Kenya MTF Graphs

January 6, 2021

[1]:
```
# Explore Altair package with mini Kenya MTF dataset

!pip install altair
!pip install vega
!pip install vega_datasets

import pandas as pd
import numpy as np
import plotly as plt

import datascience
from datascience import *

import altair as alt
from vega_datasets import data
import vega
```

[2]:
```
# Load Kenya MTF Data

chunksize = 10

mtf_list = []
chunksize = 10
for chunk in pd.read_csv("kenya_mtf2_mini.csv", encoding='latin-1', chunksize =␣
 ↪chunksize):
    mtf_list.append(chunk)

kenya_mtf2_mini = pd.concat(mtf_list, axis=0)
kenya_mtf2_mini = pd.DataFrame(data = kenya_mtf2_mini)
```

[7]:
```
kenya_mtf2_mini.head()
```

[7]:

	parent_key	elc_aggr_tier	locality_ur	\
0	uuid:0006ae15-e9cf-419e-ac14-0c66a739366e	Tier 0	Urban	
1	uuid:001b24c5-30b9-41fe-ac90-9e9e3e476935	Tier 5	Urban	
2	uuid:0031732a-2efc-43e7-ba34-9447ddc31b32	Tier 2	Rural	
3	uuid:0042ae86-e063-4b44-a859-9c253d82bd38	Tier 0	Rural	

```
4  uuid:006014f9-c69e-48b2-8198-60cbbc894b59        Tier 0       Rural

   c_c_25bii  c_c_27b         c_c_30             c_c_31 c_c_119  c_c_123  \
0        NaN      NaN            NaN                NaN     NaN      NaN
1       24.0     18.0  Kerosene lamp  No back-up source     NaN      NaN
2        NaN      NaN            NaN                NaN     NaN      0.0
3        NaN      NaN            NaN                NaN     NaN      NaN
4        NaN      NaN            NaN                NaN     NaN      NaN

   c_c_149b_typicalmonth  … g_g_radio_4 g_g_radio_5  g_g_radio_6  \
0                    NaN  …          NaN         NaN          NaN
1                    NaN  …          NaN         NaN          NaN
2                   24.0  …          NaN         NaN          NaN
3                    NaN  …          NaN         NaN          NaN
4                    NaN  …          NaN         NaN          NaN

  g_g_radio_7 g_g_3_other_other_r_count g_g_3_othr_other_r g_g_9 g_g_10  \
0         NaN                       NaN                NaN   NaN    NaN
1         NaN                       NaN                NaN   NaN    NaN
2         NaN                       NaN                NaN   NaN    NaN
3         NaN                       NaN                NaN   NaN    NaN
4         NaN                       NaN                NaN   NaN    NaN

  hh_grid solar
0      No    No
1     Yes    No
2      No   Yes
3      No    No
4      No    No

[5 rows x 120 columns]
```

[40]:
```
click = alt.selection_multi(encodings = ['color'])

tiers = alt.Chart(kenya_mtf2_mini).mark_bar().encode(
    x = alt.X('elc_aggr_tier', axis = alt.Axis(title = "Tier")),
    y = 'count(elc_aggr_tier)',
    color = 'elc_aggr_tier',
    tooltip = 'elc_aggr_tier'
).properties(width = 400, selection = click).interactive()

lighting_source = alt.Chart(kenya_mtf2_mini).mark_bar(color='green').encode(
    x = alt.X('c_c_159', axis = alt.Axis(title='Of all the electricity sources␣
 ↪you mentioned above, which is the source that you use most of the time in␣
 ↪the household?')),
    y = 'count(c_c_159)',
    #color = 'c_c_159',
```

```
    tooltip = 'c_c_159'
).properties(width = 400, selection = click).transform_filter(click).
 ↪interactive()

tiers & lighting_source
```

[40]: alt.VConcatChart(…)

[3]:
```
locality = alt.Chart(kenya_mtf2_mini).mark_bar().encode(
    x = alt.X('locality_ur', axis = alt.Axis(title = "National Locality")),
    y = 'count(locality_ur)',
    color = 'locality_ur'
).properties(width = 200).interactive()

locality
```

[3]: alt.Chart(…)

[22]:
```
# Emily's comments

# As of December 2020, you can't make any pie charts in Altair, which is a bit␣
 ↪concerning:
# https://github.com/altair-viz/altair/issues/2148

# However, we can use matplotlib, plotly or dash as a workaround.

# I'm not sure if there are any things Altair can do that Tableau can't. To␣
 ↪create these charts in Altair, I'm working in
# Jupyter Notebooks. As such, there is a memory limit on the notebook (1GB). If␣
 ↪I were to try and load all 778 columns and
# 4,590 rows of data from the MTF, the notebook crashes. The charts made above␣
 ↪are used with select columns (120 in total)
# from the MTF. The smaller dataframe is workable in the notebook. I chose␣
 ↪those columns based on the AIP Module Mock-Ups.
# However, I am worried that users would not be able to fully explore the␣
 ↪breadth of the MTF (and the other variables) due
# to the memory issues of the notebook.

# From what I understand based on the Jam Board and team meetings, we want a␣
 ↪tool that allows the user to select variables and
# the type of visualization they'd like to see in a single screen -- not a␣
 ↪long, scrolling website. As I think more about the
# AIP project, it seems like we are trying to create something like Tableau␣
 ↪from scratch using MTF Data? I will have to explore
# Altair more, but I am unsure if the package can do something like that; based␣
 ↪on the documentation available, most of the
```

```
# visualizations are pre-made charts in which the user can interact with, but␣
 ↪does not provide the "pick your own visualization"
# functionality...
```

[ ]: