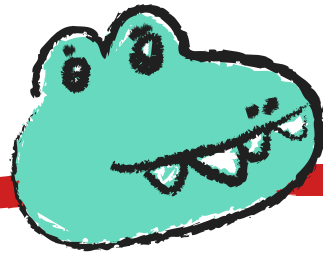
사실은 저번주 과제 다시

발표주제는요...

좌표의 밀도감과 정확도의 관계분석



핵심 키워드는

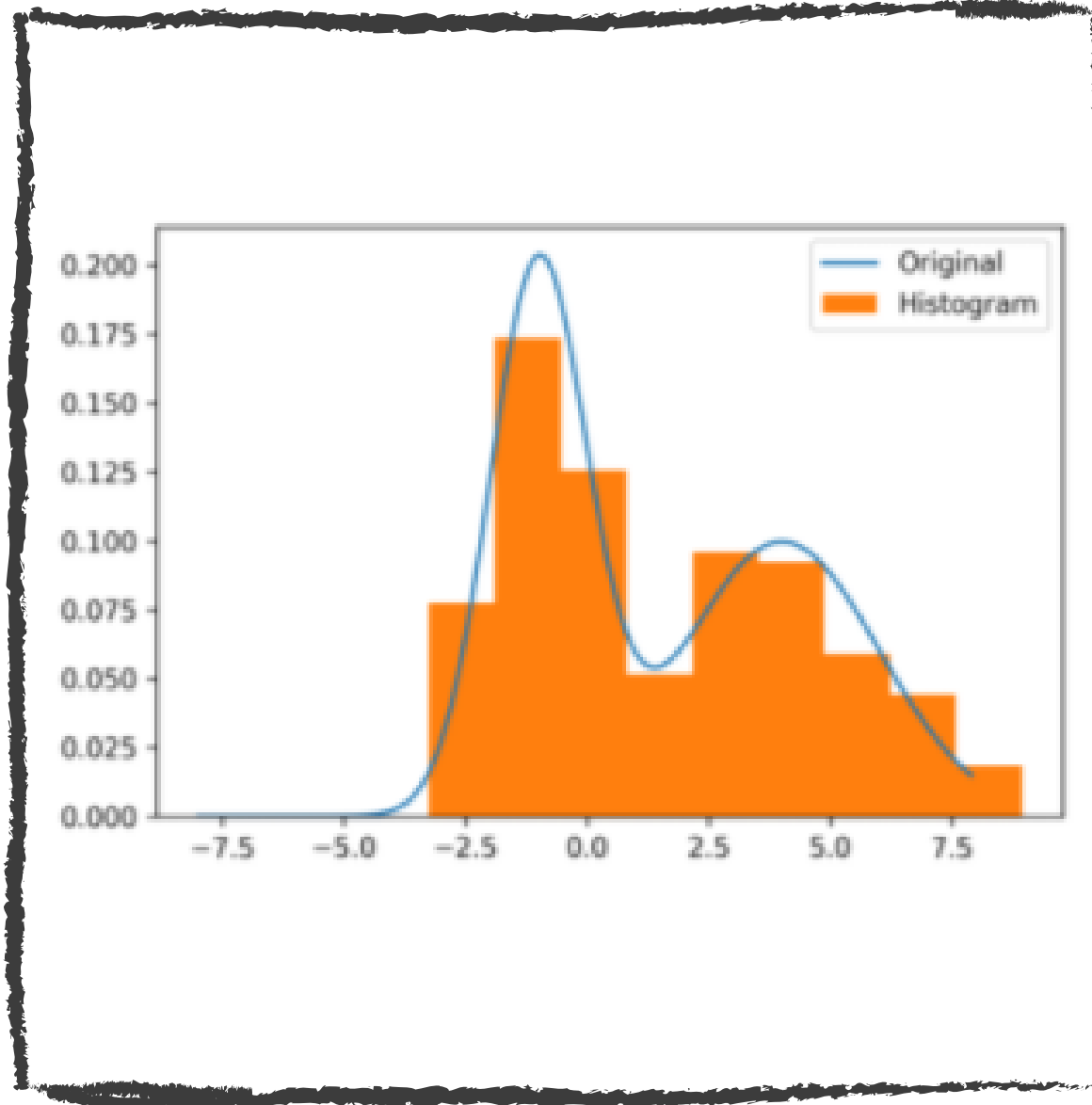


Gaussian KDE

방향을 고려한 점의 밀도를 구하는 방법

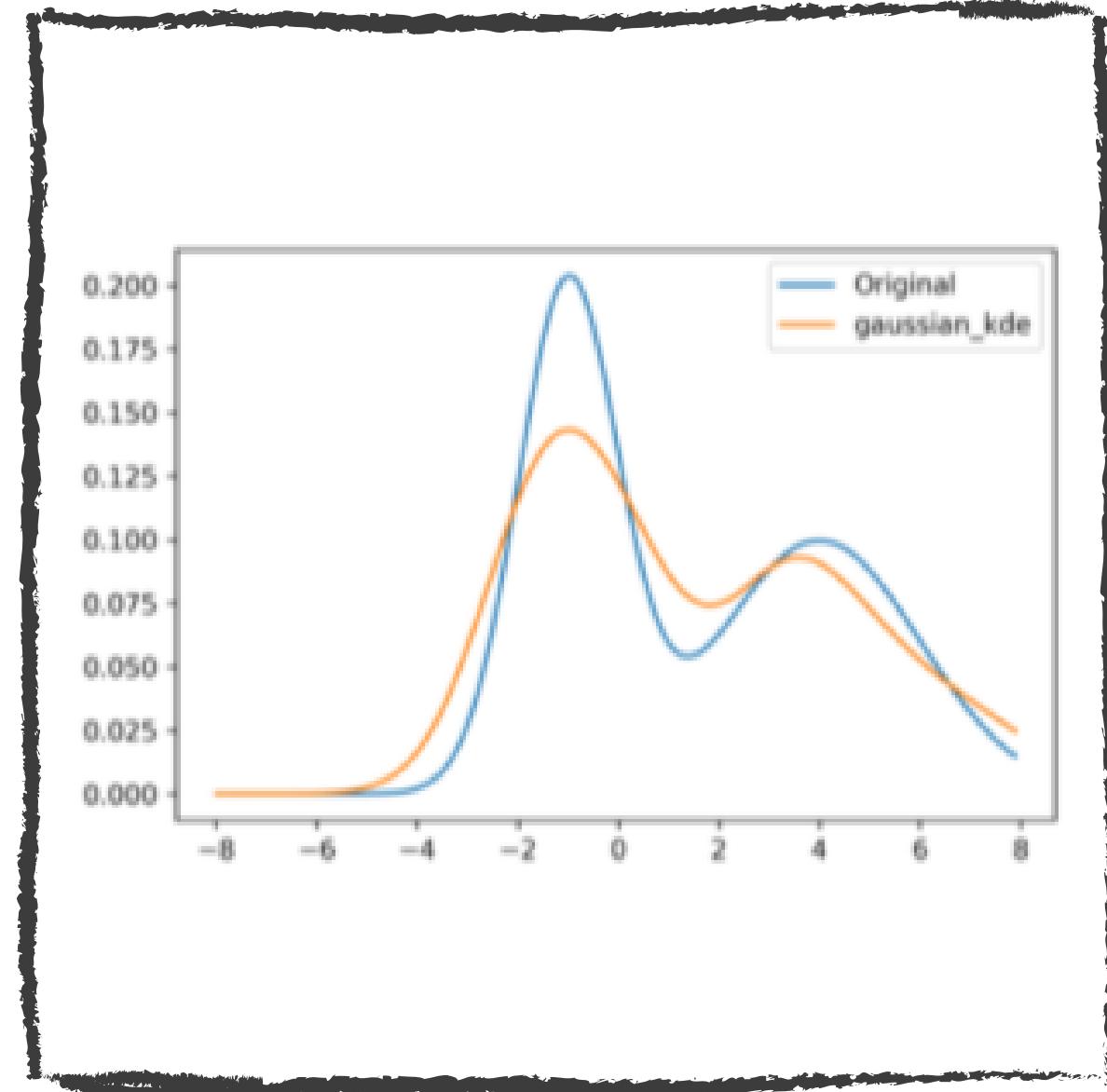
KDE

(Kernel Density Estimation)



히스토그램 사각형모양의
커널을 사용하는 비모수적 추정방법

Gaussian Kernel



부드러운(smooth)형태의 밀도함수를 추정하기위해
사각형kernel 대신 정규분포를 사용하는 방법 중 하나

Gaussian KDE로 좌표 밀도 값 구하기

```
pointer=data.guess==0
cat_ddh=data['category']=='ddh'
cat_daeyou=data['category']=='daeyou'

p0_ddh=data[pointer&cat_ddh]
p0_daeyou=data[pointer&cat_daeyou]

x=p0_ddh['x']
y=p0_ddh['y']
z=p0_ddh['z']

xyz=np.vstack([x,y,z])
density=stats.gaussian_kde(xyz)(xyz)
density_norm=np.interp(density,(density.min(),density.max()),(0,10))
p0_ddh['density']=density_norm

x=p0_daeyou['x']
y=p0_daeyou['y']
z=p0_daeyou['z']

xyz=np.vstack([x,y,z])
density=stats.gaussian_kde(xyz)(xyz)
density_norm=np.interp(density,(density.min(),density.max()),(0,10))
p0_daeyou['density']=density_norm
```

[SciPy.org](#)[Docs](#)[SciPy v1.6.0 Reference Guide](#)[Statistical functions \(scipy.stats\)](#)

scipy.stats.gaussian_kde

	x	y	z	real	guess	Accuracy	category	density
0	-24.349979	-6.683811	-25.320509	0	0	True	ddh	1.788799
1	-24.231140	-6.902044	-25.344255	0	0	True	ddh	1.738052
2	-24.265633	-6.800621	-25.410980	0	0	True	ddh	1.723510
3	-24.186052	-6.917467	-25.108315	0	0	True	ddh	1.839687
4	-24.256601	-6.849070	-25.456841	0	0	True	ddh	1.695253
...
141011	30.782660	-1.412679	-20.226223	0	15	False	daeyou	0.144306
141019	31.076153	-0.760345	-19.456327	0	15	False	daeyou	0.785488
141030	29.796114	-1.712174	-20.740591	0	15	False	daeyou	0.152463
141032	29.644148	-1.526540	-20.576386	0	15	False	daeyou	0.376461
141053	29.288887	-1.352540	-20.407276	0	15	False	daeyou	0.879533

145082 rows x 8 columns

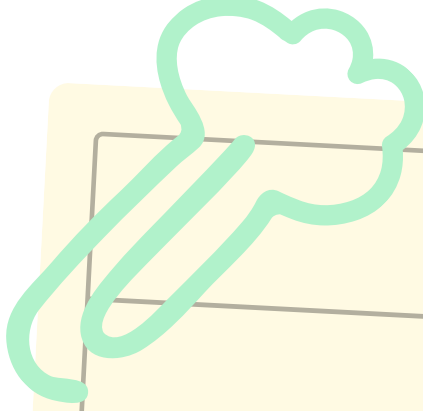
포인트별 좌표(x,y,z) 평균값

	ddh_x	ddh_y	ddh_z	daeyou_x	daeyou_y	daeyou_z
guess						
0	0.258322	-9.700276	4.158897	0.557206	-13.912546	-0.697775
1	-1.063499	-2.144917	23.130348	-5.761027	-4.110385	24.100440
2	-6.156426	-2.337704	21.702296	-11.458482	-5.146785	23.202691
3	-11.206992	-2.687634	18.572619	-15.573249	-5.427690	18.039032
4	-15.162300	-2.596730	12.817983	-18.899552	-4.813370	10.758434
5	-18.122400	-2.924231	6.575720	-21.817719	-5.198527	3.986561
6	-20.886420	-3.429490	-1.641792	-22.733033	-5.062937	-5.869676
7	-23.699441	-3.947418	-11.945116	-24.861616	-3.458533	-16.881468
9	4.181167	-2.201658	23.108704	-0.033712	-4.095855	24.625949
10	9.433486	-2.414758	21.231965	5.822317	-4.896934	24.296267
11	14.340098	-2.581406	17.586675	11.074912	-4.898581	20.442153
12	17.926875	-2.648007	11.653090	15.736959	-4.830126	13.937447
13	20.371443	-2.948560	4.971292	20.229566	-4.954480	7.744589
14	22.114279	-3.650245	-3.763092	24.057618	-5.171826	-1.461284
15	24.912700	-3.859588	-13.370466	27.101229	-3.417101	-12.573885


포인트별 정확도(%) / 밀도 평균값

	ddh_acp	daeyou_acp	ddh_dm	daeyou_dm
guess				
0	0.954334	0.981233	3.639621	5.525581
1	0.927057	0.949495	5.582180	6.130941
2	0.977438	0.983901	6.216757	6.817378
3	0.971877	0.936351	5.745079	6.898880
4	0.973918	0.930000	5.351281	5.027732
5	0.979429	0.966109	5.042169	4.950138
6	0.965644	0.971163	4.728636	5.396629
7	0.984293	0.969407	4.625976	5.092884
9	0.938475	0.925148	5.988536	6.441377
10	0.955150	0.921749	5.502643	6.515352
11	0.972438	0.915381	5.530108	5.049440
12	0.976433	0.949323	5.033219	6.511610
13	0.968811	0.948741	5.192313	5.480366
14	0.975933	0.979879	4.816980	5.264335
15	0.982945	0.883512	4.308470	3.354343

시각화 및 분석 ipynb



<https://colab.research.google.com/drive/1K1DU1RUXmJ-uzSYkzDzZ5df9X2Wy43CM?usp=sharing>



마지막으로

발표를 들어주셔서

감사합니다

