

**** Python, Netværksanalyse, gis og QGIS

Indhold

**** Python, Netværksanalyse, gis og QGIS	1
Netværksanalyse , hurtigt effektivt med Python værktøjer; tabeltal kan fremvises på kort med QGIS.....	2
Kritiske netværks-punkter kan findes.....	2
Konkrete netværks-analyse for Albertslund-området	4
Vejetyper i Albertslund(primary=motorveje mangler).....	7
Albertslund -skoleafstand(i vejnetværket).....	9
Bygninger i Albertslund	11
Fordelingen af restauranter i Albertslund	12
****Travel-Time til bestemt netværks-destination	15
Traveltime til City2 fra steder på Sjælland, ca kl 17, onsdag, altså tæt ved myldretiden(bort fra København)	17

Her viser jeg skærmdump af dele af filindholdet i mappe med gis under Python. Jeg har senest arbejdet en del med netværksanalyse og andre gis-opgaver med [OpenStreetMap](#), OSM, open source data.

Navn	Ændringsdato	Type
 OpenStreetMap and Urban Data	30-10-2024 19:33	Microsoft Word-d...
 osmx-network	29-10-2024 09:09	Microsoft Word-d...
 network-analysis, finding distances to schools, restaurants in a city	28-10-2024 19:50	Microsoft Word-d...
 Analysis of schools, restaurants, buildings with pandana network	27-10-2024 19:43	Microsoft Word-d...
 Travel time – genvej	27-10-2024 10:55	Genvej
 turbo-osm-request	08-08-2024 07:32	Microsoft Word-d...
 Automating GIS Processes 2023-course	08-08-2024 07:32	Microsoft Word-d...
 network-analysis, finding distances to schools, restaurants in a city	02-07-2024 19:13	HTML-dokument
 Pandana-network-analysis demo -used in my own analysis	02-07-2024 19:10	HTML-dokument
 Pandana-network-analysis demo, used in my own analysis-2	02-07-2024 18:39	Microsoft Word-d...
 Pandana-network-analysis demo -used in my own analysis	02-07-2024 18:18	Microsoft Edge M...
 Pandana-network-analysis demo -used in my own analysis	02-07-2024 18:16	Microsoft Word-d...
 Dataisochrone	01-07-2024 22:08	Tekstdokument
 Travel time -med-python– genvej	01-07-2024 16:30	Genvej
 Traveltime-account (Genoprettet automatisk)	01-07-2024 16:22	Microsoft Word-d...
 Pandana-network-analysis demo	28-06-2024 19:22	Microsoft Word-d...
 Pandana-network-analysis demo, used in my own analysis	28-06-2024 18:42	Microsoft Word-d...
 Traveltime-account	15-06-2024 13:36	Microsoft Word-d...
 Introduction to Python GIS	11-06-2024 11:56	Microsoft Word-d...
 Travel-time-matrix, kræver ArcgisPro	11-06-2024 11:14	Microsoft Word-d...
 Use of GIS in smart city projects	11-06-2024 11:08	Microsoft Word-d...
 UrbanAccess, problemer	03-06-2024 16:49	Microsoft Word-d...
 Network analysis in Python	28-05-2024 17:26	Microsoft Word-d...
 Pandana network	09-05-2024 20:13	Microsoft Word-d...

Dataanalyser laves hurtigt og effektivt i Python, som også muliggør gis-plot, men man kan også nemt eksportere til geopackage, så QGIS, ArcGIS også kan bruges. Det er nemmere at arbejde med flere lag samtidigt i QGIS.

Netværksanalyse , hurtigt effektivt med Python værktøjer; tabeltal kan fremvises på kort med QGIS

Jeg har arbejdet med stærke netværks analyse værktøjer i Python med open source pakken osmx, som tilgår [OpenStreetMap](#), OSM, open source data; udgangspunktet er offentlige data, suppleret med borger-indtastede oplysninger.

Netværksanalyse er et meget generelt og stærkt værktøj: Et netværk består af knudepunkter(nodes) forbundne med edges(kanter i netværket, her vejstykker). Edges har info om hastighed, kan have info om maksimal kapacitet mv.

Edges 'mødes' i knudepunkter. Der kan nemt laves værdifuld statistik over antal edges per knudepunkt, det kan måske bruges til at finde flaskehalse.

Kritiske netværks-punkter kan findes

Man kan nemt finde knudepunkter, som har særligt mange edges tilknyttet, disse knudepunkter er kritiske i netværket:

Nemmest direkte at bruge street_count-feltet i nodes

```
t1=time.time()
nodes_sorted=nodes.sort_values(by='street_count',ascending=False ) #virker
t2=time.time()
print('dt[s]:',round(t2-t1,3)) #0.004, imponerende!
nodes_sorted.head()
```

	y	x	street_count	highway	ref	\
osmid						
251082224	55.656913	12.363864	8	traffic_signals	NaN	
290005874	55.688721	12.341908	7	NaN	NaN	
277859993	55.690190	12.340706	6	NaN	NaN	
420803925	55.655116	12.354814	6	NaN	NaN	
1712515294	55.674551	12.378951	5	NaN	NaN	

Så kan man lave kort med roads og lag med måske de første 100 af disse nodes_sorted på kort

Laget med nodes og edges er gemt i GeoPackage og indlæst i QGIS, her kun disse kritiske punkter fremvises



Der kan lynhurtigt laves tabel med statistikker over hastigheder, kapaciteter, knudepunkter ud fra aktuelle data. Det kan nemt foregå næsten realtime. Det har jeg ikke afprøvet, men man må kunne sætter alarmer op, der advarer og viser kritiske forhold i netværket.

Med gis kan man få overblik over vigtige tal fra statistik rundt omkring i netværket på et kort, hvilket igen kan hjælpe med til at finde flaskehalse både for knudepunkter og edges.

Konkrete netværks-analyse for Albertslund-området

Konkret har jeg arbejdet med geografisk data for Albertslund, forstad til Kbh, med ca 25000 indbyggere. Beregninger er hurtige, de tager fra få sek til 30 sek.

Man kan lynhurtigt finde antal km vejstykker per vejtype, farvelægge hastigheder for vejstykker i området, se figur nedenfor.

Jeg har fundet områder med afstande større end 1km fra fx nærmeste ressource(skole, restaurant). På den måde kan planlæggere finde områder med 'problematiske' tilgængelighed.

Det kan selvfølgelig generaliseres til andre typer netværksdata end vejtrafik.

Kender man realistiske hastigheder i edges, kan man også beregne og vise køretider.

Jeg har bl.a. farvelagt **bilkørsel-tider til fx City2** fra celler på Sjælland, så fx City2 planlæggere kan vurdere kundepotentialet rundt omkring på Sjælland. Kender man demografisk forhold i cellerne, kan kundepotentialet indenfor forskellige tidsintervaller også beregnes og vises på et kort vha. gis-værktøjer.

Bygninger findes også i OSM databasen, med information om arealstørrelser, bygningstyper(bolig/ikke-bolig i hvert fald); bygningers placering i netværket kan nemt beregnes. Det gør at man faktisk kan finde antal bygninger/boliger indenfor fx 1km fra nærmest ressource, og hvis man kender demografien i geografisk celler kan man også sætte tal på antal borgere indenfor afstanden.

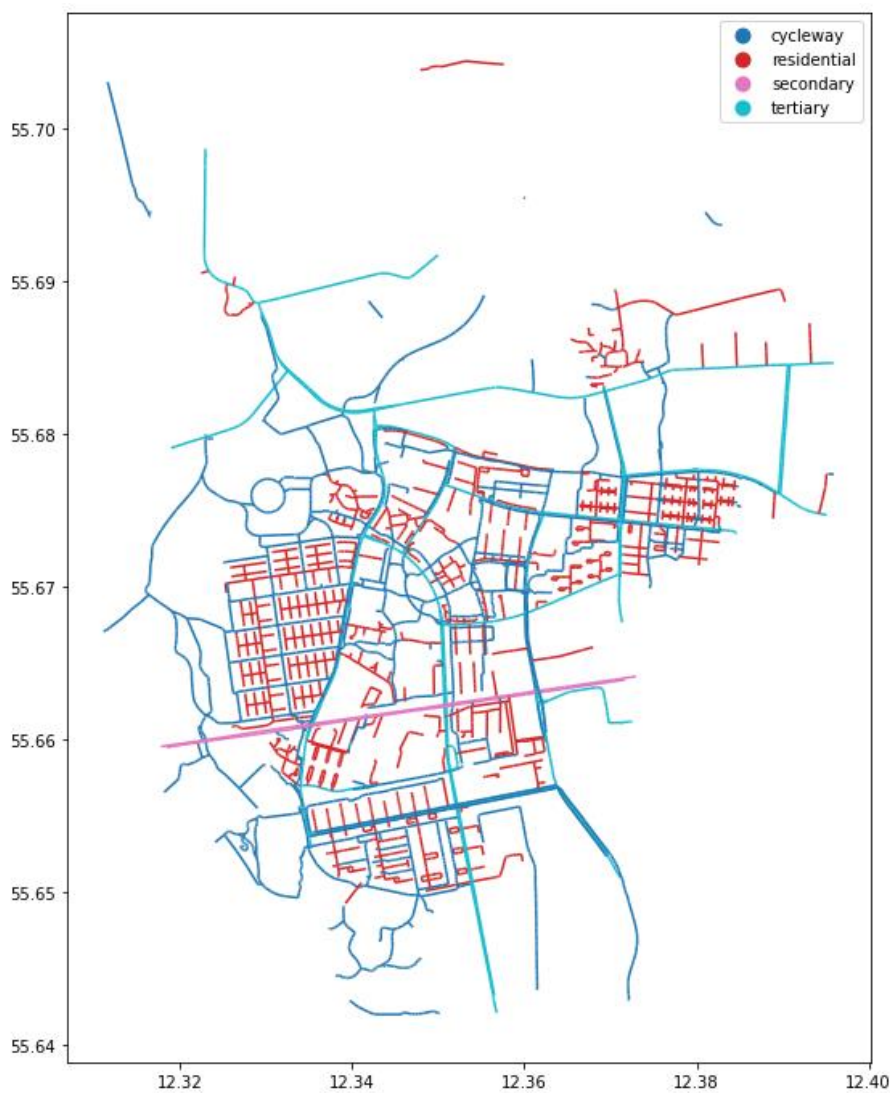


albertslund med osmnx.stats.basic_stat() funktionen, uden simplificeringer

```
{'n': 7279, antal nodes
'm': 20901, antal edges
'k_avg': 5.742821816183541,
'edge_length_total': 1003807.0600000000 m , ca 2*street, fordi edge er dobbelt-rettet
'edge_length_avg': 48.02674800248793 m,
'streets_per_node_avg': 2.965929385904657,
'streets_per_node_counts': {
0: 0, #igen fejl med løsrevne edge!
1: 796,
2: 6,
3: 5165, fleste noder må have 3 edges tilknyttet
4: 1281,
5: 27,
6: 2,
```

7: 1,
 8: 1},
 'intersection_count': 6483,
 'street_length_total': 528442.0549999983, 528km
 'street_segment_count': 10744, passer nogenlunde med at edges antal er ca det dobbelte
 'street_length_avg': 49.18485247580029,
 'circuitry_avg': 1.080010812221381,
 'self_loop_proportion': 0.0013030528667163069,
 'node_density_km': 176.84421912193943, per km²
 'intersection_density_km': 157.5052991575125,
 'edge_density_km': 24387.618584254684
 'street_density_km': 12838.56608980185}, 12.8 km vej per km²

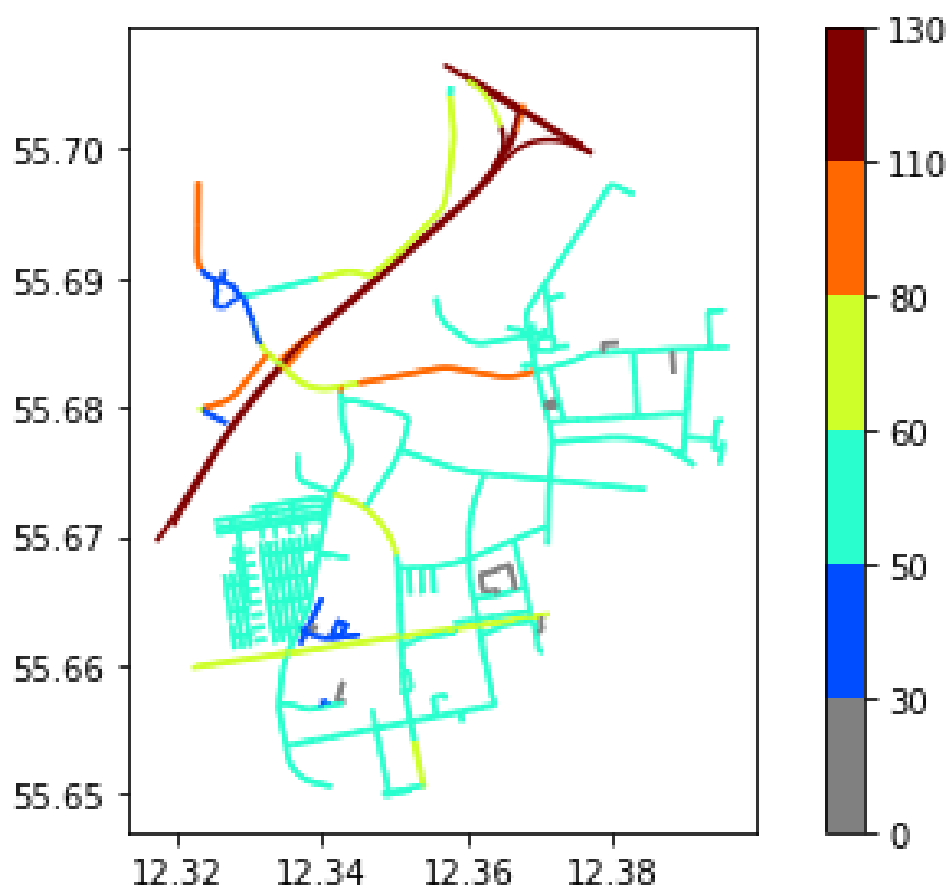
Vejetyper i Albertslund(primary=motorveje mangler)



residential er boligområder,

secondary er Roskildevejen

Vejhastigheder, angivet maxSpeed for vejene i albertslund



meget simplificeret

netværk, fordi kun ca 2000 ud af 20700 har maxSpeed-værdi angivet

Vejhastigheder i Albertslund: her er mørkerøde motorvejsstykker med. Området med tætliggende veje omkring (12.33-12.34, 55.66-55.67) er et parcelhusområde, øjensynligt er hastigheden her 50km/t, mon ikke den er 30km/t i virkeligheden; OSM-databasen er ikke 100% pålidelig. Den er jo vedligeholdt af normale borgere og derfor findes uden tvivl fejl i denne.

Laver speed-felt for de manglende maxSpeed-værdier

```
#gennemsnit af maxspeed for hver vejtype X findes, vejtype X med maxspeed=null tildeles
gennemsnitsværdien, værdier med maxspeed får vist speed=maxspeed
ox.add_edge_speeds(graph, hwy_speeds=None, fallback=None, precision=None, agg=numpy.mean)
```

residential , count: 3328 , speed: 49 uden tvivl findes maxSpeed både 30 og 50, langt flest må have 50, 49 er jo middelværdien

service , count: 2694 , speed: 33

tertiary , count: 646 , speed: 51

unclassified , count: 494 , speed: 48

secondary , count: 83 , speed: 60

motorway_link , count: 20 , speed: 81

motorway , count: 11 , speed: 110, alle de mørkerøde på figuren ovenfor

tertiary_link , count: 8 , speed: 65

secondary_link , count: 7 , speed: 61

Total count for roads with speed value: 7291 ,

fra ca 2000 har vi nu fundet værdier for alle(?) bilveje; footway 7000, cycleway 2800 styks har ingen speed oplysninger

Det er klart at køretidsanalyse med osmx-netværk vil kæve, ikke maxSpeed, men reele typiske køretider(afhængig af tid på dagen).

I min analyse om over køretider til City2 bruger jeg 3.parts platform TravelTime.

Residential-vej-analyse

```
residential=edges[ edges['highway']=='residential' ]
```

```
residential.describe()
```

```
""" max_speed er ikke numerisk, kommer derfor ikke ud med describe()
```

```
      length      speed
count  3328.000000  581.000000 , så kun 581 har haft value-værdi
mean    33.610707   49.080895 , NaN indgår ikke i beregninger af
speed-værdier
std      37.650930    4.667052
min       0.242000   15.000000
25%      12.690250   50.000000
50%      23.687500   50.000000
75%      40.781750   50.000000
max      542.486000   59.000000 """
```

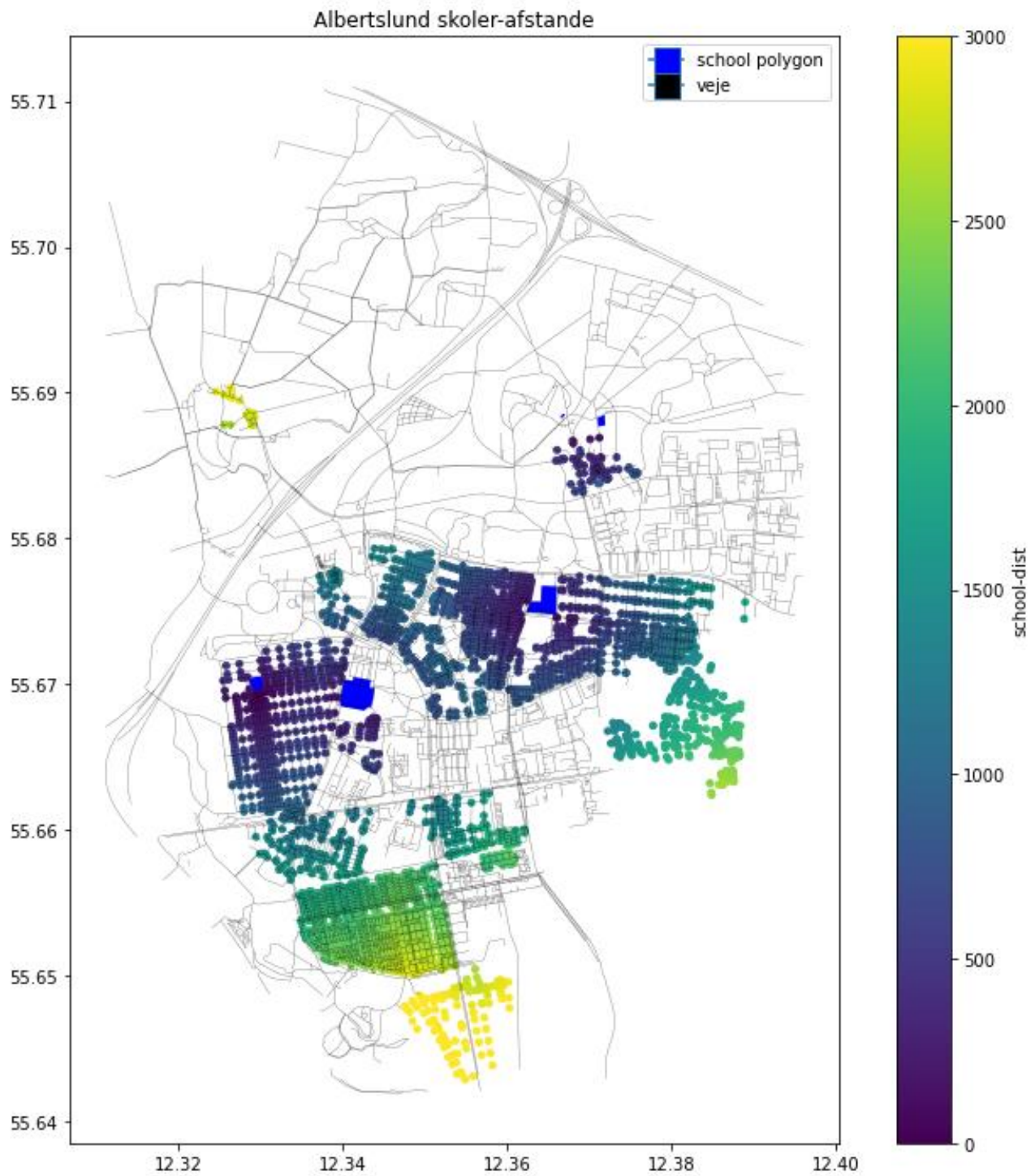
```
residential_gr=residential.groupby('maxspeed')
```

```
residential_gr.describe()
```

```
"""      speed
      max count mean std min 25% 50% 75% max
maxspeed
15    183.145  4.0 15.0 0.0 15.0 15.0 15.0 15.0
30    240.371 18.0 30.0 0.0 30.0 30.0 30.0 30.0
40    244.362  7.0 40.0 0.0 40.0 40.0 40.0 40.0
50    201.894 548.0 50.0 0.0 50.0 50.0 50.0 50.0
59     31.531  4.0 59.0 0.0 59.0 59.0 59.0 59.0
```

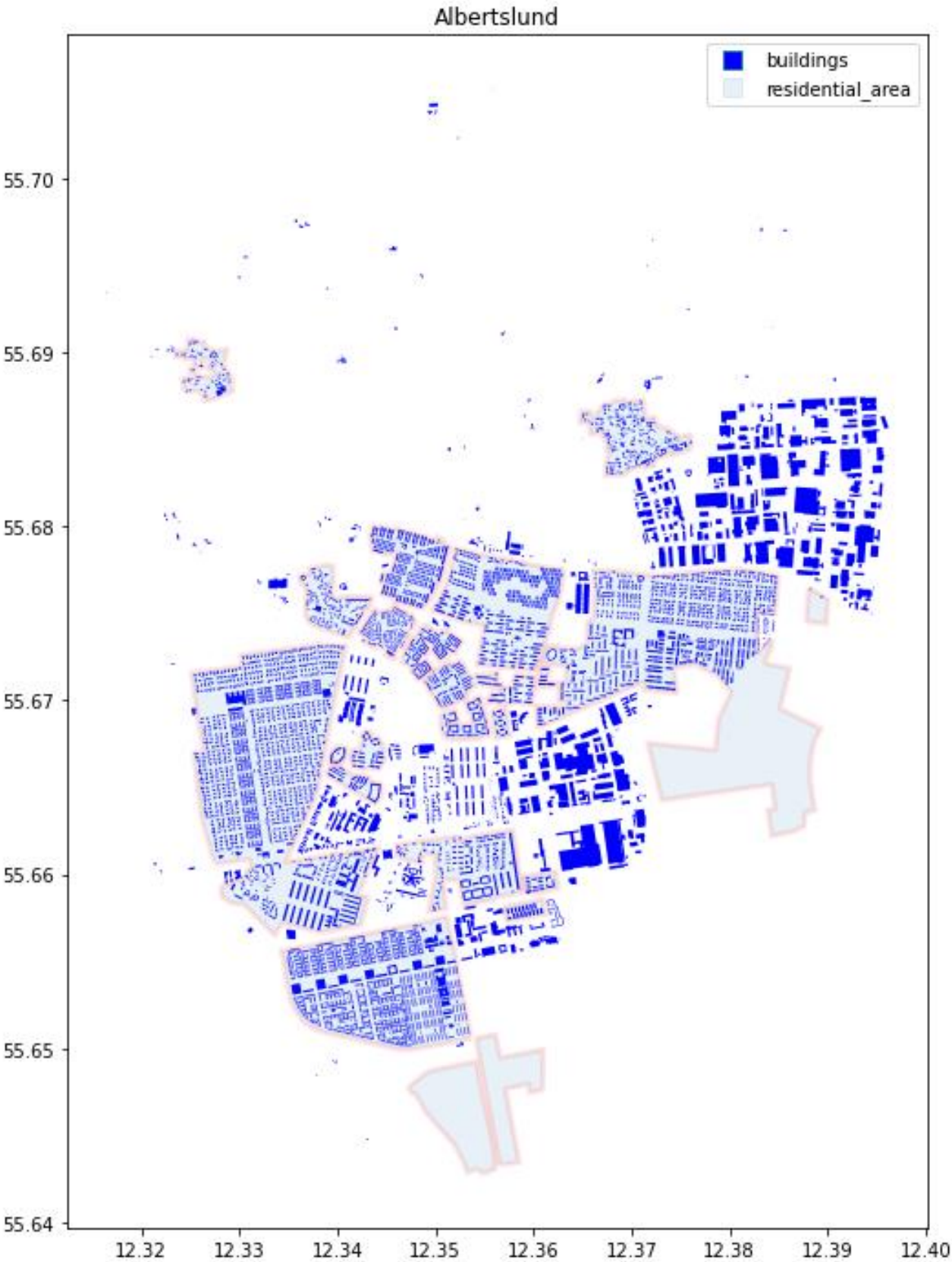
```
ser ok ud, men hvad er første max ? (4*15+18*30+7*40+548*50+4*59)/581=49.08 """
```

[Albertslund -skoleafstand\(i vejnetværket\)](#)



Der er ingen boliger i gullige noder nederst, det er et grønt rekreativt område.

Bygninger i Albertslund



Det er tydeligt, at de helt store bygninger er i kommercielt centrum, omkring (12.365, 55.665), og i industriområdet omkring (12.39, 55.68). Nederst område, markeret som residential_area, er et grønt område uden boliger, efter min mening fejlregistreret som residential area i OSM-databasen.

Størrelse af boligbygninger

Fordeling over bygningsstørrelser for residential area

areal=x , huse for $x > 80$ and $x < 170$ (øvre grænse bør nok hæves til fx 230 m^2), **findes meget hurtigt, 0.008 s**

```
t1=time.time()
parcelhuse = buildings_residential[ buildings_residential['area'].apply(lambda x:  $x > 80$  and  $x < 170$ ) ] #1092
t2= time.time()
print('dt[s]=',t2-t1) # 0.007980108261108398
parcelhuse["area"].describe()
Out[27]:
count    1092.0000    antal i alt
mean      135.5128    i gennemsnit  $135.5 \text{ m}^2$ 
std        24.9553
min         80.2128
25%       121.9719
50%       140.2932
75%       155.8284
max       169.9703
```

Fordelingen af restauranter i Albertslund

statistik over distance for all nodes to nearby restaurant within 1000m:

CPU times: total: 1.56 s

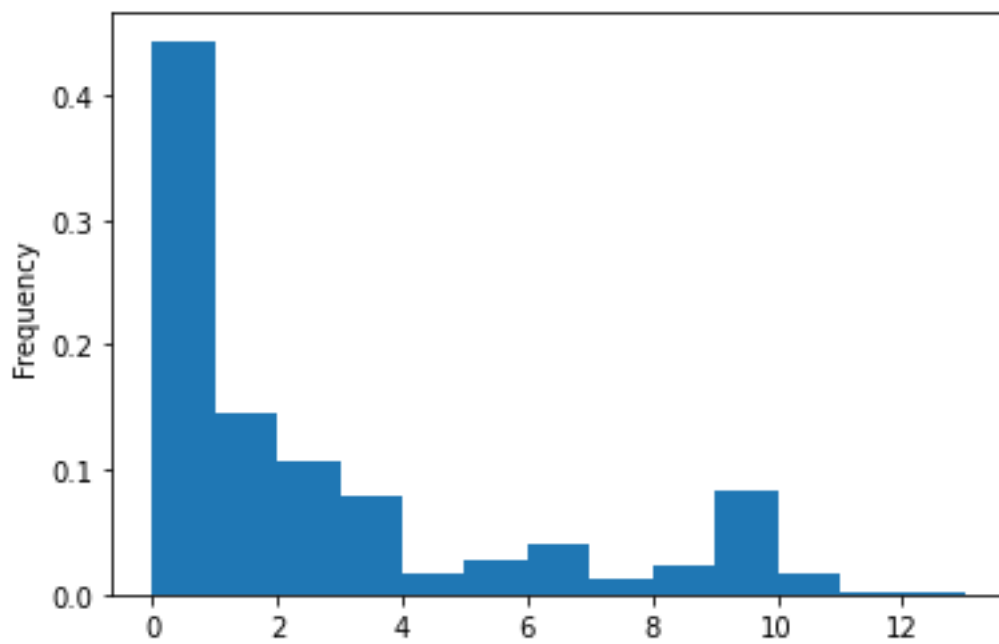
Wall time: 230 ms

Out[228]:

count 11335.00000 , antal noder i denne beregning. ovenfor 20700; netværket afhænger af det præcise areal og eventuelle simplificeringer

```
mean      2.26061 #each node have little over 2 restaurants within 1000m
std        3.08611
min         0.00000
25%         0.00000
50%         1.00000
75%         3.00000
max        13.00000
```

```
accessibility.plot(kind = 'hist',bins=13,
                    weights=np.ones_like(accessibility) / len(accessibility)) # frekvenser
```

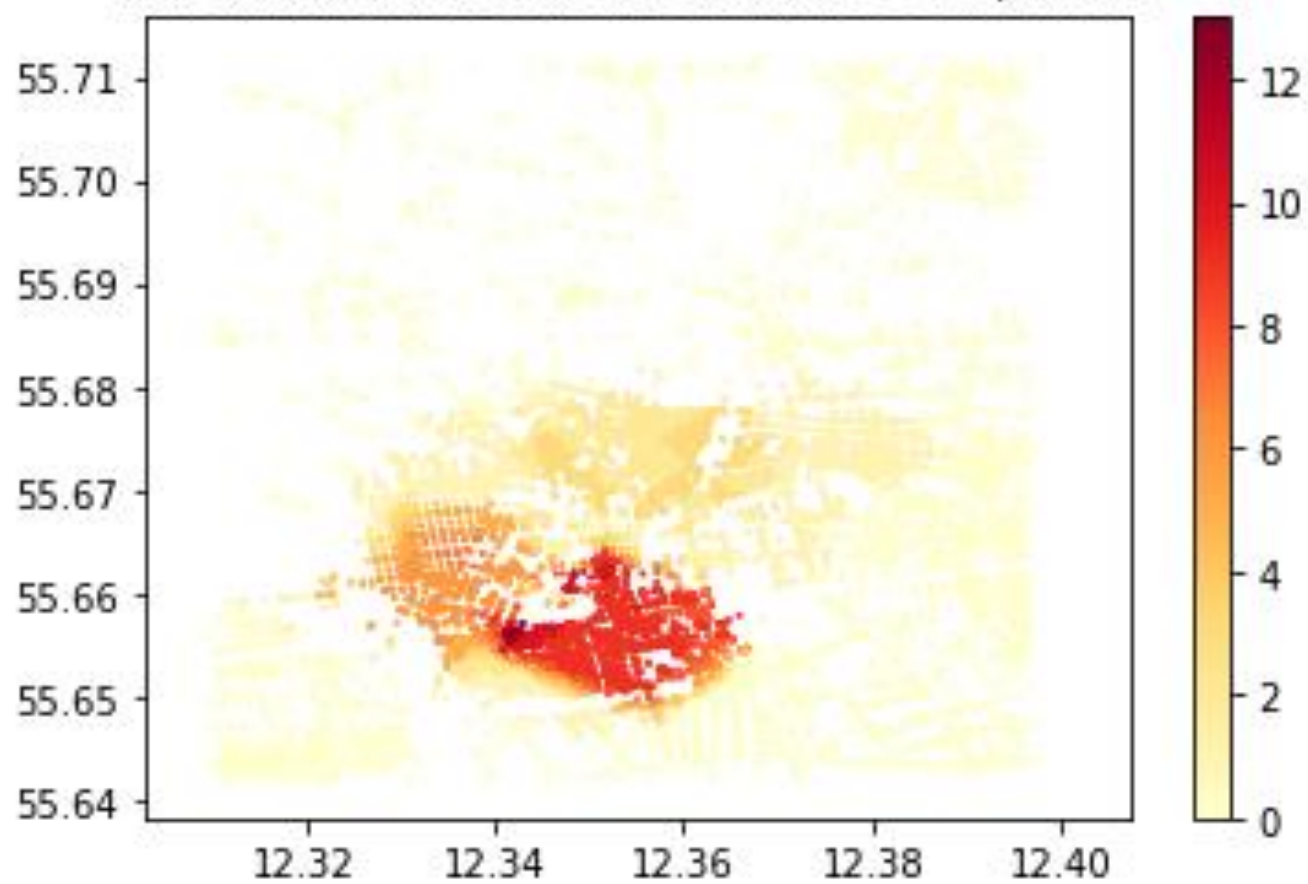


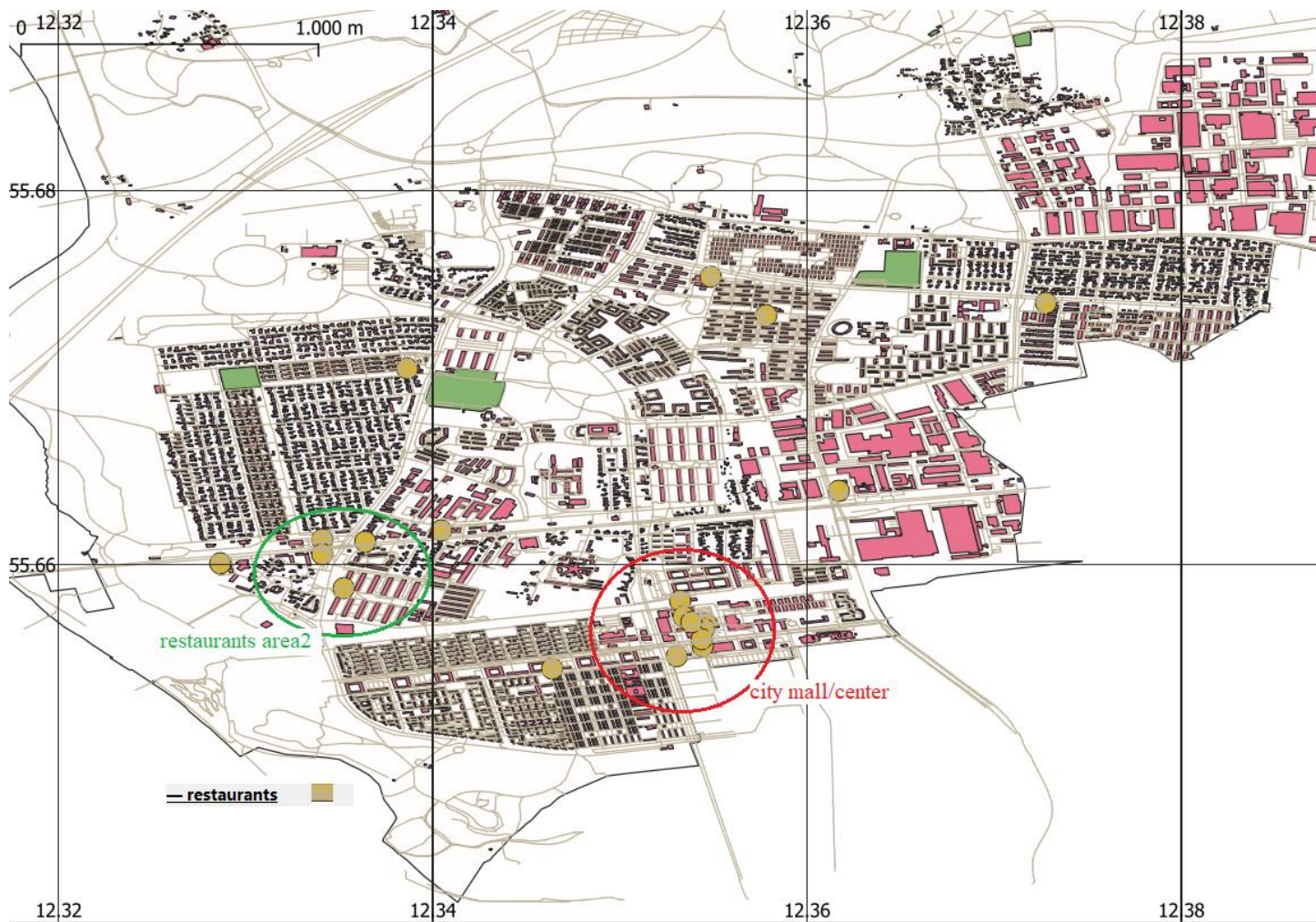
Ca 45% noder har 0 restaurant indenfor 1000m, har altså afstand >1000m fra nærmeste restaurant.

Plot

```
tit='Albertslund: Restaurants within 1000m, all 20'  
plt.title(tit)  
plt.scatter(network.nodes_df.x, network.nodes_df.y,  
            c=accessibility, s=1, cmap='YlOrRd',  
            )  
cb = plt.colorbar()  
plt.show()
```

Albertslund: Restaurants within 1000m, all 20



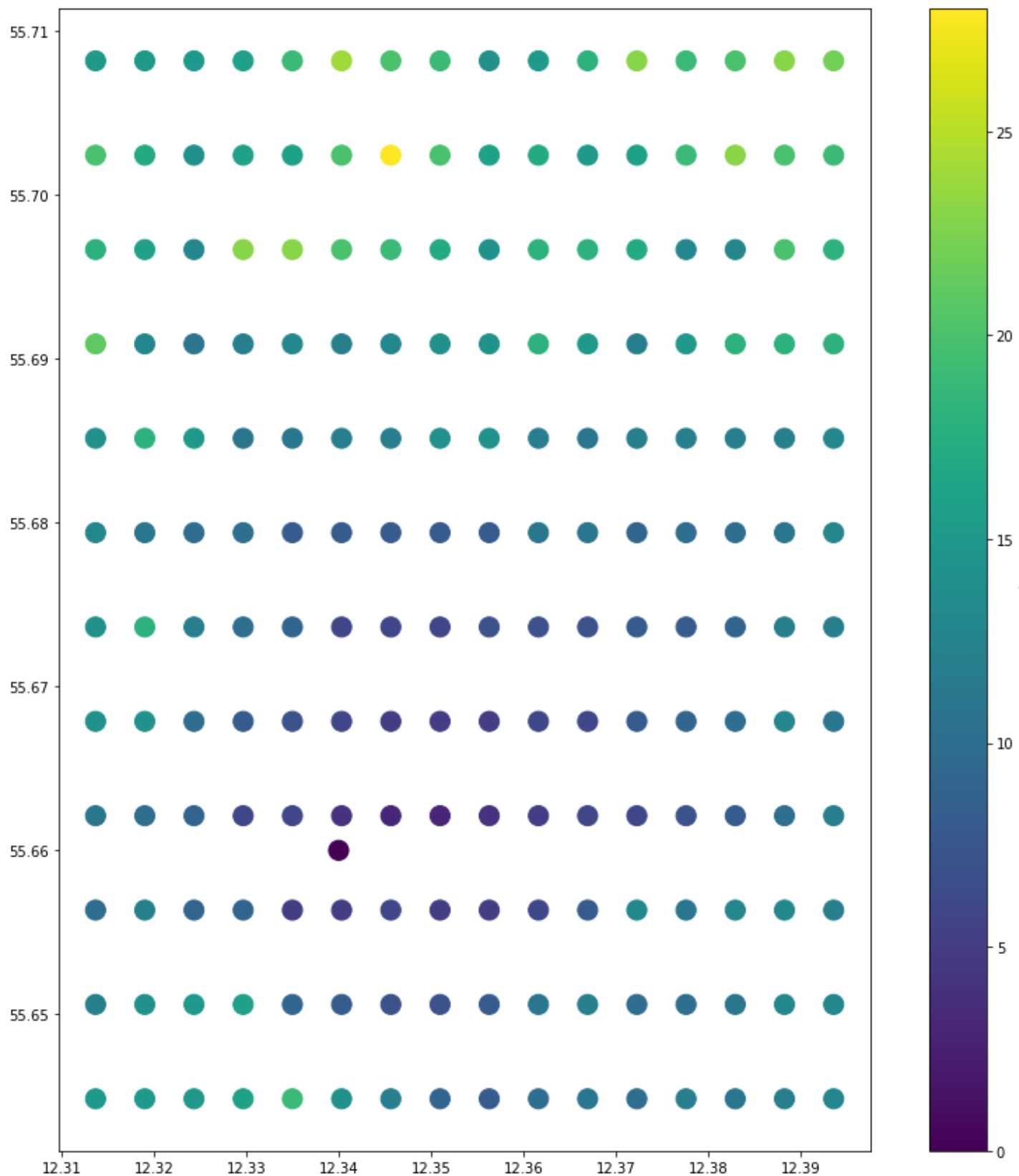


Seems reasonable: area around (12.34,56.655) have 13 restaurants within 1000m, it is within 1000m from restaurants in city center as well as area2.

***Travel-Time til bestemt netværks-destination

Med Travel Time API kan udviklere beregne bedste rute og rutens køretid mellem mange afgangspunkt til mange målpunkter.

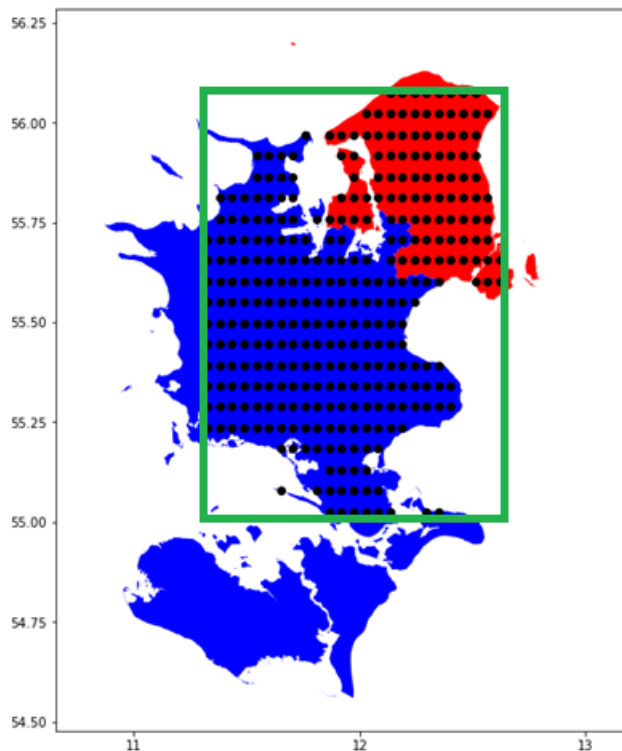
Først lille beregning med køretider til centrum i Albertslund, ca (12.34, 55.66), stor sort cirkel, fra steder i Albertslund, herunder ses plot af rå dataset, køretid i min. Jeg har først lavet regelmæssigt punkter i et gitter, og så beregnet køretiden ved kald til Travel Time fra hvert punkt til centrum.



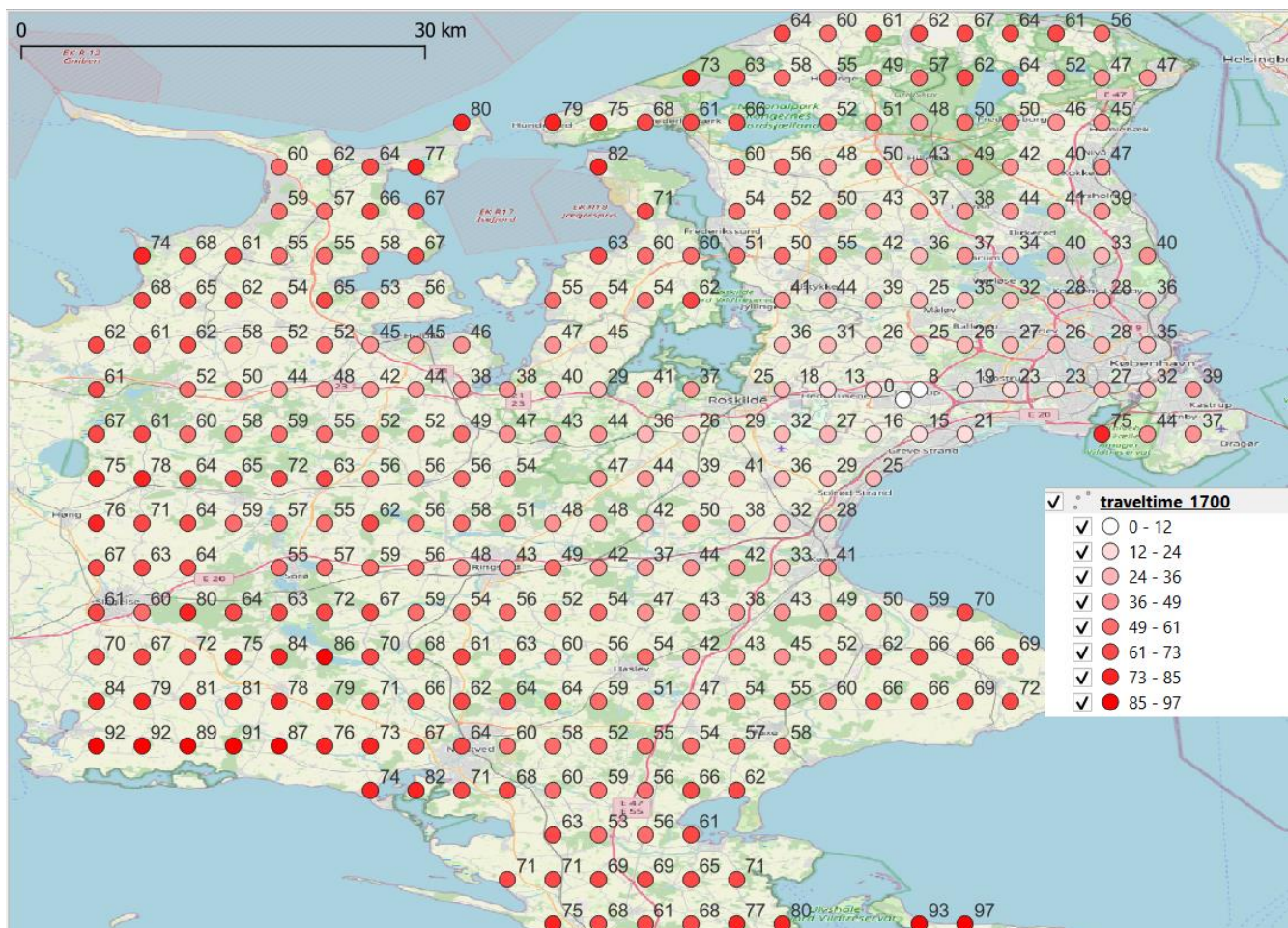
Det rå datasæt, med køretider tilføjet, læses ind i QGIS med OpenStreetMap som underlag, vist nedenfor. Køretid til Albertslund centrum, hvide cirkel med køretids-label=0, markeret med grøn firkant. Det ser fornuftigt ud.



Traveltime til City2 fra steder på Sjælland, ca kl 17, onsdag, altså tæt ved myldretiden(bort fra København)



Har først lavet regelmæssigt gitter af sorte punkter indenfor grønne firkant, punkterne klippes med 'Region Sjælland'(blå)+'Hovedstaden'(rød) i OSM-datapakken, så ca 1/3 fjernes[ude i havområder!], kortet ovenfor viser ca 330. Så beregnes køretiden fra hvert punkt til City2, stort indkøbscenter ved Høje Tåstrup station. 8 af punkterne er 'unreachable', ligger 'udenfor lands lov og kørevej'. Punkterne med køretider tilføjet gemmes i et lag, og indlæses i QGIS, vist nedenfor:



City2 er hvide cirkel med label=0min, midt mellem Roskilde og København. **Langs motorvejene er tiderne relativt korte.** Har tjekket med Googlemaps køretider, det er ikke helt tosset. Lang køretid på 39min for punkt på Amager, må skyldes del kørsel gennem byen, beregning gælder jo for myldretidstrafik.