

bb

July 3, 2018

```
In [1]: """
        Import required modules and data, and transform all coordinates in the data to WSG84.
        This may take a while (a minute or more, on a typical laptop).

        """
import pandas as pd
import math
from pyproj import Proj, transform

def getDistance(k1, k2):
    """ Great-circle distances """
    lat1 = float(k1["lat"])
    lon1 = float(k1["lon"])
    lat2 = float(k2["lat"])
    lon2 = float(k2["lon"])

    radius = 6371 # km

    dlat = math.radians(lat2 - lat1)
    dlon = math.radians(lon2 - lon1)
    a = math.sin(dlat/2) * math.sin(dlat/2) + math.cos(math.radians(lat1)) \
        * math.cos(math.radians(lat2)) * math.sin(dlon/2) * math.sin(dlon/2)
    c = 2 * math.atan2(math.sqrt(a), math.sqrt(1 - a))
    d = radius * c

    return math.floor(d)

# Useful constants
SWEREF = Proj(init='epsg:3006')
WGS84 = Proj(init='epsg:4326')
NOW = "2017-06-30"
THEN = "2000-01-01"

# Load data
bb = pd.read_csv("bb.csv")
```

```

# There was a problem with the geometry in the earilier SCB file. For now, loadonly on
pop = pd.read_csv("population_1.csv")
coords = pop.apply(lambda x: pd.Series(transform(SWEREF, WGS84, x.X, x.Y)), axis=1)
coords.columns = ["x", "y"]
pop = pop.join(coords)

"""
population = [pd.read_csv("population_0.csv"), pd.read_csv("population_1.csv")]
for i, pop in enumerate(population):
    coords = pop.apply(lambda x: pd.Series(transform(sweref, wgs84, x.X, x.Y)), axis=1)
    coords.columns = ["x", "y"]
    population[i] = pop.join(coords)
"""

print("Sanity check: The following line should say (56, 113332):")
(len(bb), len(pop))

```

Sanity check: The following line should say (56, 113332):

Out[1]: (56, 113332)

```

In [2]: """
    Add distance to the nearest maternity clinic for each square in the population grid, .
    This may take a while (ten minutes or more on a typical laptop).
"""

# ISO dates work nicely in string comparisons.
f_now = (bb.edate.astype(str) > NOW) & (bb.sdate.astype(str) <= NOW)
f_then = (bb.edate.astype(str) > THEN) & (bb.sdate.astype(str) <= THEN)

def nearest_bb(x, y, bb_selection):
    mindist = bb_selection.apply(lambda row: getDistance({"lat": row.Y, "lon": row.X},
    return mindist

pop["dist_to_nearest_bb_now"] = pop.apply(lambda ruta: nearest_bb(ruta.x, ruta.y, bb[f
pop["dist_to_nearest_bb_then"] = pop.apply(lambda ruta: nearest_bb(ruta.x, ruta.y, bb[f

In [3]: """ Add proper municipality names from https://github.com/jplusplus/statscraper-dataty
"""

municipalities = pd.read_csv("datatypes/values/regions/sweden/municipalities.csv")
pop = pop.merge(municipalities[["label", "dialect:numerical"]], left_on="mcode", right
pop.mname = pop.mname.replace("Uppsala kommun före 2003", "Uppsala kommun") # Don't d

```

1 The nation as a whole

1.1 Average distance, now and then

```

In [4]: # Weighted average distance in the whole nation
total_pop = pop["pop"].sum()

```

```

avg_0 = (pop.dist_to_nearest_bb_then * (pop["pop"] / total_pop)).sum()
avg_1 = (pop.dist_to_nearest_bb_now * (pop["pop"] / total_pop)).sum()

print("Average distance, 2000 and 2017 (km):")
(avg_0, avg_1)

```

Average distance, 2000 and 2017 (km):

Out[4]: (15.78869734322321, 17.132958118712338)

1.2 Number of people with more than X km

```

In [5]: # Number of people with more than x km
distances = [40, 50, 60, 70, 80, 90, 100]
print("Dist.\t2000\t2017")
print("=====")
for distance in distances:
    num_0 = pop[pop.dist_to_nearest_bb_then > distance]["pop"].sum()
    num_1 = pop[pop.dist_to_nearest_bb_now > distance]["pop"].sum()
    print("\t".join([str(distance), str(num_0), str(num_1)]))

```

Dist.	2000	2017
40	966703	1172884
50	544902	684108
60	259696	370779
70	158141	260538
80	106929	165619
90	68702	104121
100	40073	72665

2 By municipality

2.1 Key figures for a specific municipality (using Kiruna as an example)

```

In [6]: # Key figures for a specific municipality
municipality = "Kiruna kommun"

local = pop[pop.mname == municipality]
total_pop = local["pop"].sum()
avg_now = (local.dist_to_nearest_bb_now * (local["pop"] / total_pop)).sum()
avg_then = (local.dist_to_nearest_bb_then * (local["pop"] / total_pop)).sum()
max_dist_now = local.dist_to_nearest_bb_now.max()
max_dist_then = local.dist_to_nearest_bb_then.max()
num_above_100_now = local[local.dist_to_nearest_bb_now > 100]["pop"].sum()
num_above_100_then = local[local.dist_to_nearest_bb_then > 100]["pop"].sum()

```

```

print("Total population:", total_pop)
print("Average distance 2000:", avg_then)
print("Average distance 2017:", avg_now)
print("Maximum distance 2000:", max_dist_then)
print("Maximum distance 2017:", max_dist_now)
print("Number of people with more than 100 km, 2000:", num_above_100_then)
print("Number of people with more than 100 km, 2017:", num_above_100_now)

```

```

Total population: 23120
Average distance 2000: 12.49212802768166
Average distance 2017: 85.480276816609
Maximum distance 2000: 126
Maximum distance 2017: 194
Number of people with more than 100 km, 2000: 696
Number of people with more than 100 km, 2017: 1694

```

2.2 Maximum distances by municipality

```

In [7]: # Maximum distances by municipality, 2017
m_groups = pop.groupby("mname")
m_groups.dist_to_nearest_bb_now.max().sort_values()

```

```

Out[7]: mname
Solna kommun          3
Danderyds kommun      5
Sundbybergs kommun    5
Burlövs kommun        9
Partille kommun       9
Salems kommun         10
Malmö kommun          10
Huddinge kommun       11
Lomma kommun          12
Täby kommun           13
Hammarö kommun        13
Lidingö kommun        14
Sollentuna kommun     14
Staffanstorps kommun  14
Oxelösunds kommun     14
Stockholms kommun     14
Botkyrka kommun       15
Mölndals kommun       16
Göteborgs kommun      16
Järfälla kommun       17
Nacka kommun          17
Helsingborgs kommun   19
Bjuvs kommun          20

```

Kävlinge kommun	20
Karlskoga kommun	21
Skövde kommun	21
Upplands Väsby kommun	22
Vaxholms kommun	22
Nykvarns kommun	22
Trollhättans kommun	23
...	
Åsele kommun	104
Rättviks kommun	104
Ånge kommun	104
Haparanda kommun	104
Strömstads kommun	104
Överkalix kommun	109
Bergs kommun	111
Hagfors kommun	112
Orsa kommun	118
Sollefteå kommun	119
Krokoms kommun	121
Åre kommun	128
Jokkmokks kommun	130
Ljusdals kommun	130
Mora kommun	135
Övertorneå kommun	147
Arvidsjaurs kommun	149
Gällivare kommun	154
Pajala kommun	156
Härjedalens kommun	160
Malung-Sälens kommun	169
Dorotea kommun	176
Torsby kommun	188
Kiruna kommun	194
Strömsunds kommun	195
Sorsele kommun	200
Älvdalens kommun	202
Arjeplogs kommun	204
Vilhelmina kommun	217
Storumans kommun	252

Name: dist_to_nearest_bb_now, Length: 290, dtype: int64

```
In [8]: # Maximum distances by municipality, 2000
m_groups = pop.groupby("mname")
m_groups.dist_to_nearest_bb_then.max().sort_values()
```

```
Out [8]: mname
Solna kommun      3
Sundbybergs kommun 5
Danderyds kommun  5
```

Partille kommun	9
Burlövs kommun	9
Malmö kommun	10
Salems kommun	10
Huddinge kommun	11
Lomma kommun	12
Täby kommun	13
Lidingö kommun	13
Hammarö kommun	13
Oxelösunds kommun	14
Stockholms kommun	14
Sollentuna kommun	14
Staffanstorps kommun	14
Botkyrka kommun	15
Göteborgs kommun	16
Mölndals kommun	16
Nacka kommun	16
Järfälla kommun	17
Helsingborgs kommun	19
Bjuvs kommun	20
Kävlinge kommun	20
Karlskoga kommun	21
Skövde kommun	21
Upplands Väsby kommun	22
Nykvarns kommun	22
Vaxholms kommun	22
Öckerö kommun	23
...	
Arvika kommun	97
Överkalix kommun	97
Dals-Eds kommun	97
Hagfors kommun	98
Malung-Sälens kommun	99
Årjängs kommun	100
Malå kommun	102
Eda kommun	102
Ånge kommun	104
Åsele kommun	104
Strömstads kommun	104
Ljusdals kommun	106
Bergs kommun	111
Gällivare kommun	114
Krokoms kommun	121
Övertorneå kommun	122
Torsby kommun	125
Kiruna kommun	126
Åre kommun	128
Jokkmokks kommun	130

Pajala kommun	142
Härjedalens kommun	143
Arvidsjaur kommun	147
Älvdalens kommun	169
Dorotea kommun	176
Strömsunds kommun	195
Sorsele kommun	200
Arjeplogs kommun	204
Vilhelmina kommun	217
Storumans kommun	252

Name: dist_to_nearest_bb_then, Length: 290, dtype: int64

2.3 Number of people with more than 100 km, by municipality

In [9]: *# Number of people with more than 100 km, by municipality, 2017*
 pop[pop.dist_to_nearest_bb_now > 100].groupby("mname")["pop"].sum().sort_values(ascending=True)

Out [9]: mname

Malung-Sälens kommun	9380
Härjedalens kommun	9173
Haparanda kommun	7069
Älvdalens kommun	6855
Arvidsjaur kommun	5717
Torsby kommun	4727
Pajala kommun	4517
Övertorneå kommun	4289
Strömsunds kommun	4184
Arjeplogs kommun	2803
Dorotea kommun	2672
Storumans kommun	2243
Vilhelmina kommun	2058
Sorsele kommun	1953
Kiruna kommun	1694
Sollefteå kommun	1194
Ljusdals kommun	662
Åre kommun	374
Strömstads kommun	319
Mora kommun	283
Jokkmokks kommun	139
Överkalix kommun	85
Krokoms kommun	78
Hagfors kommun	42
Orsa kommun	32
Gällivare kommun	28
Åsele kommun	28
Bergs kommun	23
Rättviks kommun	16
Eda kommun	13

```

Ånge kommun          13
Malå kommun           2
Name: pop, dtype: int64

```

```

In [10]: # Number of people with more than 100 km, by municipality, 2000
pop[pop.dist_to_nearest_bb_then > 100].groupby("mname")["pop"].sum().sort_values(ascen

```

```

Out[10]: mname
Härjedalens kommun    8515
Arvidsjaurs kommun    5433
Pajala kommun         4447
Strömsunds kommun     2991
Arjeplogs kommun      2803
Dorotea kommun        2659
Torsby kommun         2621
Storumans kommun      2243
Älvdalens kommun      2124
Vilhelmina kommun     2058
Sorsele kommun        1953
Kiruna kommun         696
Övertorneå kommun     513
Åre kommun            374
Strömstads kommun     319
Jokkmokks kommun      139
Krokoms kommun        78
Ljusdals kommun       32
Bergs kommun          23
Åsele kommun          18
Eda kommun            13
Ånge kommun           13
Gällivare kommun      6
Malå kommun           2
Name: pop, dtype: int64

```

2.4 Percentage of people with more than 100 km, by municipality

```

In [11]: # Share of people with more than 100 km, by municipality, 2017
(pop[pop.dist_to_nearest_bb_now > 100].groupby("mname")["pop"].sum() / pop.groupby("m

```

```

Out[11]: mname
Arjeplogs kommun      1.000000
Dorotea kommun        0.999252
Älvdalens kommun      0.975107
Malung-Sälens kommun  0.938563
Övertorneå kommun     0.933812
Härjedalens kommun    0.896151
Arvidsjaurs kommun    0.885533
Sorsele kommun        0.752891
Pajala kommun         0.731024

```


Haparanda kommun	0.720298
Torsby kommun	0.394410
Storumans kommun	0.377102
Strömsunds kommun	0.355722
Vilhelmina kommun	0.301274
Kiruna kommun	0.073270
Sollefteå kommun	0.060680
Ljusdals kommun	0.034986
Åre kommun	0.034736
Jokkmokks kommun	0.027239
Strömstads kommun	0.025168

Name: pop, dtype: float64

```
In [12]: # Share of people with more than 100 km, by municipality, 2000
(pop[pop.dist_to_nearest_bb_then > 100].groupby("mname")["pop"].sum() / pop.groupby("mname")["pop"].sum()).sort_values()
```

```
Out[12]: mname
Arjeplogs kommun    1.000000
Dorotea kommun      0.994390
Arvidsjaurs kommun  0.841543
Härjedalens kommun  0.831868
Sorsele kommun      0.752891
Pajala kommun       0.719696
Storumans kommun    0.377102
Älvdalens kommun    0.302134
Vilhelmina kommun   0.301274
Strömsunds kommun   0.254293
Torsby kommun       0.218690
Övertorneå kommun   0.111692
Åre kommun          0.034736
Kiruna kommun       0.030104
Jokkmokks kommun    0.027239
Strömstads kommun   0.025168
Åsele kommun        0.006388
Krokoms kommun      0.005315
Bergs kommun        0.003215
Ljusdals kommun     0.001691
Name: pop, dtype: float64
```

2.5 Weighted average distances, by municipality

```
In [13]: # Weighted average distances 2017, by municipality
```

```
pop["product_now"] = (pop.dist_to_nearest_bb_now * pop["pop"])
(pop.groupby("mname").product_now.sum() / pop.groupby("mname")["pop"].sum()).sort_values()
```

```
Out[13]: mname
Solna kommun        2.034533
Danderyds kommun    2.076101
```

Karlskoga kommun	2.260476
Malmö kommun	2.382794
Helsingborgs kommun	3.541761
Ystads kommun	3.703545
Västerås kommun	3.928438
Stockholms kommun	3.965619
Sundbybergs kommun	4.086889
Mölndals kommun	4.096337
Partille kommun	4.266087
Lunds kommun	4.283928
Göteborgs kommun	4.306591
Örebro kommun	4.375090
Eskilstuna kommun	4.515851
Borås kommun	4.605348
Linköpings kommun	4.950215
Södertälje kommun	5.021872
Hammarö kommun	5.031993
Skövde kommun	5.155763
Gävle kommun	5.243858
Halmstads kommun	5.305799
Huddinge kommun	5.378977
Botkyrka kommun	5.396720
Trollhättans kommun	5.606090
Uppsala kommun	5.670384
Norrköpings kommun	5.769828
Nyköpings kommun	5.782631
Östersunds kommun	5.785038
Kalmar kommun	5.973615
...	
Jokkmokks kommun	70.265726
Ragunda kommun	72.233781
Bengtsfors kommun	72.276725
Ånge kommun	72.761744
Mora kommun	74.036154
Ovanåkers kommun	74.126058
Vansbro kommun	74.916853
Hagfors kommun	75.174200
Åsele kommun	77.382186
Orsa kommun	78.584934
Sollefteå kommun	79.708492
Årjängs kommun	81.433415
Eda kommun	84.571983
Överkalix kommun	84.718401
Kiruna kommun	85.480277
Strömstads kommun	92.516213
Strömsunds kommun	96.310917
Haparanda kommun	98.438557
Torsby kommun	104.193325

Vilhelmina kommun	105.679696
Pajala kommun	107.468199
Arvidsjaurs kommun	111.069083
Övertorneå kommun	116.302852
Malung-Sälens kommun	116.478787
Sorsele kommun	118.446029
Dorotea kommun	119.025804
Härjedalens kommun	121.207796
Storumans kommun	123.054808
Älvdalens kommun	132.402418
Arjeplogs kommun	164.167321

Length: 290, dtype: float64