

Explore GeoHack data

```
In [1]: !ls data
```

```
leedsAll.dbf   leedsAll.shp   leedsRoads.geojson  mwayRF.csv  
leedsAll.prj  leedsAll.shx   leedsTags.csv       README.txt  
leedsAll.qpj  leedsOSMTags.csv mwayBR.csv
```

```
In [2]: import pandas, numpy
```

```
/usr/lib/python3/dist-packages/pandas/io/excel.py:626: UserWarning:  
Installed openpyxl is not supported at this time. Use >=1.6.1 and <2  
.0.0.  
.format(openpyxl_compat.start_ver, openpyxl_compat.stop_ver))
```

leedsAll shapefile

The leedsAll.xxx files together form a shapefile specification. See <https://github.com/GeospatialPython/pyshp/blob/master/README.md> for some explanation of the module used to read this specification.

```
In [3]: import shapefile
```

```
In [4]: sf = shapefile.Reader('data/leedsAll')  
shapes = sf.shapes()
```

```
In [5]: len(shapes)  # number of shapes
```

```
Out[5]: 31026
```

```
In [6]: shapes[0].bbox  # bounding box
```

```
Out[6]: [-1.627225, 53.798138, -1.626639, 53.798321]
```

```
In [7]: shapes[0].parts  # ??? [0] means no parts
```

```
Out[7]: [0]
```

```
In [8]: shapes[0].points
```

```
Out[8]: [[-1.627225, 53.798321], [-1.62682, 53.798183], [-1.626639, 53.798138]]
```

```
In [9]: shapes[0].shapeType  # according to some specification
```

```
Out[9]: 3
```

In [10]: `sf.fields` # every shape has a record with the following fields

```
Out[10]: [('DeletionFlag', 'C', 1, 0),
          ['id', 'N', 24, 15],
          ['wayNmId', 'N', 24, 15],
          ['lengthMt', 'N', 24, 15],
          ['rtngFcI', 'N', 24, 15],
          ['flow', 'C', 80, 0],
          ['island', 'N', 24, 15],
          ['points', 'C', 80, 0],
          ['elevtns', 'C', 80, 0],
          ['distncs', 'C', 80, 0]]
```

In [11]: `records = sf.records()`
`len(records), len(shapes)` # number of records equals number of shapes

Out[11]: (31026, 31026)

In [12]: `records[0]`

```
Out[12]: [1628965.0,
          596060.0,
          44.0,
          272.0,
          'with',
          7.0,
          b'
          ',
          '60,58,57',
          '31,13']
```

leedsTags.csv

In [13]: `lt = pandas.read_csv('data/leedsTags.csv')`

In [14]: `len(lt), len(shapes)` # as many rows in this table as there are shapes in the leedsAll.xxx shapefile

Out[14]: (31026, 31026)

In [15]: `lt.head()`

Out[15]:

	Unnamed: 0	id	highway	cycleway	access	foot	bicycle	oneway	routeId	route
0	4443	1628965	secondary	NaN	no	no	yes	yes	0	272
1	5036	1709451	unclassified	NaN	yes	yes	yes	yes	0	337
2	5037	1709456	tertiary	NaN	yes	yes	yes	no	0	67
3	5038	1709460	unclassified	NaN	yes	yes	yes	no	0	337
4	7542	2340822	trunk	NaN	no	no	yes	yes	0	396

```
In [16]: # Check that numbers in ids column are the same as the ones in the s
         hapefile
         ids_from_csv = lt.id.values
         ids_from_sf = [int(r[0]) for r in sf.iterRecords()]
         (ids_from_csv == ids_from_sf).all()
```

Out[16]: True

```
In [17]: # Possible values (with counts) for highway column
         lt.groupby('highway').size()
```

```
Out[17]: highway
         bridleway          245
         cycleway          489
         footway          5344
         living_street        6
         motorway_link       164
         path              276
         pedestrian         123
         primary            614
         primary_link        55
         residential      13097
         road              49
         secondary         294
         service          5270
         steps             287
         tertiary         1256
         track             885
         trunk             744
         trunk_link        139
         unclassified      1689
         dtype: int64
```

```
In [18]: lt.groupby('cycleway').size()
```

```
Out[18]: cycleway
         lane              58
         opposite_lane      5
         opposite_track      3
         share_busway        7
         shared              5
         track             534
         dtype: int64
```

```
In [19]: lt.groupby('access').size()
```

```
Out[19]: access
         designated         3
         destination       87
         no              7940
         permissive       200
         private          22
         yes            22533
         dtype: int64
```

```
In [20]: lt.groupby('foot').size()
```

```
Out[20]: foot
designated      4805
destination      23
no             2130
permissive      179
yes           23880
dtype: int64
```

```
In [21]: lt.groupby('bicycle').size()
```

```
Out[21]: bicycle
designated      355
destination      73
dismount      5842
permissive      140
private         11
yes          24590
dtype: int64
```

```
In [22]: lt.groupby('oneway').size()
```

```
Out[22]: oneway
-1          26
no        28461
yes        2539
dtype: int64
```

```
In [23]: lt.groupby('routeId').size()
```

```
Out[23]: routeId
0          30443
17883         27
357215        37
1548002        73
2054077        34
3177071        61
3177124        48
3177601        64
3179002        72
3179113        67
3723504        99
3723713         1
dtype: int64
```

```
In [24]: lt.routingFactorId.min(), lt.routingFactorId.max()
```

```
Out[24]: (2, 657)
```

mwayBR.csv

```
In [25]: mbr = pandas.read_csv("data/mwayBR.csv", encoding="latin1")
```

```
In [26]: len(mbr)
```

```
Out[26]: 1585
```

In [27]: `mbr.head()`

Out[27]:

	Unnamed: 0	id	originalTags	name	network	cnRef	sumLengthsKm	rating
0	1	2649	network=lc\n\tref=4 \troute=bicycle \ttype=route	Local cycle network 4	lc\n	4	1	suggest
1	2	2650	network=lc\n\tref=4A \troute=bicycle \ttype=route	Local cycle network 4A	lc\n	4A	3	suggest
2	3	2695	network=lc\n\tref=29 \troute=bicycle \ttype=route	Local cycle network 29	lc\n	29	7	suggest
3	4	2696	name=London Cycle Network Route 3\n\tnetwork=lc\n...	London Cycle Network Route 3	lc\n	3	19	suggest
4	5	2706	name=London Cycle Network Route 2\n\tnetwork=lc\n...	London Cycle Network Route 2	lc\n	2	14	suggest

In [28]: `# Get possible values of routeId in leedsTags table
gb = lt.groupby('routeId')
gb.groups.keys()`

Out[28]: `dict_keys([0, 3177601, 1548002, 3177124, 3723713, 3179113, 3177071,
3723504, 3179002, 17883, 2054077, 357215])`

In [29]: `# Get entries in mwayBR table corresponding to these id's`
`mbr[mbr.id.isin(gb.groups.keys())]`

Out[29]:

	Unnamed: 0	id	originalTags	name	network	cnRef	sumLengths
71	72	17883	name=NCN National Route 66\tnetwork=ncn\topera...	NCN National Route 66	ncn	66	179
360	361	357215	name=NCN National Route 67\tnetwork=ncn\topera...	NCN National Route 67	ncn	67	181
641	642	1548002	name=West Yorkshire Cycle Route\tnetwork=rcn\t...	West Yorkshire Cycle Route	rcn	NaN	49
834	835	2054077	name=Roundhay Park to Temple Newsam Core Cycle...	Roundhay Park to Temple Newsam Core Cycle Route	lcn	RP-TN	15
1234	1235	3177071	name=Garforth to Leeds City Centre Core Cycle ...	Garforth to Leeds City Centre Core Cycle Route	lcn	G-CC	12
1235	1236	3177124	name=Armley to Leeds City Centre Core Cycle Ro...	Armley to Leeds City Centre Core Cycle Route	lcn	Ar-CC	7
1237	1238	3177601	name=Middleton to Leeds City Centre Core Cycle...	Middleton to Leeds City Centre Core Cycle Route	lcn	M-CC	9
1239	1240	3179002	ref=A-CC\troute=bicycle\troute=route	Alwoodley to Leeds City Centre	lcn	A-CC	11

18/06/14 16:17

mwayRF.csv

```
In [30]: mrf = pandas.read_csv("data/mwayRF.csv")
```

```
In [31]: len(mrf)
```

```
Out[31]: 676
```

```
In [32]: mrf.head()
```

```
Out[32]:
```

	Unnamed: 0	id	color	cyclable	walkable	speed	quietness	pause	rfHash
0	1	1	#EAEAFF	no	no	1	1	0	0
1	2	2	#00DD00	yes	yes	26	90	0	3667591168
2	3	3	#FCC100	yes	yes	12	89	7	3649896441
3	4	4	#00DD00	yes	yes	9	94	7	3733585913
4	5	5	#FCC100	yes	yes	24	80	1	3499687935

```
In [33]: # Check that id's are continuous
(mrf.id == numpy.arange(1, len(mrf)+1)).all()
```

```
Out[33]: True
```

leedsOSMTags.csv

```
In [34]: lot = pandas.read_csv("data/leedsOSMTags.csv")
```

```
In [35]: len(lot)
```

```
Out[35]: 63450
```

```
In [36]: lot.head()
```

```
Out[36]:
```

	Unnamed: 0	id	k	v
0	13792	1628965	highway	secondary
1	13793	1628965	name	Pudsey Road
2	13794	1628965	oneway	yes
3	13795	1628965	ref	B6154
4	15603	1709451	highway	unclassified

This corresponds to data from OpenStreetMap; see <http://www.openstreetmap.org/way/1628965>

```
In [37]: # Check that there are as many items in leedsAll as unique ids in th
e above table
len(lot.groupby('id')), len(lt)
```

```
In [38]: # Check that the items are the same
ids_from_lot = [id for id, _ in lot.groupby('id')]
ids_from_sf == ids_from_sf
```

Out[38]: True

leedsRoads.geojson

```
In [39]: lr = pandas.read_json("data/leedsRoads.geojson")
```

```
In [40]: len(lr), len(shapes) # same number as in shapefile
```

Out[40]: (31026, 31026)

```
In [41]: lr.head()
```

Out[41]:

	features	type
0	{'properties': {'wayNmId': 596060.0, 'lngthMt'...	FeatureCollection
1	{'properties': {'wayNmId': 91826.0, 'lngthMt':...	FeatureCollection
2	{'properties': {'wayNmId': 267314.0, 'lngthMt'...	FeatureCollection
3	{'properties': {'wayNmId': 294793.0, 'lngthMt'...	FeatureCollection
4	{'properties': {'wayNmId': 2465.0, 'lngthMt': ...	FeatureCollection

```
In [42]: lr.features[0]
```

Out[42]: {'properties': {'wayNmId': 596060.0,
'lngthMt': 44.0,
'rtngFcI': 272.0,
'points': None,
'id': 1628965.0,
'elevtns': '60,58,57',
'island': 7.0,
'distncs': '31,13',
'flow': 'with'},
'type': 'Feature',
'geometry': {'coordinates': [[-1.627225, 53.798321],
[-1.62682, 53.798183],
[-1.626639, 53.798138]],
'type': 'LineString'}}

Looks like the same information as in the leedsAll.xxx shapefile

```
In [42]:
```