quiz1

January 23, 2023

0.1 The Gapminder bubble chart

We will use the Gapminder data that you are seen in lab and in the tutorials. I have kept the attributes that are relevant to this exercise

Column	Description
country	Country name
year	Year of observation
population	Population in the country at each year
region	Continent the country belongs to
sub_region	Sub-region the country belongs to
income_group	Income group as specified by the world bank in 2018
income	GDP per capita (in USD) adjusted for differences in purchasing power
children_per_woman	Average number of children born per woman

[2]:
```
# Run this cell to ensure that altair plots show up on gradescope
import altair as alt
import pandas as pd

# Handle large data sets without embedding them in the notebook
alt.data_transformers.enable('data_server')
# Include an image for each plot since Gradescope only supports displaying␣
 ↪plots as images
alt.renderers.enable('mimetype')


url = 'https://raw.githubusercontent.com/UofTCoders/workshops-dc-py/master/data/
 ↪processed/world-data-gapminder.csv'
# Read in the data using pandas
gm = pd.read_csv(url, parse_dates=['year'])

gm.head()
```

[2]:
```
       country       year  population region     sub_region income_group  \
0  Afghanistan 1800-01-01     3280000   Asia  Southern Asia          Low
1  Afghanistan 1801-01-01     3280000   Asia  Southern Asia          Low
2  Afghanistan 1802-01-01     3280000   Asia  Southern Asia          Low
```

```
3  Afghanistan 1803-01-01     3280000   Asia  Southern Asia          Low
4  Afghanistan 1804-01-01     3280000   Asia  Southern Asia          Low

   life_expectancy  income  children_per_woman  child_mortality  pop_density  \
0             28.2     603                 7.0            469.0          NaN
1             28.2     603                 7.0            469.0          NaN
2             28.2     603                 7.0            469.0          NaN
3             28.2     603                 7.0            469.0          NaN
4             28.2     603                 7.0            469.0          NaN

   co2_per_capita  years_in_school_men  years_in_school_women
0             NaN                  NaN                    NaN
1             NaN                  NaN                    NaN
2             NaN                  NaN                    NaN
3             NaN                  NaN                    NaN
4             NaN                  NaN                    NaN
```

0.1.1 Question 1

Filter the dataframe to only keep observations from the year 2018 and from countries in the region of Asia and assign this to a new variable name `gm_2018_asia`. Dates can be matched as strings when filtering. Finally show the first 15 rows of the dataframe.

[3]:
```
gm_2018_asia = gm.query("year == 2018 & region == 'Asia'")
# Print the top 15 rows of the data frame
gm_2018_asia.head(15)
```

[3]:
```
           country       year  population region          sub_region  \
218    Afghanistan 2018-01-01    36400000   Asia       Southern Asia
1532       Armenia 2018-01-01     2930000   Asia        Western Asia
2189    Azerbaijan 2018-01-01     9920000   Asia        Western Asia
2627       Bahrain 2018-01-01     1570000   Asia        Western Asia
2846    Bangladesh 2018-01-01   166000000   Asia       Southern Asia
4160        Bhutan 2018-01-01      817000   Asia       Southern Asia
5912      Cambodia 2018-01-01    16200000   Asia  South-eastern Asia
7226         China 2018-01-01  1420000000   Asia        Eastern Asia
9197        Cyprus 2018-01-01     1190000   Asia        Western Asia
12920      Georgia 2018-01-01     3910000   Asia        Western Asia
15767        India 2018-01-01  1350000000   Asia       Southern Asia
15986    Indonesia 2018-01-01   267000000   Asia  South-eastern Asia
16205         Iran 2018-01-01    82000000   Asia       Southern Asia
16424         Iraq 2018-01-01    39300000   Asia        Western Asia
16862       Israel 2018-01-01     8450000   Asia        Western Asia

       income_group  life_expectancy  income  children_per_woman  \
218             Low             58.7    1870                4.33
1532   Upper middle             76.0    8660                1.60
2189   Upper middle             72.3   16600                2.04
```

```
2627          High             77.2   44300               1.99
2846   Lower middle            73.4    3720               2.05
4160   Lower middle            74.8    9930               1.99
5912   Lower middle            69.3    3830               2.50
7226   Upper middle            76.9   16000               1.64
9197          High             80.8   32200               1.34
12920  Lower middle            74.3   10100               1.98
15767  Lower middle            69.1    6890               2.28
15986  Lower middle            72.0   11700               2.31
16205  Upper middle            76.5   17400               1.61
16424  Upper middle            68.0   15900               4.25
16862         High             82.4   33400               2.92

       child_mortality  pop_density  co2_per_capita  years_in_school_men  \
218              65.90         55.7             NaN                  NaN
1532             12.90        103.0             NaN                  NaN
2189             30.30        120.0             NaN                  NaN
2627              7.10       2060.0             NaN                  NaN
2846             32.00       1280.0             NaN                  NaN
4160             29.50         21.4             NaN                  NaN
5912             27.00         92.0             NaN                  NaN
7226              9.95        151.0             NaN                  NaN
9197              2.45        129.0             NaN                  NaN
12920            10.60         56.2             NaN                  NaN
15767            41.10        455.0             NaN                  NaN
15986            25.00        147.0             NaN                  NaN
16205            13.90         50.4             NaN                  NaN
16424            29.20         90.6             NaN                  NaN
16862             3.33        391.0             NaN                  NaN

       years_in_school_women
218                      NaN
1532                     NaN
2189                     NaN
2627                     NaN
2846                     NaN
4160                     NaN
5912                     NaN
7226                     NaN
9197                     NaN
12920                    NaN
15767                    NaN
15986                    NaN
16205                    NaN
16424                    NaN
16862                    NaN
```

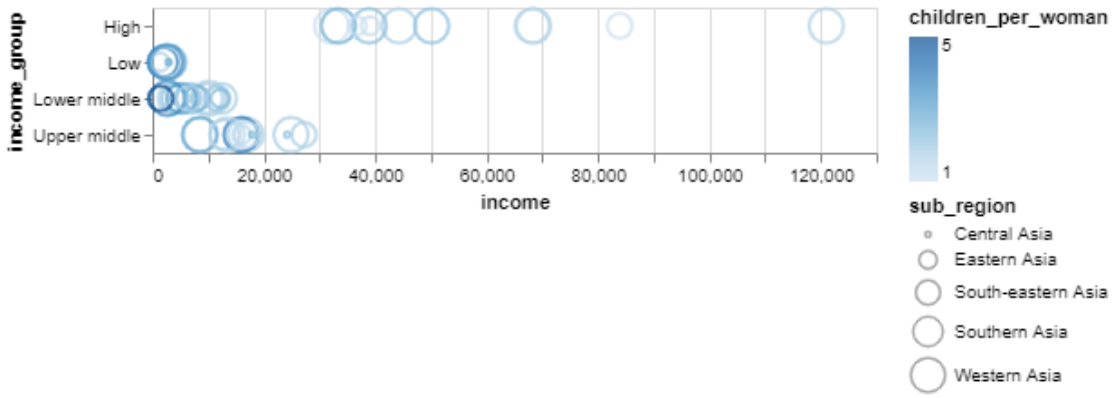
0.1.2 Question 2

Using the gm_2018_asia dataframe, create a bubble chart with the following

use the circle mark

```
<li> <code>income</code> using the vertical position channel (x) </li>
<li> <code>income_group</code> using the horizontal position channel (y) </li>
<li> <code>children_per_woman</code>using the color channel (note depending on the data attribu
<li> <code>sub_region</code>  using the size channel </li>
```

[30]:
```
chart_fert_money = alt.Chart(gm_2018_asia).mark_point().encode(
    x = "income:Q",# vertical position channel
    y = "income_group:O", # horizontal position channel
    color = "children_per_woman:Q",
    size = "sub_region:N"
)
# Show the plot
chart_fert_money
```

[30]:


0.1.3 Questions 3 - 6

Critique the visualization above by doing the following:

Describe the data attributes and the types used for each data

```
<li> Discuss the benefits of using certain channels to encode specific attributes</li>
<li> Name one question that the visualization is good at answering in its current state</li>
<li> Name one question that the visualization is not good at answering</li>
```

0.2 3 Describe the data attributes and the types used for each data

The data attributes here are `income`, `income_group`, `children_per_woman` ,and `sub_region`. Each of their attribute meaning is the following:

- sub_region Sub-region the country belongs to

- income_group Income group [as specified by the world bank in 2018]
- income GDP per capita (in USD) adjusted for differences in purchasing power
- children_per_woman Average number of children born per woman

The types of each are quantitative, ordinal, quantitative, nominal, in that order. This would make sense, since `income` consists of actual or real-value quantity of money people earned, `income_group` has natural groupings of levels, where High is top, and Low is min; `Children_per_woman` also could be expressed by real-valued quantity, lastly, `sub_region` consists of different categories, where there's no such who comes first so nominal would be its type.

0.3 4 Discuss the benefits of using certain channels to encode specific attributes

Using bubble chart with a known scale of axis could already explain information on income quite well, the use of `color` and `size` channel could help us ilustrate which group of people belonging to a sub_region gains more than others.

0.4 5 Name one question that the visualization is good at answering in its current state

Describing the groups of people income of Asia in the year of 2018

0.5 6 Name one question that the visualization is not good at answering

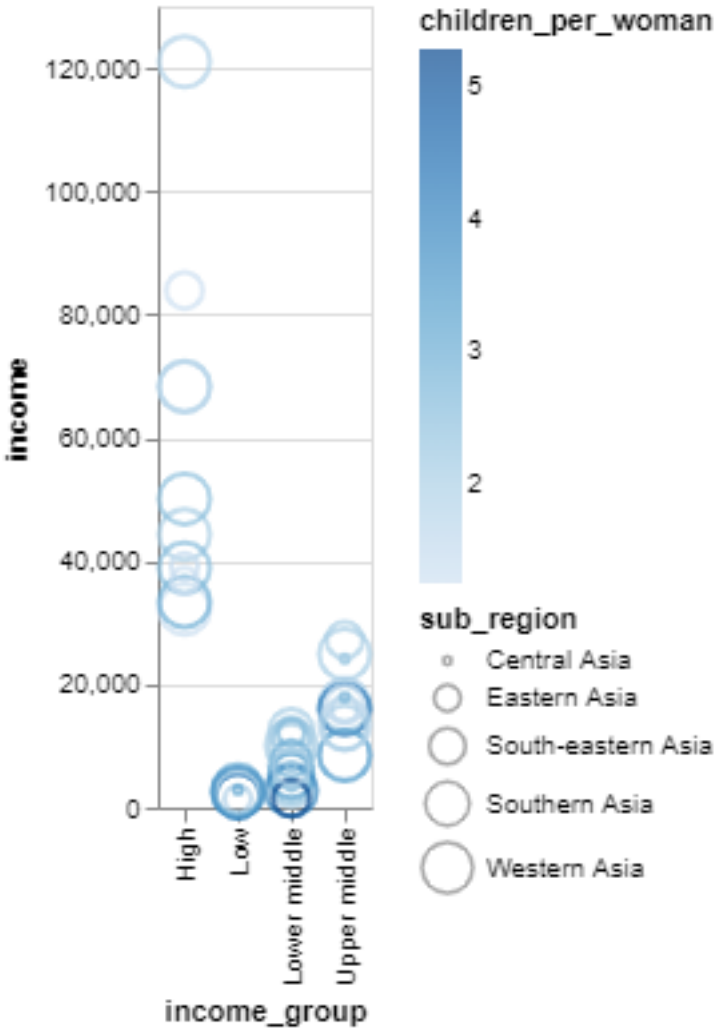
What are the chareteristics of people with most income? (And not effective order in ordinal channel), and the axis are inverted

0.5.1 Question 7

Using, your answer for the last question (i.e., Q6) redesign the bubble chart (from Q2) by changing which attribute each of the 4 given channels encode.

[29]:
```
my_chart = alt.Chart(gm_2018_asia).mark_point().encode(
    y = "income:Q",# vertical position channel
    x = "income_group:O", # horizontal position channel
    color = "children_per_woman:Q",
    size = "sub_region:N"
)
# Show the plot
my_chart
```

[29]:


children_per_woman
5
4
3
2
120,000
100,000
80,000
60,000
40,000
20,000
0
income
High
Low
Lower middle
Upper middle
income_group
sub_region
Central Asia
Eastern Asia
South-eastern Asia
Southern Asia
Western Asia