

Final Notebook

March 14, 2022

1 Final Notebook

In previous notebooks, we have gathered general information about the Gauteng region and its community members, as well as information about social attitudes and BRT use. In this final notebook, I create some of the final outputs we need for our project such as: - interactive maps of the percentage of survey respondents who identify as African with the BRT lines overlayed - created tables to supplement some of our interactive maps - interactive maps of BRT use with the BRT lines overlayed - identified the wards in which BRT use is the highest (over 40%) which we then used to examine social attitudes in these wards - identified and isolated the wards through which the BRT passes and examined changes in social attitudes in those wards between 2011 and 2017

These outputs help us tie our previous explorations together.

1.1 Import necessary libraries

```
[1]: import pandas as pd

import geopandas as gpd

import contextily as ctx

import matplotlib.pyplot as plt

import plotly.express as px
```

```
/opt/conda/lib/python3.9/site-packages/geopandas/_compat.py:106: UserWarning:
The Shapely GEOS version (3.9.1-CAPI-1.14.2) is incompatible with the GEOS
version PyGEOS was compiled with (3.10.1-CAPI-1.16.0). Conversions between both
will be slow.
```

```
warnings.warn(
```

1.2 Import data

1.2.1 GCRO Quality of Life Survey Data

We had previously trimmed the GCRO data, eliminating all columns except those we would consistently use, and merged it with the respective ward geospatial data in order to avoid conducting a merge in every new notebook

```
[2]: gdf_2011 = gpd.read_file('data_2011_clean')
```

```
[3]: gdf_2017 = gpd.read_file('data_2017_clean')
```

1.2.2 BRT Data

We will be overlaying the BRT lines (provided by the GCRO) on our choropleth maps and creating an interactive map of the BRT stations. I hand coded the BRT stations using geojson.io and will import them here as well for use in our final presentation

```
[4]: BRT_line = gpd.read_file('Gauteng_BRT_lines.zip', encoding="utf-8")
```

```
[5]: BRT_stations = gpd.read_file('BRT_stations2.geojson')
```

1.3 Brief Data Exploration

I just want to ensure that all the necessary information is present in the data

```
[6]: gdf_2011.head(5)
```

```
[6]:      WardID LocalMunicipalityName DistrictMunicipalityName  A1_Pop_Group  \
0  74803012      Rand West City      West Rand      1.0
1  74803012      Rand West City      West Rand      1.0
2  74803012      Rand West City      West Rand      1.0
3  74803012      Rand West City      West Rand      1.0
4  74803012      Rand West City      West Rand      1.0
```

```
      A1_Pop_Group_recode  Q4_5 Train_Frequent_Trip_Transp  \
0      African      NaN
1      African      NaN
2      African      NaN
3      African      NaN
4      African      1.0
```

```
      Q4_5 BRT_Frequent_Trip_Transp  A_4_8_Walking_proxim_pub_transp  \
0      NaN      1.0
1      NaN      1.0
2      NaN      1.0
3      NaN      1.0
4      NaN      3.0
```

```
      A_4_8_Walking_proxim_pub_transp_recode  A_6_31_BW_trust  \
0      Up to 10 minutes walk      NaN
1      Up to 10 minutes walk      1.0
2      Up to 10 minutes walk      2.0
3      Up to 10 minutes walk      1.0
4      From 21 to 30 minutes walk      1.0
```

	A_6_31_BW_trust_recode	A_6_36_foreigners	A_6_36_foreigners_recode	African	\
0	None	3.0	Third person	40	
1	Strongly Agree	1.0	First person	40	
2	Agree	1.0	First person	40	
3	Strongly Agree	1.0	First person	40	
4	Strongly Agree	1.0	First person	40	

	Percent African	White	Percent White	\
0	100.0	0	0.0	
1	100.0	0	0.0	
2	100.0	0	0.0	
3	100.0	0	0.0	
4	100.0	0	0.0	

```

                                geometry
0 POLYGON ((27.73847 -26.28225, 27.72304 -26.289...
1 POLYGON ((27.73847 -26.28225, 27.72304 -26.289...
2 POLYGON ((27.73847 -26.28225, 27.72304 -26.289...
3 POLYGON ((27.73847 -26.28225, 27.72304 -26.289...
4 POLYGON ((27.73847 -26.28225, 27.72304 -26.289...

```

```
[7]: gdf_2017.head(5)
```

```

[7]:      WardID LocalMunicipalityName DistrictMunicipalityName A1_Pop_group \
0  79700060      Ekurhuleni      Ekurhuleni      African
1  79700060      Ekurhuleni      Ekurhuleni      African
2  79700060      Ekurhuleni      Ekurhuleni      African
3  79700060      Ekurhuleni      Ekurhuleni      African
4  79700060      Ekurhuleni      Ekurhuleni      African

```

	Q5_06_8_train	Q5_06_10_BRT_TRT	Q5_11_BRT_Freq	Q5_18_Walk_prox_public_transp	\
0	No	No	Never	Up to 10 minutes	
1	No	No	Never	From 11 to 20 minutes	
2	No	No	Never	Up to 10 minutes	
3	No	No	Never	Up to 10 minutes	
4	No	No	Never	From 11 to 20 minutes	

	Q9_10_BW_trust	\
0	Strongly agree	
1	Agree	
2	Disagree	
3	Neither agree nor disagree	
4	Agree	

	Q9_12_foreigners_scenario	African	\
0	Gauteng is for South Africans, foreigners sho...	54	

```

1  Foreigners are alright, but they must have le... 54
2  Gauteng is for South Africans, foreigners sho... 54
3  We have a lot in common with most of the fore... 54
4  Foreigners are alright, but they must have le... 54

```

```

    Percent African  White  Percent White  \
0          100.0      0          0.0
1          100.0      0          0.0
2          100.0      0          0.0
3          100.0      0          0.0
4          100.0      0          0.0

```

```

                                geometry
0  POLYGON ((28.16493 -26.35885, 28.16632 -26.358...
1  POLYGON ((28.16493 -26.35885, 28.16632 -26.358...
2  POLYGON ((28.16493 -26.35885, 28.16632 -26.358...
3  POLYGON ((28.16493 -26.35885, 28.16632 -26.358...
4  POLYGON ((28.16493 -26.35885, 28.16632 -26.358...

```

For our analysis, we look at the percentage of respondents who identify as African in each ward. To avoid creating the same dataframe in each notebook, when I cleaned the data, I saved the number of African and White respondents in each ward along with the corresponding ward percentage. However, in order to create an interactive choropleth map in Plotly, we need the proportion rather than the percent, as the code multiplies these values by 100. I will add another column with this data to the 2011 and 2017 data

1.4 Interactive Choropleth Maps

1.4.1 2011

Population Groups In order to get the proportion of Africans in each ward, I will use the crosstab command which allows me to see the number of respondents in each population group for each ward.

```

[8]: pop_group_11 = pd.crosstab(index=gdf_2011['WardID'],
                                columns=gdf_2011['A1_Pop_Group_recode'],
                                margins=True,
                                margins_name='Total Number')

pop_group_11

```

```

[8]: A1_Pop_Group_recode  African  Asian/Indian  Coloured  White  Total Number
WardID
74201001                 3             1           1       16           21
74201002                27             0           0        0           27
74201003                13             0           1        8           22
74201004                11             0           0       16           27
74201005                 7             2           1       16           26

```

...
79900102	39	0	0	1	40
79900103	67	0	0	1	68
79900104	3	0	0	1	4
79900105	36	1	0	1	38
Total Number	13338	491	572	2328	16729

[508 rows x 5 columns]

Now that I have the respondent answers listed by ward, I will create normalized data for the African population group by dividing the number of African respondents by the total number of respondents in each ward

```
[9]: pop_group_11['Percent African Decimal'] = pop_group_11['African']/
      ↪pop_group_11['Total Number']
```

pop_group_11

```
[9]: A1_Pop_Group_recode  African  Asian/Indian  Coloured  White  Total Number  \
WardID
74201001                3            1            1        16            21
74201002               27            0            0            0            27
74201003               13            0            1            8            22
74201004               11            0            0         16            27
74201005                7            2            1         16            26
...
79900102               39            0            0            1            40
79900103               67            0            0            1            68
79900104                3            0            0            1            4
79900105               36            1            0            1            38
Total Number          13338           491           572        2328          16729
```

```
A1_Pop_Group_recode  Percent African Decimal
WardID
74201001            0.142857
74201002            1.000000
74201003            0.590909
74201004            0.407407
74201005            0.269231
...
79900102            0.975000
79900103            0.985294
79900104            0.750000
79900105            0.947368
Total Number        0.797298
```

[508 rows x 6 columns]

I want to drop the 'Total Number' column from this dataframe as to not be included in my final output

```
[10]: pop_group_11 = pop_group_11.drop(['Total Number'])
```

I want to save a dataframe with *just* the newly created column so I will isolate that column

```
[11]: pop_group_11=pop_group_11[['Percent African Decimal']]

pop_group_11
```

```
[11]: A1_Pop_Group_recode  Percent African Decimal
WardID
74201001                0.142857
74201002                1.000000
74201003                0.590909
74201004                0.407407
74201005                0.269231
...
79900101                0.400000
79900102                0.975000
79900103                0.985294
79900104                0.750000
79900105                0.947368
```

```
[507 rows x 1 columns]
```

I want to merge this dataframe with the 2011 geospatial ward data in order to create a choropleth map so first I'll import the necessary data (the wards in 2011) and then conduct a merge.

```
[12]: wards_2011 = gpd.read_file('MDB_Wards_2011.geojson')
```

```
[13]: pop_group_11_gdf = wards_2011.merge(pop_group_11, on='WardID')

pop_group_11_gdf
```

```
[13]:
```

	OBJECTID	ProvinceCode	ProvinceName	LocalMunicipalityCode	WardNumber	\
0	330	GT	Gauteng	GT485	12	
1	4607	GT	Gauteng	GT485	12	
2	331	GT	Gauteng	GT485	13	
3	4608	GT	Gauteng	GT485	13	
4	354	GT	Gauteng	GT423	4	
...	
1009	7862	GT	Gauteng	TSH	103	
1010	3586	GT	Gauteng	TSH	104	
1011	7863	GT	Gauteng	TSH	104	
1012	3587	GT	Gauteng	TSH	105	
1013	7864	GT	Gauteng	TSH	105	

	WardID	LocalMunicipalityName	DistrictMunicipalityCode	\
0	74803012	Rand West City	DC48	
1	74803012	Rand West City	DC48	
2	74803013	Rand West City	DC48	
3	74803013	Rand West City	DC48	
4	74203004	Lesedi	DC42	
...	
1009	79900103	City of Tshwane	TSH	
1010	79900104	City of Tshwane	TSH	
1011	79900104	City of Tshwane	TSH	
1012	79900105	City of Tshwane	TSH	
1013	79900105	City of Tshwane	TSH	

	DistrictMunicipalityName	Year	Shape__Area	Shape__Length	\
0	West Rand	2011	6.661918e+06	13537.806926	
1	West Rand	2011	6.661918e+06	13537.806926	
2	West Rand	2011	4.788824e+05	4313.058347	
3	West Rand	2011	4.788824e+05	4313.058347	
4	Sedibeng	2011	9.282691e+05	4710.126049	
...	
1009	City of Tshwane	2011	6.245816e+07	47894.621520	
1010	City of Tshwane	2011	2.630179e+07	23861.425039	
1011	City of Tshwane	2011	2.630179e+07	23861.425039	
1012	City of Tshwane	2011	1.322401e+09	327991.053295	
1013	City of Tshwane	2011	1.322401e+09	327991.053295	

	geometry	\
0	POLYGON ((27.73847 -26.28225, 27.72304 -26.289...	
1	POLYGON ((27.73847 -26.28225, 27.72304 -26.289...	
2	POLYGON ((27.71000 -26.28295, 27.70973 -26.283...	
3	POLYGON ((27.71000 -26.28295, 27.70973 -26.283...	
4	POLYGON ((28.32999 -26.55073, 28.32880 -26.552...	
...	...	
1009	POLYGON ((28.71646 -25.68544, 28.71673 -25.685...	
1010	POLYGON ((28.76233 -25.66340, 28.75908 -25.677...	
1011	POLYGON ((28.76233 -25.66340, 28.75908 -25.677...	
1012	POLYGON ((29.08001 -25.49463, 29.09137 -25.530...	
1013	POLYGON ((29.08001 -25.49463, 29.09137 -25.530...	

	Percent African	Decimal
0	1.000000	
1	1.000000	
2	1.000000	
3	1.000000	
4	0.981132	
...	...	

```

1009          0.985294
1010          0.750000
1011          0.750000
1012          0.947368
1013          0.947368

```

```
[1014 rows x 14 columns]
```

It looks like the merge was successful!

Wards with the highest and lowest percentage of African respondents Now I want to identify the wards in which African population are highest and lowest

```
[14]: ward_high_afpop = pop_group_11_gdf[pop_group_11_gdf['Percent African_
↳Decimal']==1.0]
```

```
[15]: print ('There are ' + str(len(ward_high_afpop)) + ' wards in which the_
↳population of Africans is 100%')
```

There are 262 wards in which the population of Africans is 100%

We can see that there are over 250 wards in which the population of Africans is 100%. 262 is too long to create a table (I think respondents would be scrolling forever!) So I will create a table with ten values

```
[16]: highest_percent_afpop_2011 = pop_group_11_gdf.sort_values(by = "Percent African_
↳Decimal", ascending=False).head(10)
```

```
highest_percent_afpop_2011
```

```
[16]:
```

	OBJECTID	ProvinceCode	ProvinceName	LocalMunicipalityCode	WardNumber	\
0	330	GT	Gauteng	GT485	12	
271	7449	GT	Gauteng	EKU	57	
269	7448	GT	Gauteng	EKU	56	
268	3171	GT	Gauteng	EKU	56	
679	7697	GT	Gauteng	JHB	44	
680	3421	GT	Gauteng	JHB	45	
681	7698	GT	Gauteng	JHB	45	
684	3423	GT	Gauteng	JHB	47	
685	7700	GT	Gauteng	JHB	47	
686	3424	GT	Gauteng	JHB	48	

	WardID	LocalMunicipalityName	DistrictMunicipalityCode	\
0	74803012	Rand West City	DC48	
271	79700057	Ekurhuleni	EKU	
269	79700056	Ekurhuleni	EKU	
268	79700056	Ekurhuleni	EKU	
679	79800044	City of Johannesburg	JHB	

680	79800045	City of Johannesburg	JHB
681	79800045	City of Johannesburg	JHB
684	79800047	City of Johannesburg	JHB
685	79800047	City of Johannesburg	JHB
686	79800048	City of Johannesburg	JHB

	DistrictMunicipalityName	Year	Shape__Area	Shape__Length	\
0	West Rand	2011	6.661918e+06	13537.806926	
271	Ekurhuleni	2011	5.490537e+06	16690.678628	
269	Ekurhuleni	2011	3.050105e+06	8089.045752	
268	Ekurhuleni	2011	3.050105e+06	8089.045752	
679	City of Johannesburg	2011	7.720567e+06	15541.446187	
680	City of Johannesburg	2011	2.326851e+06	7956.596716	
681	City of Johannesburg	2011	2.326851e+06	7956.596716	
684	City of Johannesburg	2011	2.891492e+06	9748.138396	
685	City of Johannesburg	2011	2.891492e+06	9748.138396	
686	City of Johannesburg	2011	3.910968e+06	11637.406476	

	geometry	\
0	POLYGON ((27.73847 -26.28225, 27.72304 -26.289...	
271	POLYGON ((28.10968 -26.35560, 28.11993 -26.360...	
269	POLYGON ((28.13349 -26.34377, 28.13549 -26.344...	
268	POLYGON ((28.13349 -26.34377, 28.13549 -26.344...	
679	POLYGON ((27.86695 -26.18700, 27.86695 -26.187...	
680	POLYGON ((27.88383 -26.21438, 27.88383 -26.214...	
681	POLYGON ((27.88383 -26.21438, 27.88383 -26.214...	
684	POLYGON ((27.87186 -26.21974, 27.87194 -26.220...	
685	POLYGON ((27.87186 -26.21974, 27.87194 -26.220...	
686	POLYGON ((27.85848 -26.21739, 27.85862 -26.220...	

	Percent African Decimal
0	1.0
271	1.0
269	1.0
268	1.0
679	1.0
680	1.0
681	1.0
684	1.0
685	1.0
686	1.0

I want this table to be a supplement to the 2011 choropleth map so I will trim the columns to only the ones I believe will be helpful

[17]:

```
highest_percent_afpop_2011 = highest_percent_afpop_2011[['WardID',
↳'LocalMunicipalityName', 'DistrictMunicipalityName', 'Percent African',
↳Decimal']]
```

```
highest_percent_afpop_2011
```

```
[17]:      WardID LocalMunicipalityName DistrictMunicipalityName \
0      74803012      Rand West City      West Rand
271    79700057      Ekurhuleni      Ekurhuleni
269    79700056      Ekurhuleni      Ekurhuleni
268    79700056      Ekurhuleni      Ekurhuleni
679    79800044  City of Johannesburg  City of Johannesburg
680    79800045  City of Johannesburg  City of Johannesburg
681    79800045  City of Johannesburg  City of Johannesburg
684    79800047  City of Johannesburg  City of Johannesburg
685    79800047  City of Johannesburg  City of Johannesburg
686    79800048  City of Johannesburg  City of Johannesburg

      Percent African Decimal
0              1.0
271            1.0
269            1.0
268            1.0
679            1.0
680            1.0
681            1.0
684            1.0
685            1.0
686            1.0
```

While I think the table offers *some* useful information, it is important to note that there are over 250 wards in which the percentage of Africans is 100% and I am not sure how Python selects the values represented in the top 10.

I would now like to examine the 2011 wards in which the percentage of Africans is the lowest

```
[18]: ward_low_afpop_2011 = pop_group_11_gdf[pop_group_11_gdf['Percent African',
↳Decimal']==0]
```

```
[19]: print ('There are ' + str(len(ward_low_afpop_2011)) + ' wards in which the',
↳population of Africans is 0%')
```

There are 4 wards in which the population of Africans is 0%

I also want to create a table for this information.

```
[20]: lowest_percent_afpop_2011 = pop_group_11_gdf.sort_values(by = "Percent African",
↳Decimal", ascending=True).head(4)
```

```
lowest_percent_afpop_2011
```

```
[20]: OBJECTID ProvinceCode ProvinceName LocalMunicipalityCode WardNumber \
998      3580          GT      Gauteng          TSH          98
999      7857          GT      Gauteng          TSH          98
910      3536          GT      Gauteng          TSH          54
911      7813          GT      Gauteng          TSH          54

      WardID LocalMunicipalityName DistrictMunicipalityCode \
998 79900098      City of Tshwane          TSH
999 79900098      City of Tshwane          TSH
910 79900054      City of Tshwane          TSH
911 79900054      City of Tshwane          TSH

      DistrictMunicipalityName Year Shape__Area Shape__Length \
998      City of Tshwane 2011 2.861113e+07 28189.266995
999      City of Tshwane 2011 2.861113e+07 28189.266995
910      City of Tshwane 2011 1.293381e+07 19047.063955
911      City of Tshwane 2011 1.293381e+07 19047.063955

                                geometry \
998 POLYGON ((28.16604 -25.66844, 28.16592 -25.669...
999 POLYGON ((28.16604 -25.66844, 28.16592 -25.669...
910 POLYGON ((28.19831 -25.69105, 28.20185 -25.691...
911 POLYGON ((28.19831 -25.69105, 28.20185 -25.691...

      Percent African Decimal
998                      0.0
999                      0.0
910                      0.0
911                      0.0
```

```
[21]: lowest_percent_afpop_2011 = lowest_percent_afpop_2011[['WardID',
↳ 'LocalMunicipalityName', 'DistrictMunicipalityName', 'Percent African
↳ Decimal']]

lowest_percent_afpop_2011
```

```
[21]:      WardID LocalMunicipalityName DistrictMunicipalityName \
998 79900098      City of Tshwane          City of Tshwane
999 79900098      City of Tshwane          City of Tshwane
910 79900054      City of Tshwane          City of Tshwane
911 79900054      City of Tshwane          City of Tshwane

      Percent African Decimal
998                      0.0
999                      0.0
```

```
910                                0.0
911                                0.0
```

Interestingly, the wards in which African population is 0 are located in Tshwane

Choropleth maps Plotly accepts geojson information, so I will convert my data into geojson to create the choropleth map

```
[22]: pop_wards_2011 = pop_group_11_gdf.to_crs(epsg=4326) # convert the coordinate_
      ↪reference system to lat/long
      pop_wards_2011_json = pop_group_11_gdf.__geo_interface__ #convert to geoJSON
```

The plotly code requires additional library imports so I will bring those in

```
[23]: from mpl_toolkits.axes_grid1 import make_axes_locatable
      import plotly.graph_objects as go
```

```
[24]: zmin = pop_wards_2011['Percent African Decimal'].min() #set the map min and max
      zmax = pop_wards_2011['Percent African Decimal'].max()

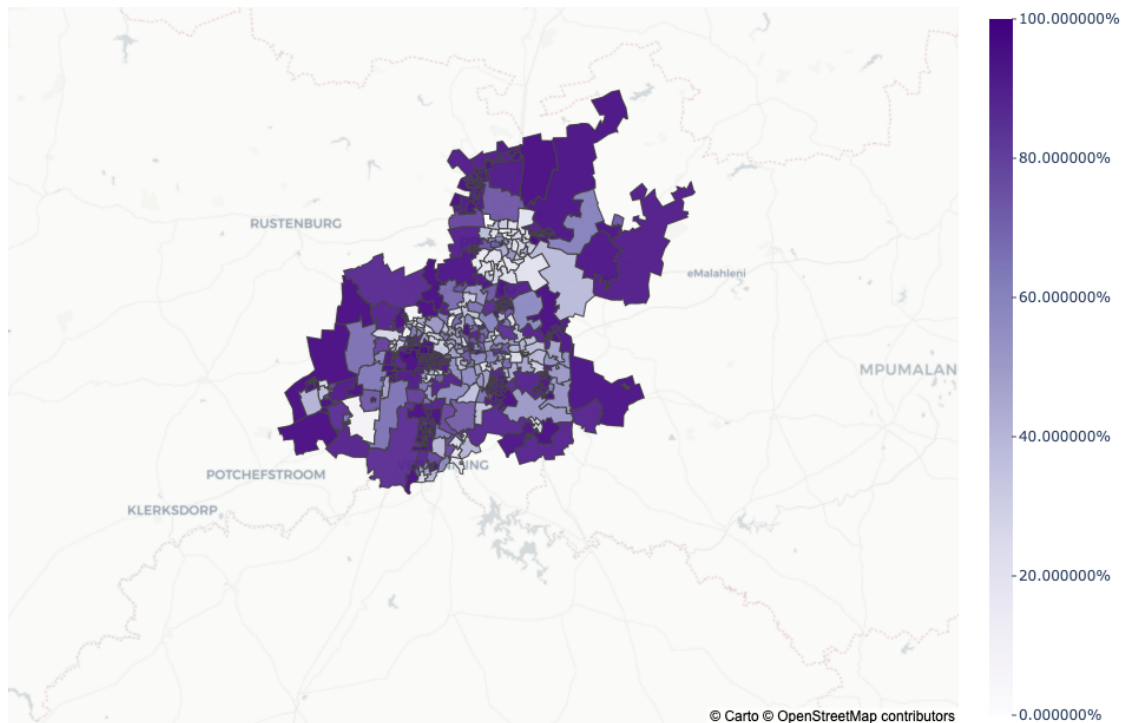
      # Set the data for the map
      data = go.Choroplethmapbox(
          geojson = pop_wards_2011_json,
          locations = pop_wards_2011.index,
          z = pop_wards_2011['Percent African Decimal'],
          text = pop_wards_2011.WardID,
          colorbar=dict(thickness=20, ticklen=3, tickformat='%',outlinewidth=0),
          marker_line_width=1, marker_opacity=0.7, colorscale="Purples",
          zmin=zmin, zmax=zmax,
          hovertemplate = "<b>{%text}</b><br>" +
                          "%{z:.1%}<br>" +
                          "<extra></extra>")

      # Set the layout for the map
      layout = go.Layout(
          title = {'text': f"Percent of Africans in Each Ward, 2011",
                  'font': {'size':24}},
          mapbox1 = dict(
              domain = {'x': [0, 1], 'y': [0, 1]},
              center = dict(lat=-26.270760, lon=28.112268),
              zoom = 7),
          autosize=True,
          height=650,
          margin=dict(l=0, r=0, t=40, b=0))

      fig=go.Figure(data=data, layout=layout)
```

```
fig.update_layout(mapbox_style="carto-positron")
fig.show()
#fig.write_html("purple_Africans_ward_2011.html") #I made this code into a
→comment so it wouldn't keep updating the html file
```

Percent of Africans in Each Ward, 2011



This interactive map shows us the percentage of Africans in each ward. Viewers can zoom into each ward and see the Ward ID number along with the percentage of Africans in each ward. This will be incredibly helpful when we need to identify the wards that the BRT lines pass through. In looking at this map, we can also see that the wards on the periphery of the region have some of the highest percentages of Africans while those more inland, closer to cities like Johannesburg and Pretoria have lower percentages of Africans

1.4.2 2017

Now that we have the interactive map for 2011, I will create one for 2017 and examine the wards in which the percentage of Africans is highest and lowest

```
[25]: pop_group_17 = pd.crosstab(index=gdf_2017['WardID'],
                                columns=gdf_2017['A1_Pop_group'],
                                margins=True,
                                margins_name='Total Number')

pop_group_17
```

```
[25]: A1_Pop_group  African  Coloured  Indian/Asian  Other  White  Total Number
WardID
74201001          11          2          0          0          23          36
74201002          36          1          0          0          0          37
74201003          28          2          0          0          4          34
74201004          24          0          0          0          10         34
74201005          13          1          0          0          17          31
...
79900104          36          1          0          0          0          37
79900105          27          1          0          0          3          31
79900106          37          0          0          1          0          38
79900107          47          0          0          0          0          47
Total Number    20964          892          370          85        2578        24889
```

[530 rows x 6 columns]

Just as I did for the 2011 data, I will create normalized data for the African population group in 2017.

```
[26]: pop_group_17['Percent African Decimal'] = pop_group_17['African']/
      ↪pop_group_17['Total Number']
pop_group_17['Percent African'] = pop_group_17['African']/pop_group_17['Total_
      ↪Number']*100

pop_group_17
```

```
[26]: A1_Pop_group  African  Coloured  Indian/Asian  Other  White  Total Number  \
WardID
74201001          11          2          0          0          23          36
74201002          36          1          0          0          0          37
74201003          28          2          0          0          4          34
74201004          24          0          0          0          10         34
74201005          13          1          0          0          17          31
...
79900104          36          1          0          0          0          37
79900105          27          1          0          0          3          31
79900106          37          0          0          1          0          38
79900107          47          0          0          0          0          47
Total Number    20964          892          370          85        2578        24889
```

```
A1_Pop_group  Percent African Decimal  Percent African
WardID
74201001          0.305556          30.555556
74201002          0.972973          97.297297
74201003          0.823529          82.352941
74201004          0.705882          70.588235
74201005          0.419355          41.935484
```

...
79900104	0.972973	97.297297
79900105	0.870968	87.096774
79900106	0.973684	97.368421
79900107	1.000000	100.000000
Total Number	0.842300	84.229981

[530 rows x 8 columns]

```
[27]: pop_group_17 = pop_group_17.drop(['Total Number'])
```

I'm going to save the 2017 dataframe with just the newly created column

```
[28]: pop_group_17=pop_group_17[['Percent African Decimal', 'Percent African']]

pop_group_17
```

```
[28]: A1_Pop_group  Percent African Decimal  Percent African
WardID
74201001          0.305556          30.555556
74201002          0.972973          97.297297
74201003          0.823529          82.352941
74201004          0.705882          70.588235
74201005          0.419355          41.935484
...
79900103          1.000000          100.000000
79900104          0.972973          97.297297
79900105          0.870968          87.096774
79900106          0.973684          97.368421
79900107          1.000000          100.000000
```

[529 rows x 2 columns]

I'm going to bring in the 2017 ward geospatial data...

```
[29]: wards_2017 = gpd.read_file('MDB_Wards_2016.geojson')
```

...and merge it with the dataframe!

```
[30]: pop_group_17_gdf = wards_2017.merge(pop_group_17, on='WardID')

pop_group_17_gdf
```

```
[30]:   OBJECTID ProvinceCode ProvinceName LocalMunicipalityCode WardNumber \
0      1504          GT      Gauteng          EKU          60
1      1505          GT      Gauteng          EKU          61
2      1506          GT      Gauteng          EKU          62
3      1507          GT      Gauteng          EKU          63
```

4	1508	GT	Gauteng	EKU	64
..
524	4319	GT	Gauteng	EKU	108
525	4320	GT	Gauteng	EKU	109
526	4321	GT	Gauteng	EKU	110
527	4322	GT	Gauteng	EKU	111
528	4323	GT	Gauteng	EKU	112

	WardID	LocalMunicipalityName	DistrictMunicipalityCode	\
0	79700060	Ekurhuleni	EKU	
1	79700061	Ekurhuleni	EKU	
2	79700062	Ekurhuleni	EKU	
3	79700063	Ekurhuleni	EKU	
4	79700064	Ekurhuleni	EKU	
..	
524	79700108	Ekurhuleni	EKU	
525	79700109	Ekurhuleni	EKU	
526	79700110	Ekurhuleni	EKU	
527	79700111	Ekurhuleni	EKU	
528	79700112	Ekurhuleni	EKU	

	DistrictMunicipalityName	Year	Shape__Area	Shape__Length	\
0	Ekurhuleni	2016	3.898607e+06	11472.007890	
1	Ekurhuleni	2016	2.040419e+07	24911.970404	
2	Ekurhuleni	2016	5.823267e+07	42030.549363	
3	Ekurhuleni	2016	3.332302e+06	10253.147328	
4	Ekurhuleni	2016	1.239711e+07	18521.992239	
..	
524	Ekurhuleni	2016	3.986026e+06	12293.042422	
525	Ekurhuleni	2016	7.126183e+06	17501.893757	
526	Ekurhuleni	2016	8.281005e+06	15886.916039	
527	Ekurhuleni	2016	2.570824e+07	29656.849051	
528	Ekurhuleni	2016	6.723852e+06	14259.004191	

	geometry	\
0	POLYGON ((28.16493 -26.35885, 28.16632 -26.358...	
1	POLYGON ((28.13552 -26.40783, 28.13665 -26.408...	
2	POLYGON ((28.24732 -26.38152, 28.25052 -26.386...	
3	POLYGON ((28.16205 -26.37103, 28.16219 -26.371...	
4	POLYGON ((28.22547 -26.35844, 28.22591 -26.358...	
..	...	
524	POLYGON ((28.19092 -26.37677, 28.19174 -26.377...	
525	POLYGON ((28.43689 -26.10593, 28.43703 -26.106...	
526	POLYGON ((28.39511 -26.14457, 28.39570 -26.145...	
527	POLYGON ((28.43582 -26.31314, 28.43659 -26.313...	
528	POLYGON ((28.35914 -26.35703, 28.35931 -26.358...	

	Percent African Decimal	Percent African
0	1.000000	100.000000
1	1.000000	100.000000
2	1.000000	100.000000
3	1.000000	100.000000
4	1.000000	100.000000
..
524	1.000000	100.000000
525	1.000000	100.000000
526	1.000000	100.000000
527	0.884615	88.461538
528	1.000000	100.000000

[529 rows x 15 columns]

Population Groups Now I want to find the wards in which African population is highest

```
[31]: ward_high_afpop_2017 = pop_group_17_gdf[pop_group_17_gdf['Percent African_
↳Decimal']==1.0]
```

```
[32]: print ('There are ' + str(len(ward_high_afpop_2017)) + ' wards in which the_
↳population of Africans is 100%')
```

There are 235 wards in which the population of Africans is 100%

Similar to 2011, in 2017 there are over 200 wards in which the population of Africans is 100%, however there are less wards in which the percentage of Africans is 100%, which could be an indication that the wards are becoming less segregated

```
[33]: highest_percent_afpop_2017 = pop_group_17_gdf.sort_values(by = "Percent African_
↳Decimal", ascending=False).head(10)
```

highest_percent_afpop_2017

```
[33]:
```

	OBJECTID	ProvinceCode	ProvinceName	LocalMunicipalityCode	WardNumber	\
0	1504	GT	Gauteng	EKU	60	
311	3842	GT	Gauteng	GT481	30	
307	3777	GT	Gauteng	GT485	35	
305	3775	GT	Gauteng	GT485	33	
304	3774	GT	Gauteng	GT485	32	
303	3773	GT	Gauteng	GT485	31	
301	3771	GT	Gauteng	GT485	29	
300	3770	GT	Gauteng	GT485	28	
299	3769	GT	Gauteng	GT485	27	
294	3764	GT	Gauteng	GT485	22	

WardID LocalMunicipalityName DistrictMunicipalityCode \

0	79700060	Ekurhuleni	EKU
311	74801030	Mogale City	DC48
307	74205035	Rand West City	DC48
305	74205033	Rand West City	DC48
304	74205032	Rand West City	DC48
303	74205031	Rand West City	DC48
301	74205029	Rand West City	DC48
300	74205028	Rand West City	DC48
299	74205027	Rand West City	DC48
294	74205022	Rand West City	DC48

	DistrictMunicipalityName	Year	Shape__Area	Shape__Length \
0	Ekurhuleni	2016	3.898607e+06	11472.007890
311	West Rand	2016	1.147148e+08	57707.915574
307	West Rand	2016	1.223508e+07	21544.061485
305	West Rand	2016	2.462735e+07	26282.347178
304	West Rand	2016	4.957059e+06	12251.483088
303	West Rand	2016	7.100345e+05	5437.014974
301	West Rand	2016	1.789664e+07	20410.847279
300	West Rand	2016	8.175189e+06	17925.272925
299	West Rand	2016	4.002420e+06	8976.487478
294	West Rand	2016	5.653726e+05	3434.717719

	geometry \
0	POLYGON ((28.16493 -26.35885, 28.16632 -26.358...
311	POLYGON ((27.70048 -26.06088, 27.70052 -26.061...
307	POLYGON ((27.70385 -26.28741, 27.70384 -26.288...
305	POLYGON ((27.72198 -26.23594, 27.72473 -26.238...
304	POLYGON ((27.74465 -26.27949, 27.73847 -26.282...
303	POLYGON ((27.70151 -26.27890, 27.69996 -26.279...
301	POLYGON ((27.70275 -26.27468, 27.70279 -26.275...
300	POLYGON ((27.70131 -26.30890, 27.70163 -26.309...
299	POLYGON ((27.68395 -26.31075, 27.68401 -26.311...
294	POLYGON ((27.70304 -26.22855, 27.70300 -26.228...

	Percent African Decimal	Percent African
0	1.0	100.0
311	1.0	100.0
307	1.0	100.0
305	1.0	100.0
304	1.0	100.0
303	1.0	100.0
301	1.0	100.0
300	1.0	100.0
299	1.0	100.0
294	1.0	100.0

```
[34]: highest_percent_afpop_2017 = highest_percent_afpop_2017[['WardID',
↳ 'LocalMunicipalityName', 'DistrictMunicipalityName', 'Percent African',
↳ 'Decimal']]
```

```
highest_percent_afpop_2017
```

```
[34]:      WardID LocalMunicipalityName DistrictMunicipalityName \
0      79700060      Ekurhuleni      Ekurhuleni
311    74801030      Mogale City      West Rand
307    74205035      Rand West City      West Rand
305    74205033      Rand West City      West Rand
304    74205032      Rand West City      West Rand
303    74205031      Rand West City      West Rand
301    74205029      Rand West City      West Rand
300    74205028      Rand West City      West Rand
299    74205027      Rand West City      West Rand
294    74205022      Rand West City      West Rand

      Percent African Decimal
0              1.0
311            1.0
307            1.0
305            1.0
304            1.0
303            1.0
301            1.0
300            1.0
299            1.0
294            1.0
```

I now want to see the wards in which the percentage of Africans is lowest, to compare it to 2011

```
[35]: ward_low_afpop_2017 = pop_group_17_gdf[pop_group_17_gdf['Percent African',
↳ 'Decimal']==0]
```

```
[36]: print ('There are ' + str(len(ward_low_afpop_2017)) + ' wards in which the',
↳ 'population of Africans is 0%')
```

There are 0 wards in which the population of Africans is 0%

In 2017, there are 0 wards in which the 100% of the ward identifies as African. I would still like to see the wards in which the percentage of Africans is lowest

```
[37]: lowest_percent_afpop_2017 = pop_group_17_gdf.sort_values(by = "Percent African",
↳ 'Decimal', ascending=True).head(5)
```

```
lowest_percent_afpop_2017
```

```
[37]: OBJECTID ProvinceCode ProvinceName LocalMunicipalityCode WardNumber \
48      2557          GT      Gauteng          JHB          18
112     2621          GT      Gauteng          JHB          82
39      2548          GT      Gauteng          JHB          9
421     4216          GT      Gauteng          GT422         2
485     4280          GT      Gauteng          EKU          38
```

```
      WardID LocalMunicipalityName DistrictMunicipalityCode \
48  79800018 City of Johannesburg          JHB
112 79800082 City of Johannesburg          JHB
39   79800009 City of Johannesburg          JHB
421  74202002          Midvaal          DC42
485  79700038          Ekurhuleni          EKU
```

```
      DistrictMunicipalityName Year Shape__Area Shape__Length \
48      City of Johannesburg  2016  7.217249e+06  12634.036845
112      City of Johannesburg  2016  7.864120e+06  12534.144259
39      City of Johannesburg  2016  1.614611e+07  27376.989897
421          Sedibeng        2016  2.147480e+07  25909.990027
485          Ekurhuleni      2016  1.512421e+07  20417.571665
```

```
                                geometry \
48  POLYGON ((27.91502 -26.28289, 27.91530 -26.282...
112 POLYGON ((27.96188 -26.17000, 27.96226 -26.170...
39  POLYGON ((27.85160 -26.29897, 27.85514 -26.300...
421 POLYGON ((27.99001 -26.60370, 27.99023 -26.604...
485 POLYGON ((28.11100 -26.33053, 28.10697 -26.338...
```

```
      Percent African Decimal Percent African
48          0.158730      15.873016
112         0.196970      19.696970
39         0.203390      20.338983
421         0.212121      21.212121
485         0.222222      22.222222
```

```
[38]: lowest_percent_afpop_2017 = lowest_percent_afpop_2017[['WardID',
↳ 'LocalMunicipalityName', 'DistrictMunicipalityName', 'Percent African',
↳ 'Decimal']]

lowest_percent_afpop_2017
```

```
[38]:      WardID LocalMunicipalityName DistrictMunicipalityName \
48  79800018 City of Johannesburg      City of Johannesburg
112 79800082 City of Johannesburg      City of Johannesburg
39   79800009 City of Johannesburg      City of Johannesburg
421  74202002          Midvaal          Sedibeng
485  79700038          Ekurhuleni          Ekurhuleni
```

	Percent African	Decimal
48		0.158730
112		0.196970
39		0.203390
421		0.212121
485		0.222222

Interestingly, the lowest percentage of Africans in a ward is now approximately 15%, a (seemingly) significant difference from 2011

Choropleth Maps I'm going to now convert this geodataframe into a geojson

```
[39]: pop_wards_2017 = pop_group_17_gdf.to_crs(epsg=4326) # convert the coordinate_
      ↪reference system to lat/long
      pop_wards_2017_json = pop_group_17_gdf.__geo_interface__ #convert to geoJSON
```

```
[40]: zmin = pop_wards_2017['Percent African Decimal'].min()
      zmax = pop_wards_2017['Percent African Decimal'].max()

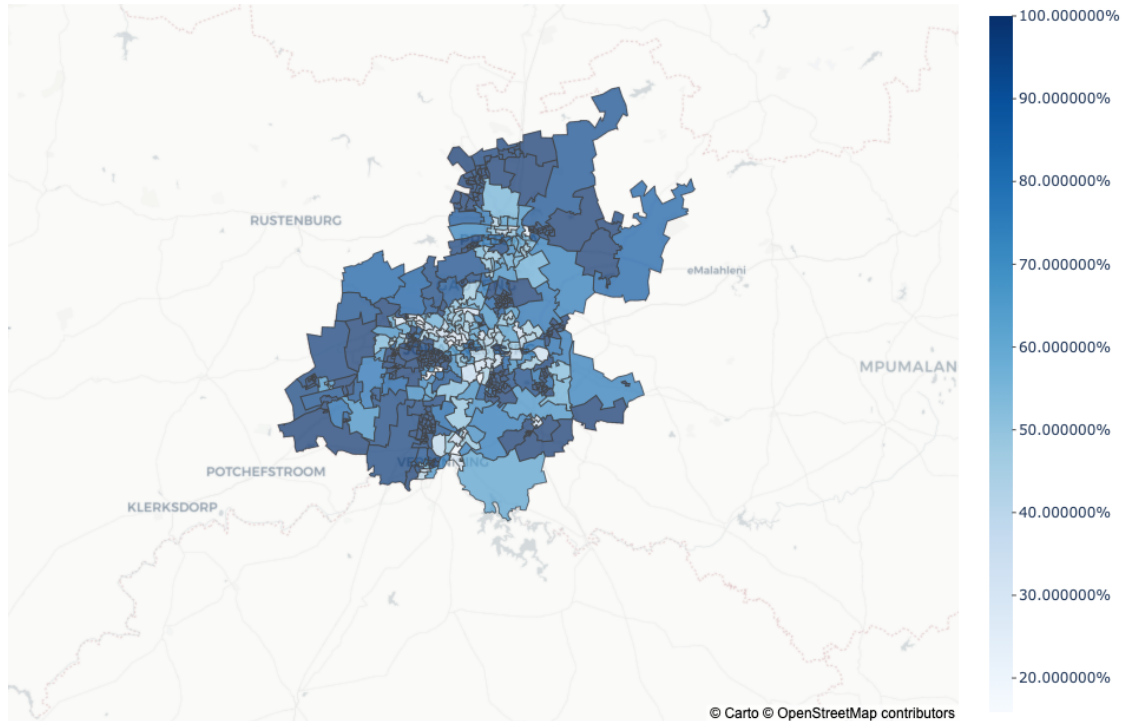
      data = go.Choroplethmapbox(
          geojson = pop_wards_2017_json,
          locations = pop_wards_2017.index,
          z = pop_wards_2017['Percent African Decimal'],
          text = pop_wards_2017.WardID,
          colorbar=dict(thickness=20, ticklen=3, tickformat='%',outlinewidth=0),
          marker_line_width=1, marker_opacity=0.7, colorscale="Blues",
          zmin=zmin, zmax=zmax,
          hovertemplate = "<b>{%text}</b><br>" +
                          "%{z:.1%}<br>" +
                          "<extra></extra>"

      # Set the layout for the map
      layout = go.Layout(
          title = {'text': f"Percent of Africans in Each Ward, 2017",
                  'font': {'size':24}},
          mapbox1 = dict(
              domain = {'x': [0, 1], 'y': [0, 1]},
              center = dict(lat=-26.270760, lon=28.112268),
              zoom = 7),
          autosize=True,
          height=650,
          margin=dict(l=0, r=0, t=40, b=0))

      # Generate the map
      fig=go.Figure(data=data, layout=layout)
      fig.update_layout(mapbox_style="carto-positron")
```

```
fig.show()
#fig.write_html("Africans_ward_2017.html") #Again, I saved this as a comment to
↳ prevent continuous updates of the HTML
```

Percent of Africans in Each Ward, 2017



Now we have an interactive choropleth map for the percentage of African respondents in the GCRO 2017 data. We will use this in our final analysis

1.5 BRT Data

Because we are examining the impact of BRT development and use on social attitudes, I want to identify both the wards in which BRT use is highest and the wards through which the BRT passes

1.5.1 BRT Lines

We received BRT line data from the GCRO which I'd like to explore and then overlay on the 2017 interactive choropleth map.

```
[41]: BRT_line.info()
```

```
<class 'geopandas.geodataframe.GeoDataFrame'>
RangeIndex: 11 entries, 0 to 10
Data columns (total 49 columns):
#   Column          Non-Null Count  Dtype
---
```

0	Name	11 non-null	object
1	Shape_Leng	11 non-null	float64
2	ID	11 non-null	int64
3	LENGTH	11 non-null	float64
4	DIR	11 non-null	int64
5	TYPE	8 non-null	object
6	BRT1	11 non-null	int64
7	BRT2	11 non-null	int64
8	BRT3	11 non-null	int64
9	BRT4	11 non-null	int64
10	BRT5	11 non-null	int64
11	BRT6	11 non-null	int64
12	BRT7	11 non-null	int64
13	BRT8	11 non-null	int64
14	BRT9	11 non-null	int64
15	BRT10	11 non-null	int64
16	BRT11	11 non-null	int64
17	BRT12	11 non-null	int64
18	BRT13	11 non-null	int64
19	BRT14	11 non-null	int64
20	BRT15	11 non-null	int64
21	BRT16	11 non-null	int64
22	BRT17	11 non-null	int64
23	BRT18	11 non-null	int64
24	BRT19	11 non-null	int64
25	BRT20	11 non-null	int64
26	BRT21	11 non-null	int64
27	BRT22	11 non-null	int64
28	LENGTH_F	11 non-null	float64
29	Year	6 non-null	object
30	Shape_Le_1	11 non-null	float64
31	ROUTE_ID	11 non-null	int64
32	ROUTE_NAME	2 non-null	object
33	DIRECTION	2 non-null	object
34	ROUTELENGT	11 non-null	float64
35	PHASE_1A	11 non-null	float64
36	PHASE_1B	11 non-null	float64
37	PHASE_1C	11 non-null	float64
38	PHASE_2	11 non-null	float64
39	PHASE_3	11 non-null	float64
40	PHASE_4	11 non-null	float64
41	PHASE_5	11 non-null	float64
42	PHASE_6	11 non-null	float64
43	STATUS	2 non-null	object
44	NETWORK_EV	2 non-null	object
45	ORIGIN	0 non-null	object
46	DESTINATIO	0 non-null	object
47	AVESPEED	0 non-null	object

```

48 geometry      11 non-null      geometry
dtypes: float64(13), geometry(1), int64(25), object(10)
memory usage: 4.3+ KB

```

There are a lot of columns but only 11 rows! Let's take a closer look

```
[42]: BRT_line.head(5)
```

```

[42]:   Name  Shape_Leng  ID  LENGTH  DIR  TYPE  BRT1  BRT2  BRT3  BRT4  ...  \
0  TSH    0.025628    0    0.00    0  None    0    0    0    0  ...
1  TSH    0.059222    6    6.50    0   BRT    1    0    0    0  ...
2  TSH    0.036972   23    5.14    0   BRT    1    0    0    0  ...
3  TSH    0.007126   21    0.79    0   BRT    1    0    0    0  ...
4  TSH    0.011501    4    1.25    0   BRT    1    0    0    0  ...

      PHASE_3  PHASE_4  PHASE_5  PHASE_6  STATUS  NETWORK_EV  ORIGIN  DESTINATIO  \
0         0.0      0.0      0.0      0.0   None         None   None      None
1         0.0      0.0      0.0      0.0   None         None   None      None
2         0.0      0.0      0.0      0.0   None         None   None      None
3         0.0      0.0      0.0      0.0   None         None   None      None
4         0.0      0.0      0.0      0.0   None         None   None      None

      AVESPEED      geometry
0      None  LINESTRING (28.22520 -25.75598, 28.22529 -25.7...
1      None  LINESTRING (28.18702 -25.68882, 28.18740 -25.6...
2      None  LINESTRING (28.19085 -25.75045, 28.19212 -25.7...
3      None  LINESTRING (28.19029 -25.74334, 28.19085 -25.7...
4      None  LINESTRING (28.18304 -25.67811, 28.18342 -25.6...

```

[5 rows x 49 columns]

Trim the data I want to trim this data to keep only the columns I will use

```
[43]: BRT_line = BRT_line[['Name', 'Shape_Leng', 'ID', 'geometry']].copy()
```

```
[44]: BRT_line
```

```

[44]:   Name  Shape_Leng  ID      geometry
0  TSH    0.025628    0  LINESTRING (28.22520 -25.75598, 28.22529 -25.7...
1  TSH    0.059222    6  LINESTRING (28.18702 -25.68882, 28.18740 -25.6...
2  TSH    0.036972   23  LINESTRING (28.19085 -25.75045, 28.19212 -25.7...
3  TSH    0.007126   21  LINESTRING (28.19029 -25.74334, 28.19085 -25.7...
4  TSH    0.011501    4  LINESTRING (28.18304 -25.67811, 28.18342 -25.6...
5  ECU    0.000000   67  LINESTRING (28.19112 -26.08927, 28.19046 -26.0...
6  ECU    0.000000   82  LINESTRING (28.19185 -26.09092, 28.19209 -26.0...
7  ECU    0.000000    0  LINESTRING (28.19112 -26.08927, 28.19299 -26.0...
8  ECU    0.000000    0  LINESTRING (28.20917 -26.14441, 28.20929 -26.1...
9  JHB    0.000000    0  LINESTRING (27.87592 -26.26884, 27.87695 -26.2...

```

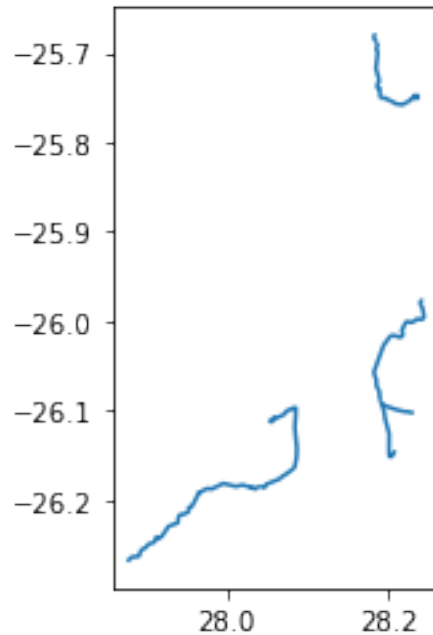


```
10 JHB      0.000000    0 LINESTRING (28.05931 -26.10887, 28.05923 -26.1...
```

Now that the data is trimmed, let's see what it looks like plotted

```
[45]: BRT_line.plot()
```

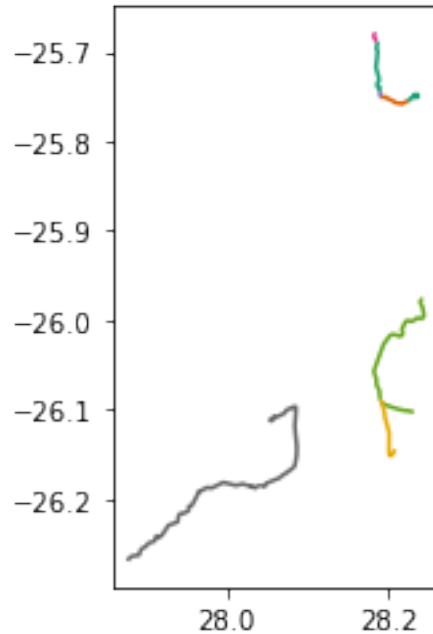
```
[45]: <AxesSubplot:>
```



I know that there are three BRT systems in the Gauteng region and we can see three distinct lines here! Let's look at them in different colors

```
[46]: BRT_line.plot(  
      cmap='Dark2',  
      markersize=10,  
      legend=True)
```

```
[46]: <AxesSubplot:>
```



By changing the color scheme, I can see that the BRT in Tshwane and Ekurhuleni have different routes. I want to be able to overlay the BRT lines on both my static and interactive maps. I have been successful in overlaying the lines on static maps but now want to overlay them on my plotly interactive maps

1.5.2 Converting BRT line geometry to lat/lon

I read on stackoverflow that in order to plot the BRT line data on the choropleth map, I need to convert the geometry linestring to lat/lon.

```
[47]: import numpy as np

import shapely.geometry
```

```
[48]: lats = []
lons = []
names = []

for feature in BRT_line.geometry:
    if isinstance(feature, shapely.geometry.linestring.LineString):
        linestrings = [feature]
    else:
        continue
    for linestring in linestrings:
        x, y = linestring.xy
        lats = np.append(lats, y)
```

```
lons = np.append(lons, x)
lats = np.append(lats, None)
lons = np.append(lons, None)
```

Since this is new code, I want to see if and how the data has changed

```
[49]: BRT_line
```

```
[49]:
```

	Name	Shape_Leng	ID	geometry
0	TSH	0.025628	0	LINESTRING (28.22520 -25.75598, 28.22529 -25.7...
1	TSH	0.059222	6	LINESTRING (28.18702 -25.68882, 28.18740 -25.6...
2	TSH	0.036972	23	LINESTRING (28.19085 -25.75045, 28.19212 -25.7...
3	TSH	0.007126	21	LINESTRING (28.19029 -25.74334, 28.19085 -25.7...
4	TSH	0.011501	4	LINESTRING (28.18304 -25.67811, 28.18342 -25.6...
5	EKU	0.000000	67	LINESTRING (28.19112 -26.08927, 28.19046 -26.0...
6	EKU	0.000000	82	LINESTRING (28.19185 -26.09092, 28.19209 -26.0...
7	EKU	0.000000	0	LINESTRING (28.19112 -26.08927, 28.19299 -26.0...
8	EKU	0.000000	0	LINESTRING (28.20917 -26.14441, 28.20929 -26.1...
9	JHB	0.000000	0	LINESTRING (27.87592 -26.26884, 27.87695 -26.2...
10	JHB	0.000000	0	LINESTRING (28.05931 -26.10887, 28.05923 -26.1...

The dataframe looks the same...so I'd like to plot it again, this time using Plotly to see if the code was successful

```
[50]: fig = px.line_geo(BRT_line, lat=lats, lon=lons)
fig.update_geos(fitbounds="locations", visible=False)
fig.update_layout(margin={"r":0,"t":0,"l":0,"b":0})
fig.show()
```



Yay! The data is now interactive and when you hover over the lines, you can see the lat/lon coordinates for the entire line.

1.5.3 BRT Stations

Now that we have the interactive BRT lines, I want to create an interactive map of the BRT stations. I handcoded the BRT stations outside of python (using geojson.io) based off of the BRT station names for [Rea Vaya](#), [A Re Yang](#), and [Harambee](#). I tried my best to map them accurately, however they should be considered approximate bus station locations!

```
[51]: BRT_stations.head(5)
```

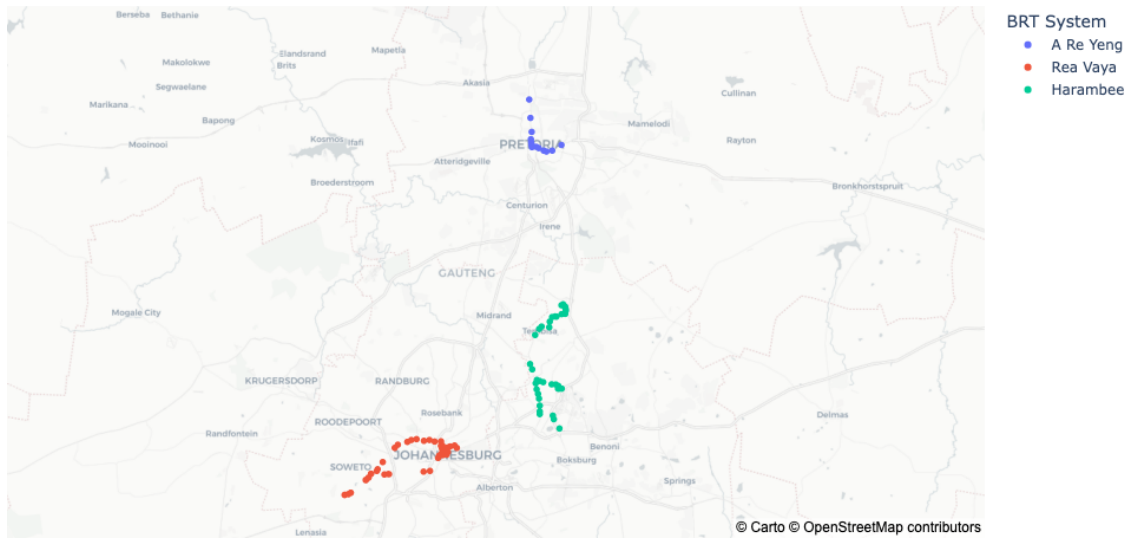
```
[51]:  marker-color marker-size marker-symbol      BRT Route Name \
0      #f28ce5      small      bus  A Re Yeng      TSH T1
1      #f28ce5      small      bus  A Re Yeng      TSH T1
2      #f28ce5      small      bus  A Re Yeng      TSH T1
3      #f28ce5      small      bus  A Re Yeng      TSH T1
4      #f28ce5      small      bus  A Re Yeng      TSH T1
```

```
      Stop Name      geometry
0  Church Square  POINT (28.18863 -25.74705)
1  Central Station  POINT (28.18907 -25.75059)
2      Nana Sita  POINT (28.19626 -25.75013)
3  Ruth Mompoti  POINT (28.20015 -25.75201)
4  Mahatma Gandhi  POINT (28.20840 -25.75565)
```

```
[52]: fig = px.scatter_mapbox(BRT_stations,
                             lat=BRT_stations.geometry.y,
                             lon=BRT_stations.geometry.x,
                             color="BRT",
                             hover_name="Stop Name",
                             mapbox_style="carto-positron",
                             title="BRT Stations in Gauteng",
                             labels={"BRT": "BRT System"},
                             center = {"lat":-25.935266, "lon": 28.1296512},
                             zoom=8.5)

fig.update_layout(height=500, margin={"r":0,"t":50,"l":0,"b":0})
fig.show()
fig.write_html("BRT_stations_2017.html")
```

BRT Stations in Gauteng



I was having difficulty overlaying the station maps on a choropleth map, and ultimately was not successful in overlaying them so we will be using the interactive BRT lines for interactive maps and will use this interactive map for readers interested in exploring the different lines and their respective stations

1.6 Interactive Choropleth maps with BRT lines

We want to examine racial attitudes in wards that the BRT passes through and in which BRT use is highest. To do so, we need to create interactive maps because they allow us to see each ward's ID number.

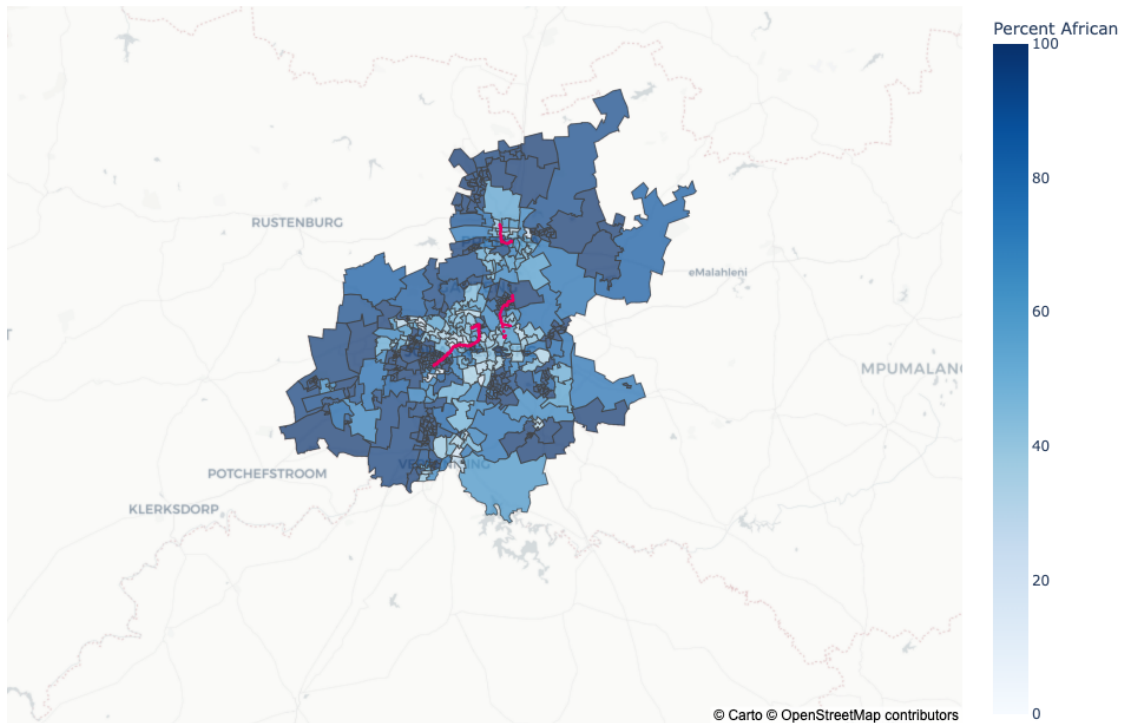
Note: For some reason, I kept getting continuous errors when running the following code. After making seemingly a million tweaks, it finally worked!

1.6.1 Population of Africans with BRT lines overlayed

```
[53]: fig = px.choropleth_mapbox(pop_wards_2017, geojson=pop_wards_2017_json,
    ↪ locations=pop_wards_2017.index, color='Percent African',
    color_continuous_scale="Blues",
    range_color=(0, 100),
    hover_name= pop_wards_2017['WardID'],
    mapbox_style="carto-positron",
    zoom=7, center = {"lat":-26.270760, "lon": 28.
    ↪ 112268},
    title="Percent of Africans in each ward with BRT",
    ↪ lines(red)",
    opacity=0.7,
    )
```

```
fig.add_scattermapbox(
    lat = lats,
    lon = lons,
    marker_size=3,
    marker_color='rgb(235, 0, 100)'
)
fig.update_layout(height=650, margin={"r":0,"t":40,"l":0,"b":0})
fig.show()
#fig.write_html("BRT Ward.html")
```

Percent of Africans in each ward with BRT lines(red)



We can now see the wards that the BRT line passes through, which will be useful for our final analysis. I manually identified the wards that have the BRT lines running through them and tracking it outside of Python to then use in our analysis

1.6.2 BRT Use

I want to create a choropleth map of BRT use in the region and then overlay the BRT line. We hypothesize that BRT use will be higher in wards that have the BRT passing through

```
[54]: BRT_use = pd.crosstab(index=gdf_2017['WardID'],
                           columns=gdf_2017['Q5_11_BRT_Freq'],
                           margins=True,
                           margins_name='Total Number')
```

BRT_use

```
[54]: Q5_11_BRT_Freq  A few times a month  Every few months  Most days  Never  \
WardID
74201001              0              0              0      35
74201002              2              1              1      33
74201003              1              0              0      32
74201004              0              0              0      33
74201005              0              0              0      30
...
79900104              0              1              1      29
79900105              0              0              0      30
79900106              2              5              0      30
79900107              3              2              7      30
Total Number          887          836          460  21400
```

```
Q5_11_BRT_Freq  Once or twice a week  Total Number
WardID
74201001              0              35
74201002              0              37
74201003              0              33
74201004              1              34
74201005              0              30
...
79900104              1              32
79900105              0              30
79900106              0              37
79900107              2              44
Total Number          341          23924
```

[530 rows x 6 columns]

Though respondents selected the frequency of their BRT use from a list of 5 options, we will be analyzing BRT use at any frequency, given that BRT use is relatively small compared to other modes of transportation. I want to create a new column of total BRT use by adding up the number of respondents that use BRT in each ward

```
[55]: BRT_use['Use BRT'] = BRT_use['A few times a month']+BRT_use['Every few_
      ↪months']+BRT_use['Most days']+BRT_use['Once or twice a week']
```

BRT_use

```
[55]: Q5_11_BRT_Freq  A few times a month  Every few months  Most days  Never  \
WardID
74201001              0              0              0      35
74201002              2              1              1      33
```

74201003	1	0	0	32
74201004	0	0	0	33
74201005	0	0	0	30
...
79900104	0	1	1	29
79900105	0	0	0	30
79900106	2	5	0	30
79900107	3	2	7	30
Total Number	887	836	460	21400

Q5_11_BRT_Freq	Once or twice a week	Total Number	Use BRT
WardID			
74201001	0	35	0
74201002	0	37	4
74201003	0	33	1
74201004	1	34	1
74201005	0	30	0
...
79900104	1	32	3
79900105	0	30	0
79900106	0	37	7
79900107	2	44	14
Total Number	341	23924	2524

[530 rows x 7 columns]

Now that I have total BRT use summed, I want to create a proportion and percentage column

```
[56]: BRT_use['YES Use BRT']=BRT_use['Use BRT']/BRT_use['Total Number']
```

BRT_use

```
[56]: Q5_11_BRT_Freq  A few times a month  Every few months  Most days  Never  \
WardID
74201001            0                0                0        35
74201002            2                1                1        33
74201003            1                0                0        32
74201004            0                0                0        33
74201005            0                0                0        30
...
79900104            0                1                1        29
79900105            0                0                0        30
79900106            2                5                0        30
79900107            3                2                7        30
Total Number        887                836                460  21400
```

Q5_11_BRT_Freq	Once or twice a week	Total Number	Use BRT	YES Use BRT
----------------	----------------------	--------------	---------	-------------

WardID				
74201001	0	35	0	0.000000
74201002	0	37	4	0.108108
74201003	0	33	1	0.030303
74201004	1	34	1	0.029412
74201005	0	30	0	0.000000
...
79900104	1	32	3	0.093750
79900105	0	30	0	0.000000
79900106	0	37	7	0.189189
79900107	2	44	14	0.318182
Total Number	341	23924	2524	0.105501

[530 rows x 8 columns]

```
[57]: BRT_use['YES Use BRT Percent'] = BRT_use['YES Use BRT']*100
```

I now want to create a column of respondents that **do not** use BRT

```
[58]: BRT_use['NO BRT use']=BRT_use['Never']/BRT_use['Total Number']
```

BRT_use

```
[58]: Q5_11_BRT_Freq  A few times a month  Every few months  Most days  Never  \
```

WardID				
74201001	0	0	0	35
74201002	2	1	1	33
74201003	1	0	0	32
74201004	0	0	0	33
74201005	0	0	0	30
...
79900104	0	1	1	29
79900105	0	0	0	30
79900106	2	5	0	30
79900107	3	2	7	30
Total Number	887	836	460	21400

```
Q5_11_BRT_Freq  Once or twice a week  Total Number  Use BRT  YES Use BRT  \
```

WardID				
74201001	0	35	0	0.000000
74201002	0	37	4	0.108108
74201003	0	33	1	0.030303
74201004	1	34	1	0.029412
74201005	0	30	0	0.000000
...
79900104	1	32	3	0.093750
79900105	0	30	0	0.000000

79900106	0	37	7	0.189189
79900107	2	44	14	0.318182
Total Number	341	23924	2524	0.105501

Q5_11_BRT_Freq	YES Use BRT Percent	NO BRT use
WardID		
74201001	0.000000	1.000000
74201002	10.810811	0.891892
74201003	3.030303	0.969697
74201004	2.941176	0.970588
74201005	0.000000	1.000000
...
79900104	9.375000	0.906250
79900105	0.000000	1.000000
79900106	18.918919	0.810811
79900107	31.818182	0.681818
Total Number	10.550075	0.894499

[530 rows x 10 columns]

```
[59]: BRT_use['No BRT Use Percent'] = BRT_use['NO BRT use']*100
```

Now that we have the percentages of BRT use and no BRT use, I will save them as a new dataframe

```
[60]: BRT_yes_no = BRT_use[['YES Use BRT Percent', 'No BRT Use Percent']]

BRT_yes_no
```

```
[60]: Q5_11_BRT_Freq  YES Use BRT Percent  No BRT Use Percent
WardID
74201001              0.000000          100.000000
74201002             10.810811           89.189189
74201003              3.030303           96.969697
74201004              2.941176           97.058824
74201005              0.000000          100.000000
...
79900104              9.375000           90.625000
79900105              0.000000          100.000000
79900106             18.918919           81.081081
79900107             31.818182           68.181818
Total Number          10.550075           89.449925
```

[530 rows x 2 columns]

However, I want to drop the “total number” column from the dataframe

```
[61]: BRT_yes_no = BRT_yes_no.drop('Total Number')
```

In order to map BRT use, I'll merge it with the 2017 wards data

```
[62]: BRT_use_2017 = wards_2017.merge(BRT_yes_no, on='WardID')
```

```
BRT_use_2017
```

```
[62]:
```

	OBJECTID	ProvinceCode	ProvinceName	LocalMunicipalityCode	WardNumber	\
0	1504	GT	Gauteng	EKU	60	
1	1505	GT	Gauteng	EKU	61	
2	1506	GT	Gauteng	EKU	62	
3	1507	GT	Gauteng	EKU	63	
4	1508	GT	Gauteng	EKU	64	
..	
524	4319	GT	Gauteng	EKU	108	
525	4320	GT	Gauteng	EKU	109	
526	4321	GT	Gauteng	EKU	110	
527	4322	GT	Gauteng	EKU	111	
528	4323	GT	Gauteng	EKU	112	

	WardID	LocalMunicipalityName	DistrictMunicipalityCode	\
0	79700060	Ekurhuleni	EKU	
1	79700061	Ekurhuleni	EKU	
2	79700062	Ekurhuleni	EKU	
3	79700063	Ekurhuleni	EKU	
4	79700064	Ekurhuleni	EKU	
..	
524	79700108	Ekurhuleni	EKU	
525	79700109	Ekurhuleni	EKU	
526	79700110	Ekurhuleni	EKU	
527	79700111	Ekurhuleni	EKU	
528	79700112	Ekurhuleni	EKU	

	DistrictMunicipalityName	Year	Shape__Area	Shape__Length	\
0	Ekurhuleni	2016	3.898607e+06	11472.007890	
1	Ekurhuleni	2016	2.040419e+07	24911.970404	
2	Ekurhuleni	2016	5.823267e+07	42030.549363	
3	Ekurhuleni	2016	3.332302e+06	10253.147328	
4	Ekurhuleni	2016	1.239711e+07	18521.992239	
..	
524	Ekurhuleni	2016	3.986026e+06	12293.042422	
525	Ekurhuleni	2016	7.126183e+06	17501.893757	
526	Ekurhuleni	2016	8.281005e+06	15886.916039	
527	Ekurhuleni	2016	2.570824e+07	29656.849051	
528	Ekurhuleni	2016	6.723852e+06	14259.004191	

	geometry	YES Use BRT Percent	\
0	POLYGON ((28.16493 -26.35885, 28.16632 -26.358...	3.773585	

```

1    POLYGON ((28.13552 -26.40783, 28.13665 -26.408...      7.407407
2    POLYGON ((28.24732 -26.38152, 28.25052 -26.386...      3.846154
3    POLYGON ((28.16205 -26.37103, 28.16219 -26.371...     10.204082
4    POLYGON ((28.22547 -26.35844, 28.22591 -26.358...      3.773585
..
524  POLYGON ((28.19092 -26.37677, 28.19174 -26.377...      7.843137
525  POLYGON ((28.43689 -26.10593, 28.43703 -26.106...      7.547170
526  POLYGON ((28.39511 -26.14457, 28.39570 -26.145...      8.620690
527  POLYGON ((28.43582 -26.31314, 28.43659 -26.313...      0.000000
528  POLYGON ((28.35914 -26.35703, 28.35931 -26.358...      1.694915

```

```

      No BRT Use Percent
0          96.226415
1          92.592593
2          96.153846
3          89.795918
4          96.226415
..
524        92.156863
525        92.452830
526        91.379310
527       100.000000
528        98.305085

```

[529 rows x 15 columns]

It looks like the merge was successful!

1.6.3 Exploring BRT Use

I want to see the wards in which BRT use is highest and lowest

```
[63]: BRT_use_highest = BRT_use_2017.sort_values(by="YES Use BRT Percent",
→ascending=False).head(10)
```

BRT_use_highest

```
[63]: OBJECTID ProvinceCode ProvinceName LocalMunicipalityCode WardNumber \
59      2568          GT      Gauteng          JHB      29
63      2572          GT      Gauteng          JHB      33
42      2551          GT      Gauteng          JHB      12
76      2585          GT      Gauteng          JHB      46
55      2564          GT      Gauteng          JHB      25
61      2570          GT      Gauteng          JHB      31
90      2599          GT      Gauteng          JHB      60
69      2578          GT      Gauteng          JHB      39
99      2608          GT      Gauteng          JHB      69
60      2569          GT      Gauteng          JHB      30

```

	WardID	LocalMunicipalityName	DistrictMunicipalityCode	\
59	79800029	City of Johannesburg	JHB	
63	79800033	City of Johannesburg	JHB	
42	79800012	City of Johannesburg	JHB	
76	79800046	City of Johannesburg	JHB	
55	79800025	City of Johannesburg	JHB	
61	79800031	City of Johannesburg	JHB	
90	79800060	City of Johannesburg	JHB	
69	79800039	City of Johannesburg	JHB	
99	79800069	City of Johannesburg	JHB	
60	79800030	City of Johannesburg	JHB	

	DistrictMunicipalityName	Year	Shape__Area	Shape__Length	\
59	City of Johannesburg	2016	6.702357e+06	13310.759165	
63	City of Johannesburg	2016	3.843595e+06	10476.516439	
42	City of Johannesburg	2016	2.946026e+06	9074.125560	
76	City of Johannesburg	2016	4.737237e+06	11810.295027	
55	City of Johannesburg	2016	5.999320e+06	13354.510774	
61	City of Johannesburg	2016	2.184496e+06	8753.932909	
90	City of Johannesburg	2016	7.107358e+06	15169.436463	
69	City of Johannesburg	2016	4.312237e+06	10606.390622	
99	City of Johannesburg	2016	5.913493e+06	15658.668156	
60	City of Johannesburg	2016	3.185999e+06	10554.740483	

	geometry	YES Use BRT Percent	\
59	POLYGON ((27.95300 -26.21879, 27.95342 -26.218...	54.166667	
63	POLYGON ((27.85990 -26.25643, 27.85992 -26.257...	50.847458	
42	POLYGON ((27.85868 -26.27761, 27.86215 -26.279...	50.000000	
76	POLYGON ((27.87445 -26.23159, 27.87516 -26.232...	47.272727	
55	POLYGON ((27.91362 -26.24990, 27.91389 -26.249...	46.774194	
61	POLYGON ((27.92653 -26.22928, 27.92691 -26.229...	44.897959	
90	POLYGON ((28.03264 -26.18669, 28.03288 -26.186...	44.615385	
69	POLYGON ((27.92109 -26.21904, 27.92116 -26.219...	43.750000	
99	POLYGON ((27.99919 -26.17160, 27.99917 -26.171...	42.592593	
60	POLYGON ((27.92039 -26.23825, 27.92045 -26.238...	41.818182	

	No BRT Use Percent
59	45.833333
63	49.152542
42	50.000000
76	52.727273
55	53.225806
61	55.102041
90	55.384615
69	56.250000
99	57.407407

60

58.181818

I want to trim the columns to only what I need

```
[64]: BRT_use_highest= BRT_use_highest[['WardID', 'LocalMunicipalityName',
    ↪ 'DistrictMunicipalityName', 'YES Use BRT Percent']]
```

BRT_use_highest

```
[64]:      WardID LocalMunicipalityName DistrictMunicipalityName \
59  79800029  City of Johannesburg      City of Johannesburg
63  79800033  City of Johannesburg      City of Johannesburg
42  79800012  City of Johannesburg      City of Johannesburg
76  79800046  City of Johannesburg      City of Johannesburg
55  79800025  City of Johannesburg      City of Johannesburg
61  79800031  City of Johannesburg      City of Johannesburg
90  79800060  City of Johannesburg      City of Johannesburg
69  79800039  City of Johannesburg      City of Johannesburg
99  79800069  City of Johannesburg      City of Johannesburg
60  79800030  City of Johannesburg      City of Johannesburg

      YES Use BRT Percent
59              54.166667
63              50.847458
42              50.000000
76              47.272727
55              46.774194
61              44.897959
90              44.615385
69              43.750000
99              42.592593
60              41.818182
```

From the table above, we can see that BRT use is the highest in Johannesburg.

Now I will explore the wards with the lowest BRT use

```
[65]: BRT_use_lowest = BRT_use_2017.sort_values(by="No BRT Use Percent",
    ↪ ascending=False).head(10)
```

BRT_use_lowest

```
[65]:      OBJECTID ProvinceCode ProvinceName LocalMunicipalityCode WardNumber \
418      4213          GT      Gauteng      GT421          44
419      4214          GT      Gauteng      GT421          45
320      3851          GT      Gauteng      GT481          39
312      3843          GT      Gauteng      GT481          31
388      4183          GT      Gauteng      GT421          14
```

389	4184	GT	Gauteng	GT421	15
390	4185	GT	Gauteng	GT421	16
311	3842	GT	Gauteng	GT481	30
309	3840	GT	Gauteng	GT481	28
307	3777	GT	Gauteng	GT485	35

	WardID	LocalMunicipalityName	DistrictMunicipalityCode	\
418	74201044	Emfuleni	DC42	
419	74201045	Emfuleni	DC42	
320	74801039	Mogale City	DC48	
312	74801031	Mogale City	DC48	
388	74201014	Emfuleni	DC42	
389	74201015	Emfuleni	DC42	
390	74201016	Emfuleni	DC42	
311	74801030	Mogale City	DC48	
309	74801028	Mogale City	DC48	
307	74205035	Rand West City	DC48	

	DistrictMunicipalityName	Year	Shape__Area	Shape__Length	\
418	Sedibeng	2016	4.761278e+06	13449.434506	
419	Sedibeng	2016	7.117027e+07	37262.654049	
320	West Rand	2016	3.180230e+08	100321.254092	
312	West Rand	2016	2.515273e+08	82753.822200	
388	Sedibeng	2016	3.602065e+06	8982.118487	
389	Sedibeng	2016	2.856751e+07	26324.569849	
390	Sedibeng	2016	3.708146e+07	36565.083948	
311	West Rand	2016	1.147148e+08	57707.915574	
309	West Rand	2016	7.648468e+06	18111.940038	
307	West Rand	2016	1.223508e+07	21544.061485	

	geometry	YES Use BRT Percent	\
418	POLYGON ((27.84854 -26.52579, 27.84919 -26.528...	0.0	
419	POLYGON ((27.92167 -26.56789, 27.92210 -26.568...	0.0	
320	POLYGON ((27.86093 -25.94484, 27.85577 -25.948...	0.0	
312	POLYGON ((27.54490 -25.99084, 27.54509 -25.991...	0.0	
388	POLYGON ((27.86692 -26.68585, 27.86778 -26.687...	0.0	
389	POLYGON ((27.95406 -26.64993, 27.95643 -26.651...	0.0	
390	POLYGON ((27.98077 -26.55415, 27.98086 -26.554...	0.0	
311	POLYGON ((27.70048 -26.06088, 27.70052 -26.061...	0.0	
309	POLYGON ((27.84682 -26.06431, 27.84773 -26.064...	0.0	
307	POLYGON ((27.70385 -26.28741, 27.70384 -26.288...	0.0	

	No BRT Use Percent
418	100.0
419	100.0
320	100.0
312	100.0

388	100.0
389	100.0
390	100.0
311	100.0
309	100.0
307	100.0

```
[66]: ward_low_BRT_use_2011 = BRT_use_2017[BRT_use_2017['No BRT Use Percent']==100]
```

```
[67]: print ('There are ' + str(len(ward_low_BRT_use_2011)) + ' wards in which BRT_
↪use is 0%')
```

There are 68 wards in which BRT use is 0%

As we can see from above, there are 68 wards in which BRT use is 0. I would like to see the wards in which BRT use is the lowest. Though 68 wards is a lot to scroll through!

```
[68]: ward_low_BRT_use_2017 = ward_low_BRT_use_2011[['WardID',
↪'LocalMunicipalityName', 'No BRT Use Percent']]
pd.set_option('display.max_rows',68)
ward_low_BRT_use_2017
```

```
[68]:
```

	WardID	LocalMunicipalityName	No BRT Use Percent
6	79700066	Ekurhuleni	100.0
9	79700069	Ekurhuleni	100.0
27	79700087	Ekurhuleni	100.0
34	79800004	City of Johannesburg	100.0
35	79800005	City of Johannesburg	100.0
109	79800079	City of Johannesburg	100.0
207	79900043	City of Tshwane	100.0
266	79900103	City of Tshwane	100.0
268	79900105	City of Tshwane	100.0
273	74205001	Rand West City	100.0
275	74205003	Rand West City	100.0
276	74205004	Rand West City	100.0
279	74205007	Rand West City	100.0
282	74205010	Rand West City	100.0
285	74205013	Rand West City	100.0
286	74205014	Rand West City	100.0
292	74205020	Rand West City	100.0
298	74205026	Rand West City	100.0
299	74205027	Rand West City	100.0
301	74205029	Rand West City	100.0
303	74205031	Rand West City	100.0
305	74205033	Rand West City	100.0
307	74205035	Rand West City	100.0
309	74801028	Mogale City	100.0
311	74801030	Mogale City	100.0

312	74801031	Mogale City	100.0
320	74801039	Mogale City	100.0
324	74804004	Merafong City	100.0
325	74804005	Merafong City	100.0
328	74804008	Merafong City	100.0
329	74804009	Merafong City	100.0
332	74804012	Merafong City	100.0
334	74804014	Merafong City	100.0
337	74804017	Merafong City	100.0
342	74804022	Merafong City	100.0
345	74804025	Merafong City	100.0
347	74804027	Merafong City	100.0
348	74804028	Merafong City	100.0
352	74801004	Mogale City	100.0
353	74801005	Mogale City	100.0
358	74801010	Mogale City	100.0
364	74801016	Mogale City	100.0
366	74801018	Mogale City	100.0
368	74801020	Mogale City	100.0
369	74801021	Mogale City	100.0
375	74201001	Emfuleni	100.0
379	74201005	Emfuleni	100.0
388	74201014	Emfuleni	100.0
389	74201015	Emfuleni	100.0
390	74201016	Emfuleni	100.0
397	74201023	Emfuleni	100.0
418	74201044	Emfuleni	100.0
419	74201045	Emfuleni	100.0
424	74202005	Midvaal	100.0
430	74202011	Midvaal	100.0
431	74202012	Midvaal	100.0
434	74202015	Midvaal	100.0
435	74203001	Lesedi	100.0
437	74203003	Lesedi	100.0
438	74203004	Lesedi	100.0
440	74203006	Lesedi	100.0
441	74203007	Lesedi	100.0
444	74203010	Lesedi	100.0
445	74203011	Lesedi	100.0
446	74203012	Lesedi	100.0
473	79700026	Ekurhuleni	100.0
512	79700096	Ekurhuleni	100.0
527	79700111	Ekurhuleni	100.0

Choropleth map And now, to create an interactive Plotly map, I'll convert the data to geojson and lat/lon

```

[69]: BRT_wards_2017 = BRT_use_2017.to_crs(epsg=4326) # convert the coordinate ↵
      ↪reference system to lat/long
      BRT_wards_2017_json = BRT_use_2017.__geo_interface__ #covert to geoJSON

[70]: fig = px.choropleth_mapbox(BRT_wards_2017, geojson=BRT_wards_2017_json, ↵
      ↪locations=BRT_wards_2017.index, color='YES Use BRT Percent',
      color_continuous_scale="Greens",
      range_color=(0, 60),
      hover_name= BRT_wards_2017['WardID'],
      mapbox_style="carto-positron",
      zoom=7, center = {"lat":-26.270760, "lon": 28.
      ↪112268},

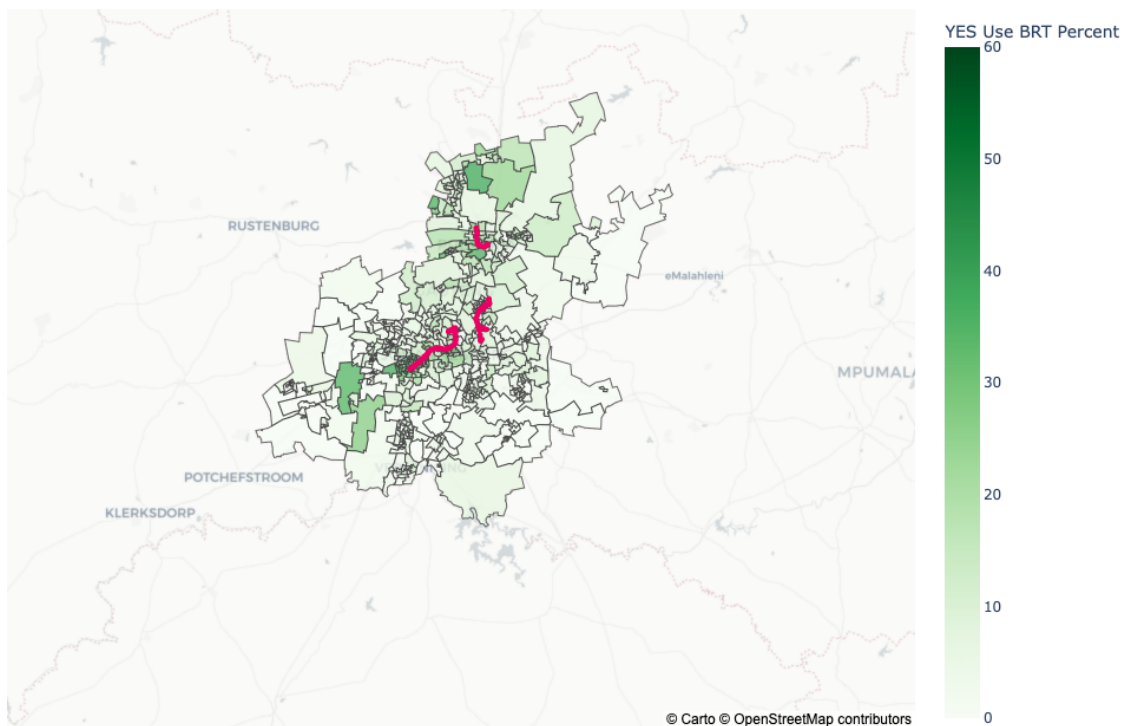
      opacity=0.8,
      title="BRT Use by ward(green) with BRT lines(red)"
      )

fig.add_scattermapbox(
    lat = lats,
    lon = lons,
    marker_size=5,
    marker_color='rgb(235, 0, 100)'
)

fig.update_layout(height=650, margin={"r":0,"t":40,"l":0,"b":0})
fig.show()
#fig.write_html("BRT use_wards.html")

```

BRT Use by ward(green) with BRT lines(red)



The interactive map allows us to see the use of BRT along with the BRT lines, and as hypothesized BRT use is higher in wards that have a BRT line. There are a few wards (74205025, 74205030, 79900095, 79900030) that have a pretty high (~30%) BRT use but aren't adjacent to a line. I want to highlight the top 10 wards in which BRT use is highest, but I haven't been able to figure out how I can do so in an interactive map, so instead I'll be creating a static map

```
[71]: fig, ax = plt.subplots(figsize=(12,12))

BRT_use_2017.plot(ax=ax,
                  column=BRT_use_2017['YES Use BRT Percent'],
                  legend=True,
                  scheme='user_defined',
                  classification_kwds={'bins': [10,20,30,40,50,60]},
                  cmap='Greens',
                  edgecolor='black',
                  linewidth=0.1
                )

BRT_use_2017[BRT_use_2017['YES Use BRT Percent'] >= 40].boundary.plot(ax=ax,
                              alpha=0.5,
                              linewidth=3,
                              hatch="///",
                              color='gold')
```

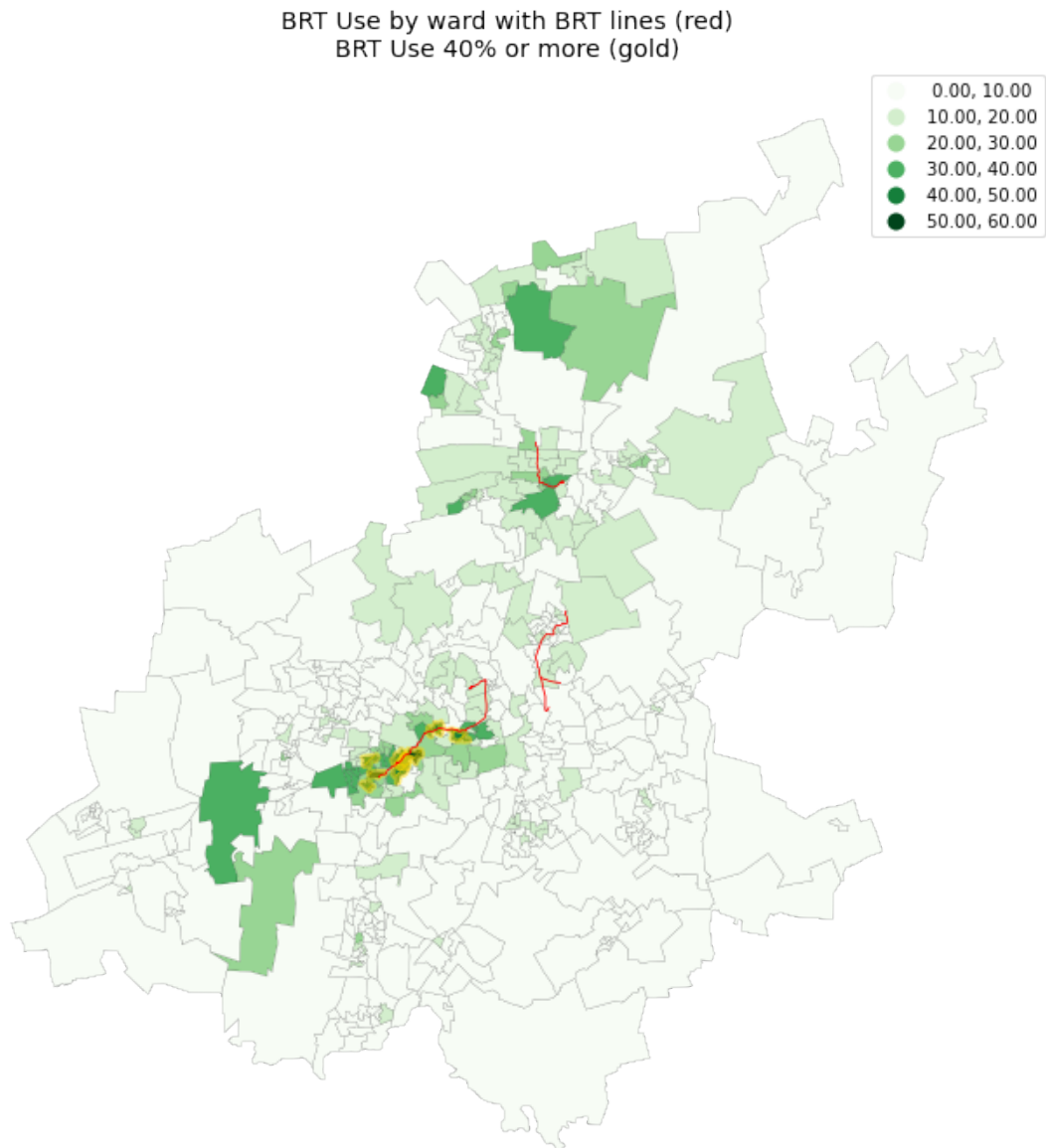
```

)

BRT_line.plot(ax=ax, color='red', linewidth=0.8)

ax.set_title('BRT Use by ward with BRT lines (red)\n' + 'BRT Use 40% or more_↪(gold)', fontsize=14)
ax.axis('off');
fig.savefig('BRT_use.png')

```



As we identified in the interactive map, we can see that the wards in which BRT is greater than

40% are closest to the Rea Vaya in Johannesburg and is generally higher in areas that are near the BRT. I want to see the list of ward numbers

```
[72]: print (BRT_use_2017[['WardID', 'YES Use BRT Percent']][BRT_use_2017['YES Use_
↳BRT Percent'] >= 40])
```

	WardID	YES Use BRT Percent
42	79800012	50.000000
55	79800025	46.774194
59	79800029	54.166667
60	79800030	41.818182
61	79800031	44.897959
63	79800033	50.847458
69	79800039	43.750000
76	79800046	47.272727
90	79800060	44.615385
99	79800069	42.592593

Now that we have the wards with BRT use more than 40%, we can see examine the attitudes in these wards

1.7 Social attitudes in wards that the BRT passes through

I will import the data for social attitudes by ward that Yasmina created. Each dataset shows the mean attitude about attitudes toward foreigners and about trust between Black and White individuals. I want to compare the attitudes of respondents in the wards the BRT passes through to see if they have improved over time

```
[73]: BRT_attitudes_2011 = pd.read_csv('BRT_wards_attitudes_2011.csv')
```

```
[74]: BRT_attitudes_2017 = pd.read_csv('BRT_wards_attitudes_2017.csv')
```

1.7.1 2011

Let's explore the data

```
[75]: BRT_attitudes_2011.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48 entries, 0 to 47
Data columns (total 4 columns):
#   Column                Non-Null Count  Dtype
---  -
0   WardID                 48 non-null    int64
1   BRT Line               48 non-null    object
2   att_foreign_mean       48 non-null    float64
3   bw_trust_mean          47 non-null    float64
dtypes: float64(2), int64(1), object(1)
memory usage: 1.6+ KB
```

```
[76]: BRT_attitudes_2011.head()
```

```
[76]:      WardID  BRT Line  att_foreign_mean  bw_trust_mean
0  79800019  Rea Vaya      0.363636      1.139535
1  79800033  Rea Vaya      0.205882      0.901961
2  79800037  Rea Vaya      0.162791      1.000000
3  79800038  Rea Vaya      0.145833      1.166667
4  79800030  Rea Vaya      0.081081      0.947368
```

I want to merge this data with ward geospatial data but first need to convert the WardID column from object to int

```
[77]: wards_2011["WardID"] = pd.to_numeric(wards_2011["WardID"])
```

```
[78]: BRT_attitudes_2011_gdf = wards_2011.merge(BRT_attitudes_2011, on='WardID')

BRT_attitudes_2011_gdf
```

```
[78]:      OBJECTID  ProvinceCode  ProvinceName  LocalMunicipalityCode  WardNumber  \
0           600             GT      Gauteng             JHB           67
1          4877             GT      Gauteng             JHB           67
2           601             GT      Gauteng             JHB           68
3          4878             GT      Gauteng             JHB           68
4           602             GT      Gauteng             JHB           69
..          ...             ...             ...             ...             ...
91          7840             GT      Gauteng             TSH           81
92          3574             GT      Gauteng             TSH           92
93          7851             GT      Gauteng             TSH           92
94          3578             GT      Gauteng             TSH           96
95          7855             GT      Gauteng             TSH           96
```

```
      WardID  LocalMunicipalityName  DistrictMunicipalityCode  \
0  79800067  City of Johannesburg             JHB
1  79800067  City of Johannesburg             JHB
2  79800068  City of Johannesburg             JHB
3  79800068  City of Johannesburg             JHB
4  79800069  City of Johannesburg             JHB
..          ...             ...             ...
91  79900081  City of Tshwane             TSH
92  79900092  City of Tshwane             TSH
93  79900092  City of Tshwane             TSH
94  79900096  City of Tshwane             TSH
95  79900096  City of Tshwane             TSH
```

```
      DistrictMunicipalityName  Year  Shape__Area  Shape__Length  \
0  City of Johannesburg  2011  5.822398e+06  14839.026704
1  City of Johannesburg  2011  5.822398e+06  14839.026704
```

```

2      City of Johannesburg  2011  2.196683e+07  26442.251672
3      City of Johannesburg  2011  2.196683e+07  26442.251672
4      City of Johannesburg  2011  9.416403e+06  17883.257578
..
91      City of Tshwane  2011  1.287117e+06  5579.556212
92      City of Tshwane  2011  6.850188e+06  14113.720798
93      City of Tshwane  2011  6.850188e+06  14113.720798
94      City of Tshwane  2011  3.015758e+08  101853.252738
95      City of Tshwane  2011  3.015758e+08  101853.252738

                                geometry  BRT Line  \
0  POLYGON ((28.05405 -26.16921, 28.05378 -26.171...  Rea Vaya
1  POLYGON ((28.05405 -26.16921, 28.05378 -26.171...  Rea Vaya
2  POLYGON ((27.97838 -26.19068, 27.97831 -26.191...  Rea Vaya
3  POLYGON ((27.97838 -26.19068, 27.97831 -26.191...  Rea Vaya
4  POLYGON ((27.99591 -26.17295, 27.99494 -26.176...  Rea Vaya
..
91  POLYGON ((28.19874 -25.74177, 28.19905 -25.741...  A Re Yang
92  POLYGON ((28.21090 -25.72931, 28.21094 -25.729...  A Re Yang
93  POLYGON ((28.21090 -25.72931, 28.21094 -25.729...  A Re Yang
94  POLYGON ((28.14057 -25.49034, 28.14042 -25.493...  A Re Yang
95  POLYGON ((28.14057 -25.49034, 28.14042 -25.493...  A Re Yang

att_foreign_mean  bw_trust_mean
0                0.066667      1.684211
1                0.066667      1.684211
2                0.307692      1.400000
3                0.307692      1.400000
4                0.180000      1.266667
..
91               0.300000      1.434783
92               0.269231      1.615385
93               0.269231      1.615385
94               0.250000      1.111111
95               0.250000      1.111111

```

[96 rows x 16 columns]

I want to create an interactive map of both attitudes toward foreigners and black and white trust variables to see if there is any change between 2011 and 2017. While we have developed a purple/blue color scheme for our variables, to easily identify any changes, I will be using sequential color schemes for these maps

```

[79]: BRT_attitudes_2011_gdf = BRT_attitudes_2011_gdf.to_crs(epsg=4326) # convert the
      ↪ coordinate reference system to lat/long
      BRT_attitudes_2011_json = BRT_attitudes_2011_gdf.__geo_interface__ #convert to
      ↪ geoJSON

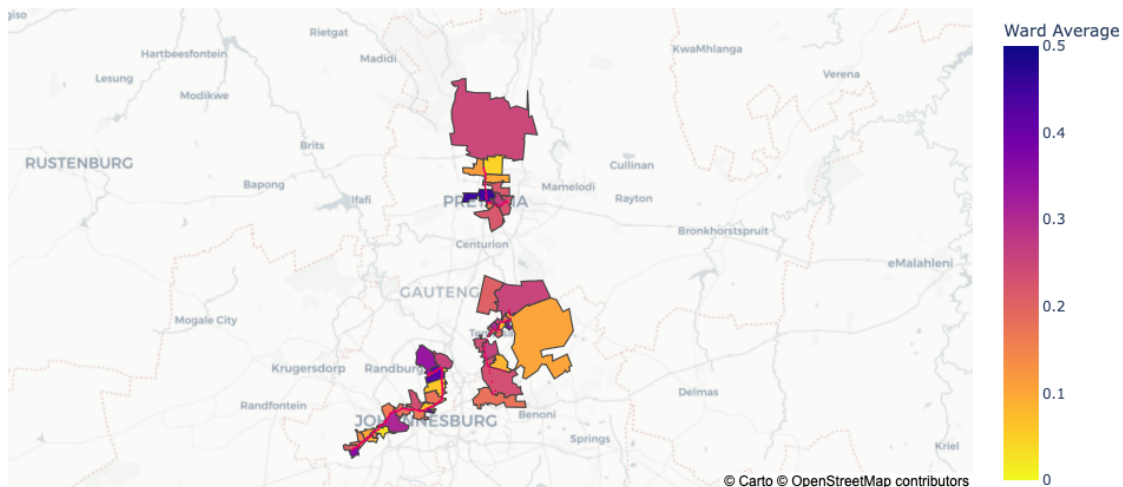
```

```
[80]: fig = px.choropleth_mapbox(BRT_attitudes_2011_gdf,
    ↳ geojson=BRT_attitudes_2011_json, locations=BRT_attitudes_2011_gdf.index,
    ↳ color='att_foreign_mean',
        color_continuous_scale="Plasma_r",
        range_color=(0, 0.5),
        hover_name= BRT_attitudes_2011_gdf['WardID'],
        mapbox_style="carto-positron",
        zoom=8, center = {"lat":-25.850187, "lon": 28.
    ↳ 198042},
        opacity=0.9,
        title="Average attitude toward foreigners with BRT
    ↳ lines(red) 2011",
        labels={"att_foreign_mean": "Ward Average"}
    )

fig.add_scattermapbox(
    lat = lats,
    lon = lons,
    marker_size=2,
    marker_color='rgb(235, 0, 100)'
)

fig.update_layout(height=450, margin={"r":0,"t":40,"l":0,"b":0})
fig.show()
fig.write_html("BRT att_foreigners_2011.html")
```

Average attitude toward foreigners with BRT lines(red) 2011



```
[81]: fig = px.choropleth_mapbox(BRT_attitudes_2011_gdf,
    ↳ geojson=BRT_attitudes_2011_json, locations=BRT_attitudes_2011_gdf.index,
    ↳ color='bw_trust_mean',
```



```

        color_continuous_scale="Viridis_r",
        range_color=(0.9, 2),
        hover_name= BRT_attitudes_2011_gdf['WardID'],
        mapbox_style="carto-positron",
        zoom=8, center = {"lat":-25.850187, "lon": 28.
↪198042},

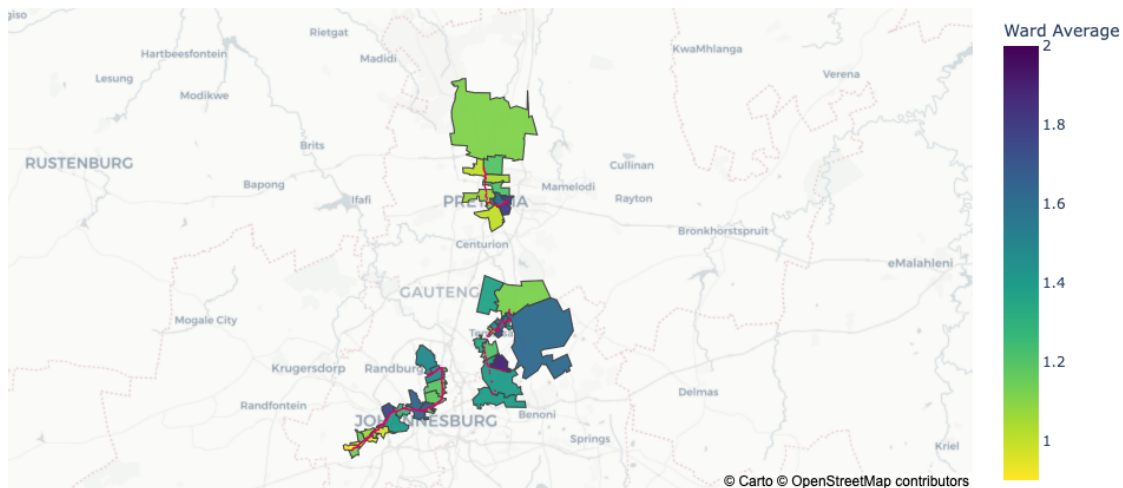
        opacity=0.8,
        title="Average attitude of black and white trust_
↪with BRT lines(red) 2011",
        labels={"bw_trust_mean": "Ward Average"}
    )

fig.add_scattermapbox(
    lat = lats,
    lon = lons,
    marker_size=2,
    marker_color='rgb(235, 0, 100)'
)

fig.update_layout(height=450, margin={"r":0,"t":40,"l":0,"b":0})
fig.show()
fig.write_html("BRT bw_trust_2011.html")

```

Average attitude of black and white trust with BRT lines(red) 2011



1.7.2 2017

```
[82]: wards_2017["WardID"] = pd.to_numeric(wards_2017["WardID"])
```

```
[83]: BRT_attitudes_2017_gdf = wards_2017.merge(BRT_attitudes_2017, on='WardID')
pd.set_option('display.max_rows',20)
```

BRT_attitudes_2017_gdf

```
[83]:
```

	OBJECTID	ProvinceCode	ProvinceName	LocalMunicipalityCode	WardNumber	\
0	1533	GT	Gauteng	EKU	89	
1	2558	GT	Gauteng	JHB	19	
2	2568	GT	Gauteng	JHB	29	
3	2569	GT	Gauteng	JHB	30	
4	2570	GT	Gauteng	JHB	31	
..	
43	4256	GT	Gauteng	EKU	14	
44	4258	GT	Gauteng	EKU	16	
45	4259	GT	Gauteng	EKU	17	
46	4303	GT	Gauteng	EKU	92	
47	4311	GT	Gauteng	EKU	100	

	WardID	LocalMunicipalityName	DistrictMunicipalityCode	\
0	79700089	Ekurhuleni	EKU	
1	79800019	City of Johannesburg	JHB	
2	79800029	City of Johannesburg	JHB	
3	79800030	City of Johannesburg	JHB	
4	79800031	City of Johannesburg	JHB	
..	
43	79700014	Ekurhuleni	EKU	
44	79700016	Ekurhuleni	EKU	
45	79700017	Ekurhuleni	EKU	
46	79700092	Ekurhuleni	EKU	
47	79700100	Ekurhuleni	EKU	

	DistrictMunicipalityName	Year	Shape__Area	Shape__Length	\
0	Ekurhuleni	2016	1.657774e+08	65119.929571	
1	City of Johannesburg	2016	4.182003e+06	10549.269035	
2	City of Johannesburg	2016	6.702357e+06	13310.759165	
3	City of Johannesburg	2016	3.185999e+06	10554.740483	
4	City of Johannesburg	2016	2.184496e+06	8753.932909	
..	
43	Ekurhuleni	2016	2.980619e+06	7936.190115	
44	Ekurhuleni	2016	1.319614e+07	18398.339029	
45	Ekurhuleni	2016	6.727162e+07	44019.922369	
46	Ekurhuleni	2016	2.229463e+07	23000.064387	
47	Ekurhuleni	2016	1.269119e+06	5480.687821	

	geometry	att_foreign_mean	\
0	POLYGON ((28.35238 -25.95460, 28.36680 -25.979...	0.135593	
1	POLYGON ((27.89006 -26.26877, 27.88949 -26.269...	0.092593	
2	POLYGON ((27.95300 -26.21879, 27.95342 -26.218...	0.211538	
3	POLYGON ((27.92039 -26.23825, 27.92045 -26.238...	0.357143	
4	POLYGON ((27.92653 -26.22928, 27.92691 -26.229...	0.196078	

```

..
43 POLYGON ((28.22457 -26.01760, 28.22458 -26.017... 0.134615
44 POLYGON ((28.26300 -26.08213, 28.25852 -26.091... 0.226415
45 POLYGON ((28.25397 -26.10145, 28.25495 -26.102... 0.129630
46 POLYGON ((28.19249 -26.13945, 28.19259 -26.139... 0.116667
47 POLYGON ((28.24533 -25.99403, 28.24537 -25.994... 0.086022

    bw_trust_mean
0      1.152542
1      1.314815
2      1.423077
3      1.035714
4      1.235294
..
43      1.173077
44      1.641509
45      1.555556
46      1.350000
47      1.290323

```

[48 rows x 15 columns]

```

[84]: BRT_attitudes_2017_gdf = BRT_attitudes_2017_gdf.to_crs(epsg=4326) # convert the
      ↪ coordinate reference system to lat/long
      BRT_attitudes_2017_json = BRT_attitudes_2017_gdf.__geo_interface__ #convert to
      ↪ geoJSON

```

```

[85]: fig = px.choropleth_mapbox(BRT_attitudes_2017_gdf,
      ↪ geojson=BRT_attitudes_2017_json, locations=BRT_attitudes_2017_gdf.index,
      ↪ color='att_foreign_mean',
      color_continuous_scale="Plasma_r",
      range_color=(0, 0.5),
      hover_name= BRT_attitudes_2017_gdf['WardID'],
      mapbox_style="carto-positron",
      zoom=8, center = {"lat":-25.850187, "lon": 28.
      ↪ 198042},
      opacity=0.8,
      title="Average attitude toward foreigners with BRT
      ↪ lines(red) 2017",
      labels={"att_foreign_mean": "Ward Average"}
      )

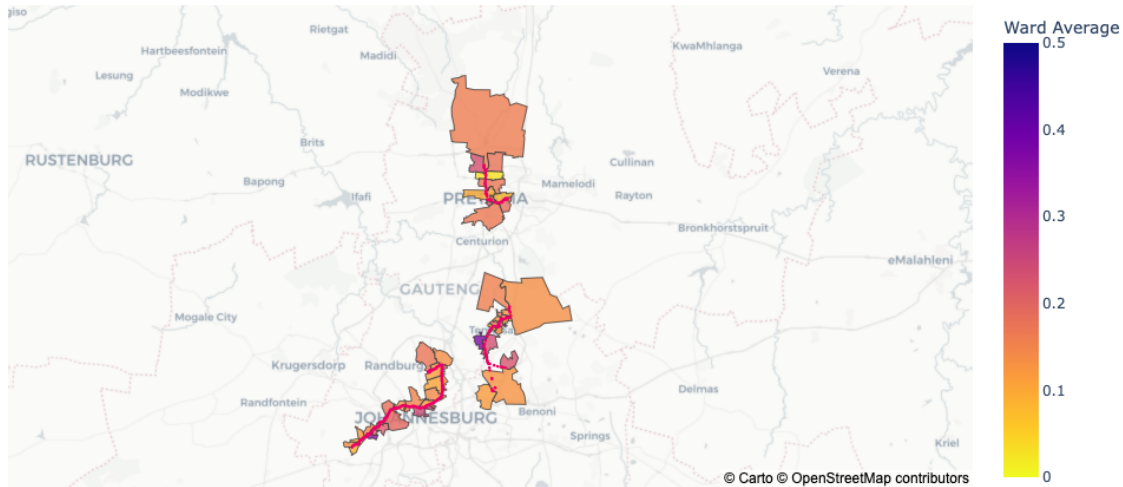
fig.add_scattermapbox(
    lat = lats,
    lon = lons,
    marker_size=3,
    marker_color='rgb(235, 0, 100)'

```

```
)

fig.update_layout(height=450, margin={"r":0,"t":40,"l":0,"b":0})
fig.show()
fig.write_html("BRT_att_foreigners_2017.html")
```

Average attitude toward foreigners with BRT lines(red) 2017

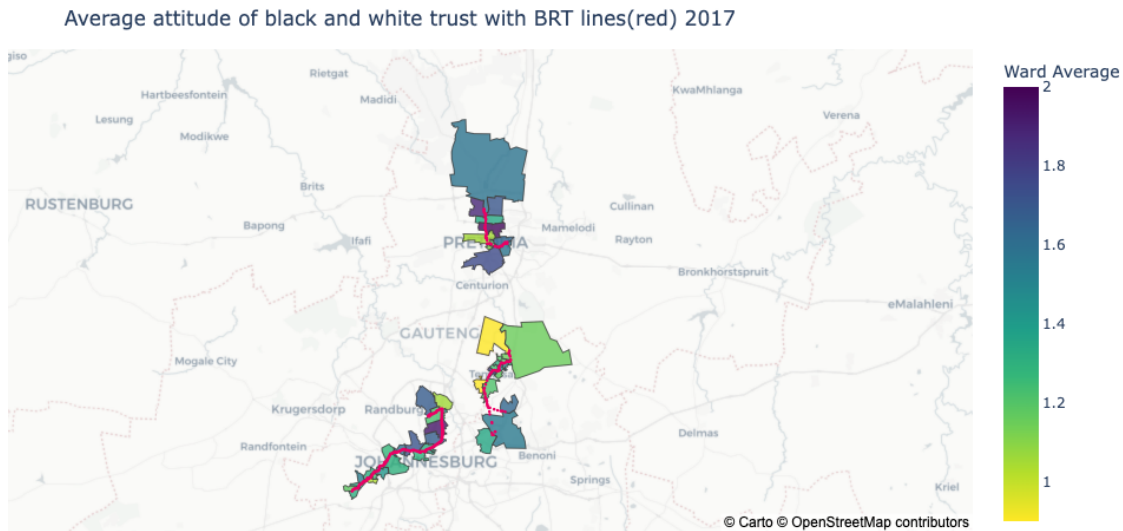


```
[86]: fig = px.choropleth_mapbox(BRT_attitudes_2017_gdf,
    ↪geojson=BRT_attitudes_2017_json, locations=BRT_attitudes_2017_gdf.index,
    ↪color='bw_trust_mean',
    color_continuous_scale="Viridis_r",
    range_color=(0.9, 2),
    hover_name= BRT_attitudes_2017_gdf['WardID'],
    mapbox_style="carto-positron",
    zoom=8, center = {"lat":-25.850187, "lon": 28.
    ↪198042},
    opacity=0.8,
    title="Average attitude of black and white trust,
    ↪with BRT lines(red) 2017",
    labels={"bw_trust_mean": "Ward Average"}
)

fig.add_scattermapbox(
    lat = lats,
    lon = lons,
    marker_size=3,
    marker_color='rgb(235, 0, 100)')

fig.update_layout(height=450, margin={"r":0,"t":40,"l":0,"b":0})
fig.show()
```

```
fig.write_html("BRT bw_trust_2017.html")
```



The interactive maps are helpful to see details of each ward, but I'd like to make a side by side static map to easily show any changes

```
[87]: BRT_attitudes_2011_gdf_web_mercator = BRT_attitudes_2011_gdf.to_crs(epsg=3857)
```

```
[88]: BRT_attitudes_2017_gdf_web_mercator = BRT_attitudes_2017_gdf.to_crs(epsg=3857)
```

```
[89]: # create the 1x2 subplots
# set sharex and sharey to true to scale each map equally
fig, axs = plt.subplots(1, 2, figsize=(15, 12), sharex=True, sharey=True)

# name each subplot
ax1, ax2 = axs

# 2011 on the left
base = BRT_attitudes_2011_gdf_web_mercator.plot(column='bw_trust_mean',
                                                  legend=True,
                                                  scheme='user_defined',
                                                  classification_kwds={'bins':[1, 1.25, 1.5, 1.75, 2.0]},
                                                  legend_kwds={'fmt':'{:0.2f}', 'loc':'upper_
→left', 'bbox_to_anchor':(1,.9)},
                                                  cmap='viridis_r', edgecolor='white', linewidth=0.05,
                                                  ax=ax1)

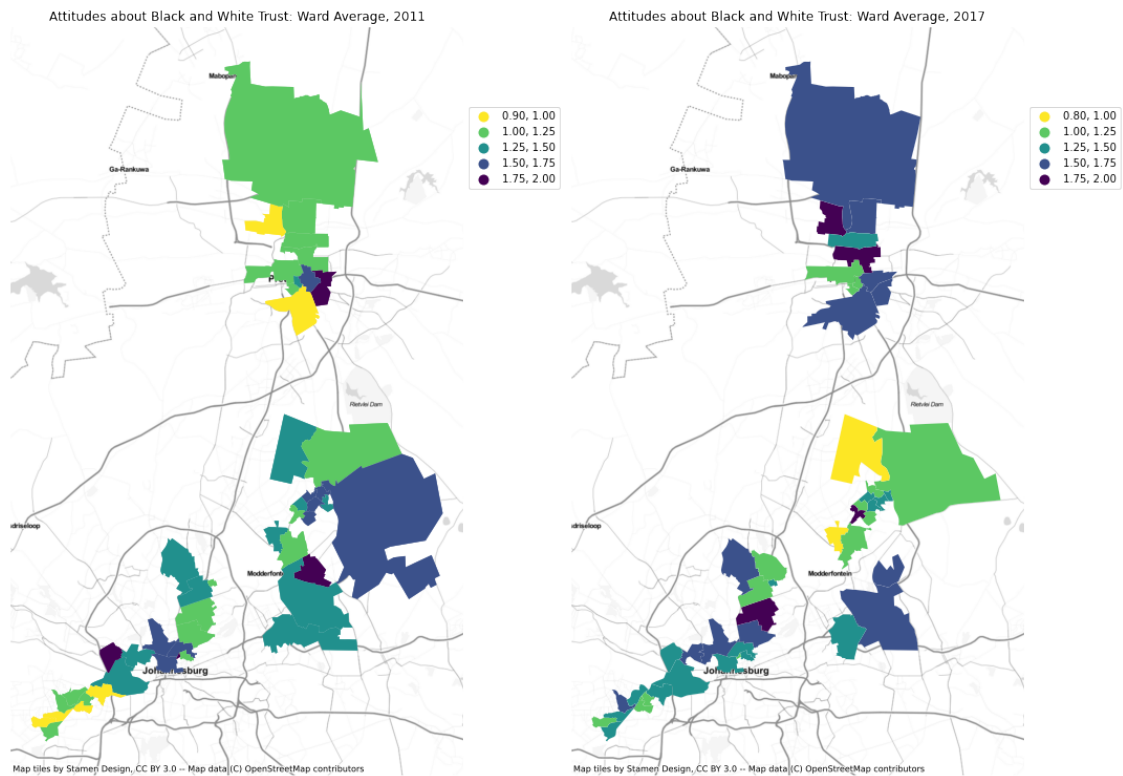
ax1.axis("off")
ax1.set_title('Attitudes about Black and White Trust: Ward Average, 2011')
ctx.add_basemap(ax1,source=ctx.providers.Stamen.TonerLite)
```

```

# 2017 on the right
base = BRT_attitudes_2017_gdf_web_mercator.plot(column='bw_trust_mean',
        legend=True,
        scheme='user_defined',
        classification_kwds={'bins':[1, 1.25, 1.5, 1.75, 2.0]},
        legend_kwds={'fmt':'{:0.2f}', 'loc':'upper_
↪left', 'bbox_to_anchor':(1,.9)},
        cmap='viridis_r', edgecolor='white', linewidth=0.05,
        ax=ax2)

ax2.axis("off")
ax2.set_title('Attitudes about Black and White Trust: Ward Average, 2017')
ctx.add_basemap(ax2,source=ctx.providers.Stamen.TonerLite)
plt.tight_layout()
fig.savefig('bw_trust_2011_2017.png', dpi='figure')

```



```

[90]: # create the 1x2 subplots
# set sharex and sharey to true to scale each map equally
fig, axs = plt.subplots(1, 2, figsize=(15, 12), sharex=True, sharey=True)

# name each subplot

```

```

ax1, ax2 = axs

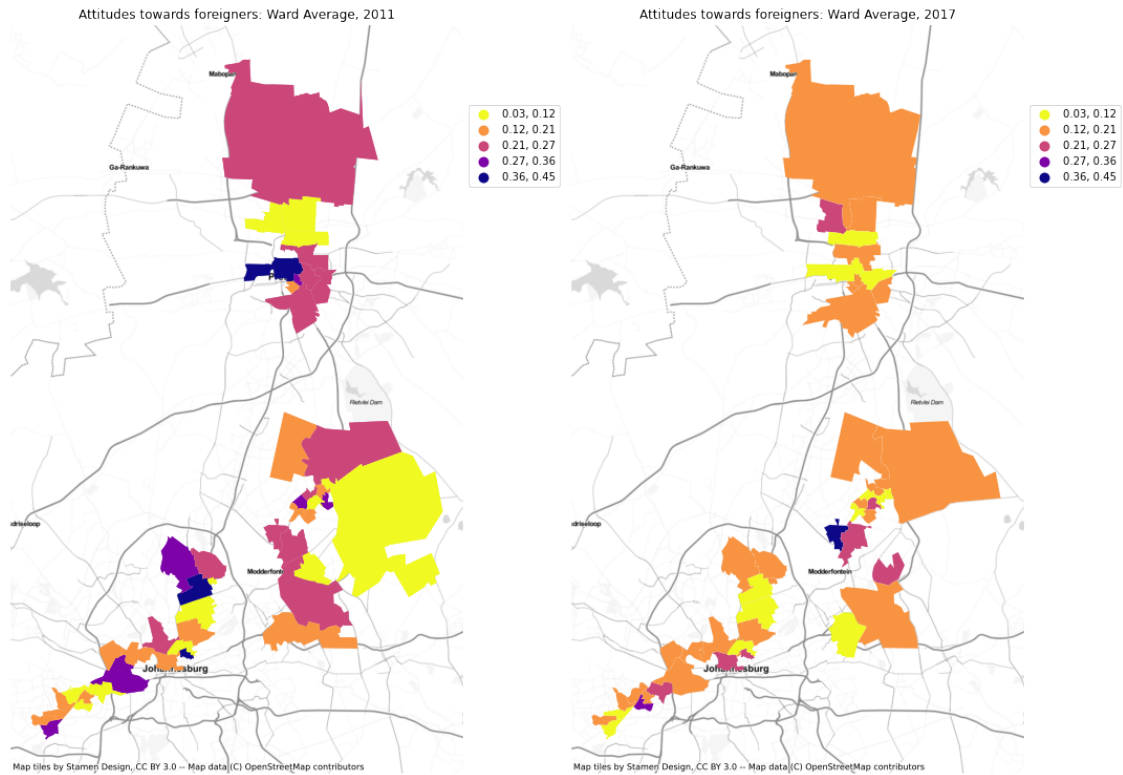
# 2011 on the left
base = BRT_attitudes_2011_gdf_web_mercator.plot(column='att_foreign_mean',
        legend=True,
        scheme='natural_breaks',
        # classification_kwds={'bins':[0.10, 0.20, 0.30, 0.40, 0.
        ↪50]}},
        legend_kwds={'fmt':'{: .2f}', 'loc':'upper_
        ↪left', 'bbox_to_anchor':(1,.9)},
        cmap='plasma_r', edgecolor='white', linewidth=0.05,
        ax=ax1)

ax1.axis("off")
ax1.set_title('Attitudes towards foreigners: Ward Average, 2011')
ctx.add_basemap(ax1,source=ctx.providers.Stamen.TonerLite)

# 2017 on the right
base = BRT_attitudes_2017_gdf_web_mercator.plot(column='att_foreign_mean',
        legend=True,
        scheme='user_defined',
        classification_kwds={'bins':[0.12, 0.21, 0.27, 0.36, 0.
        ↪45]}},
        legend_kwds={'fmt':'{: .2f}', 'loc':'upper_
        ↪left', 'bbox_to_anchor':(1,.9)},
        cmap='plasma_r', edgecolor='white', linewidth=0.05,
        ax=ax2)

ax2.axis("off")
ax2.set_title('Attitudes towards foreigners: Ward Average, 2017')
ctx.add_basemap(ax2,source=ctx.providers.Stamen.TonerLite)
plt.tight_layout()
fig.savefig('attitudes_2011_2017.png', dpi='figure')

```



1.8 Final Thoughts

We can see that there has been a shift in attitudes in both variables, however because we can't control for other conditions, we can't conclusively say BRT has made a significant impact on social attitudes. Additionally, there is still a lot of variation in responses between the wards