

Интерактивные графики с примерами Altair (Vega-Lite)

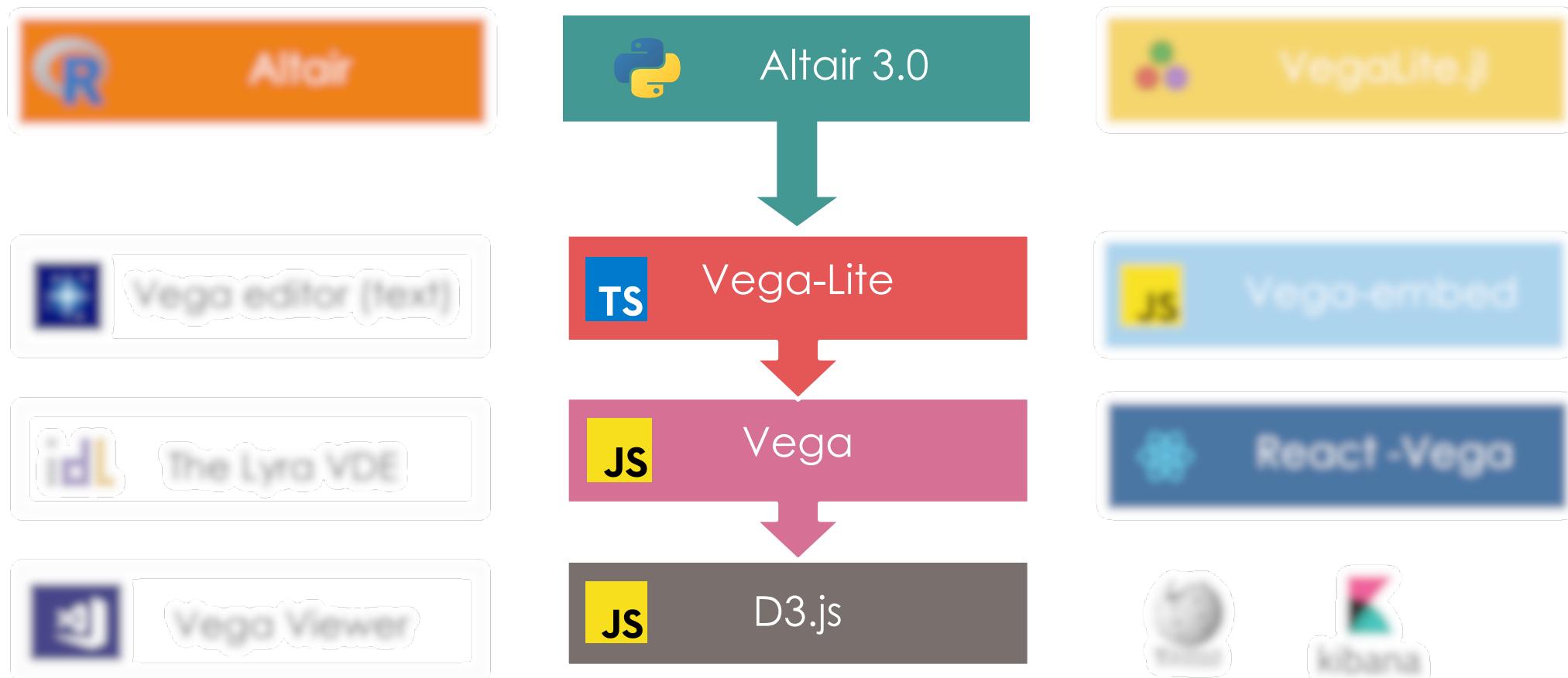
Илья Тимофеев, Старший партнер, ТТС Консалтинг



Supplementary materials (notebook, slides, links):

<https://iliatimofeev.github.io/DataFest2019Altair/>

Altair: Python API for Vega-lite Grammar of Interactive Graphics



Ilia Timofeev, Senior partner, TTS Consulting

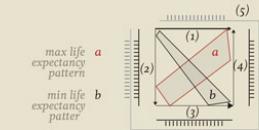
- I draw slides with charts on a daily basis since ... I remember myself
- I'm Altair User since 2017 and a bit contributor
- An author of GpdVega which integrates Altair and GeoPandas
- Also Vega/Vega-Lite tester

Further education = longer life

What is truly influencing life expectancy?
Could it be the GDP per capita? What about the size of the family and the level of education?
Is it better to live in an urban environment?
The visualization compares many different countries all over the world on these topics.

Sources: «Pocket World in Figures 2012 Edition» report of the magazine «The Economist»

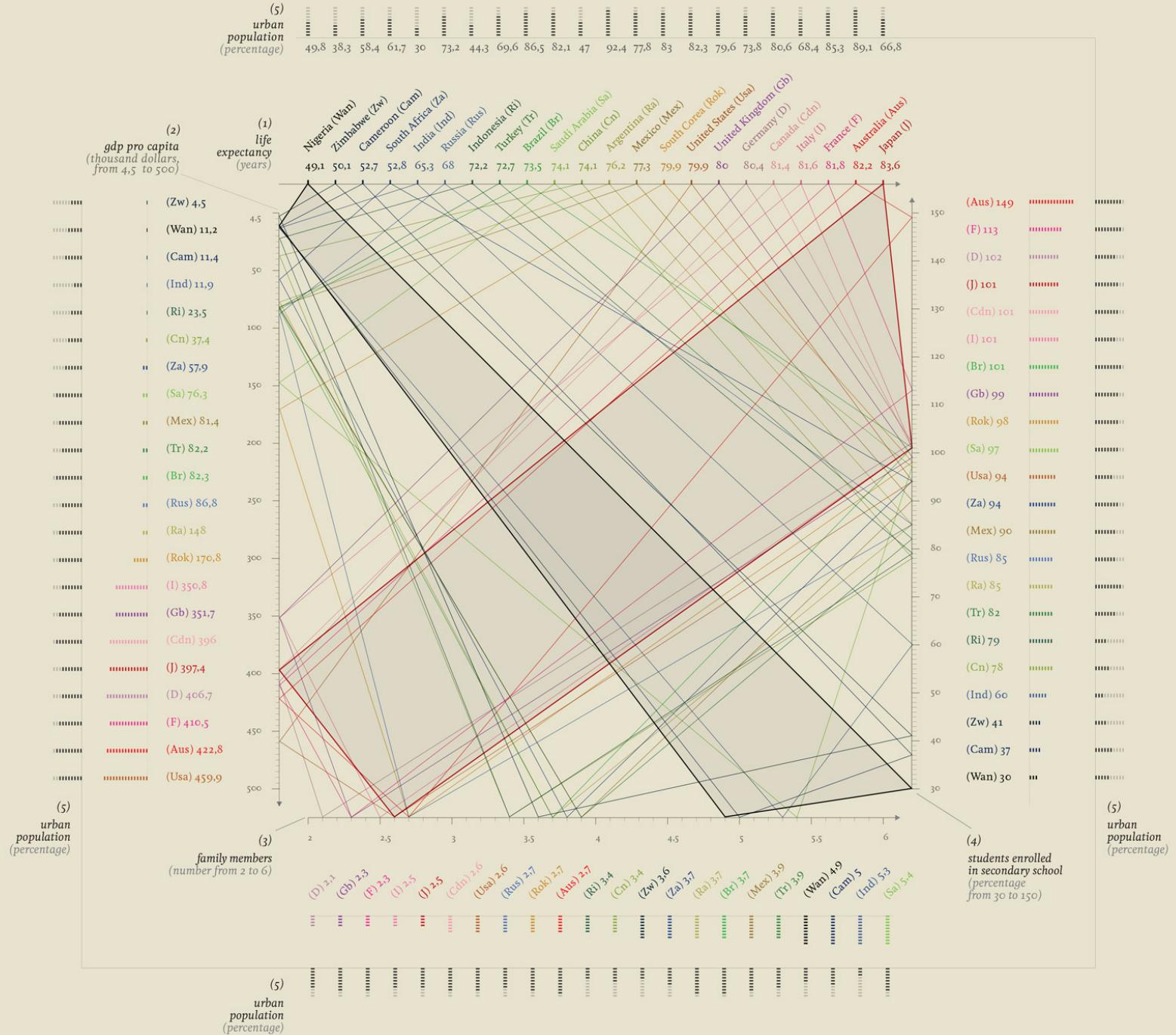
How to read it?



(1) life expectancy
life expectancy (estimated at birth date, combined genders). Each nation is represented with a colour within the ranking

(2) gdp pro capite
total value in dollars of goods and services annually produced from a nation divided by the number of its inhabitants

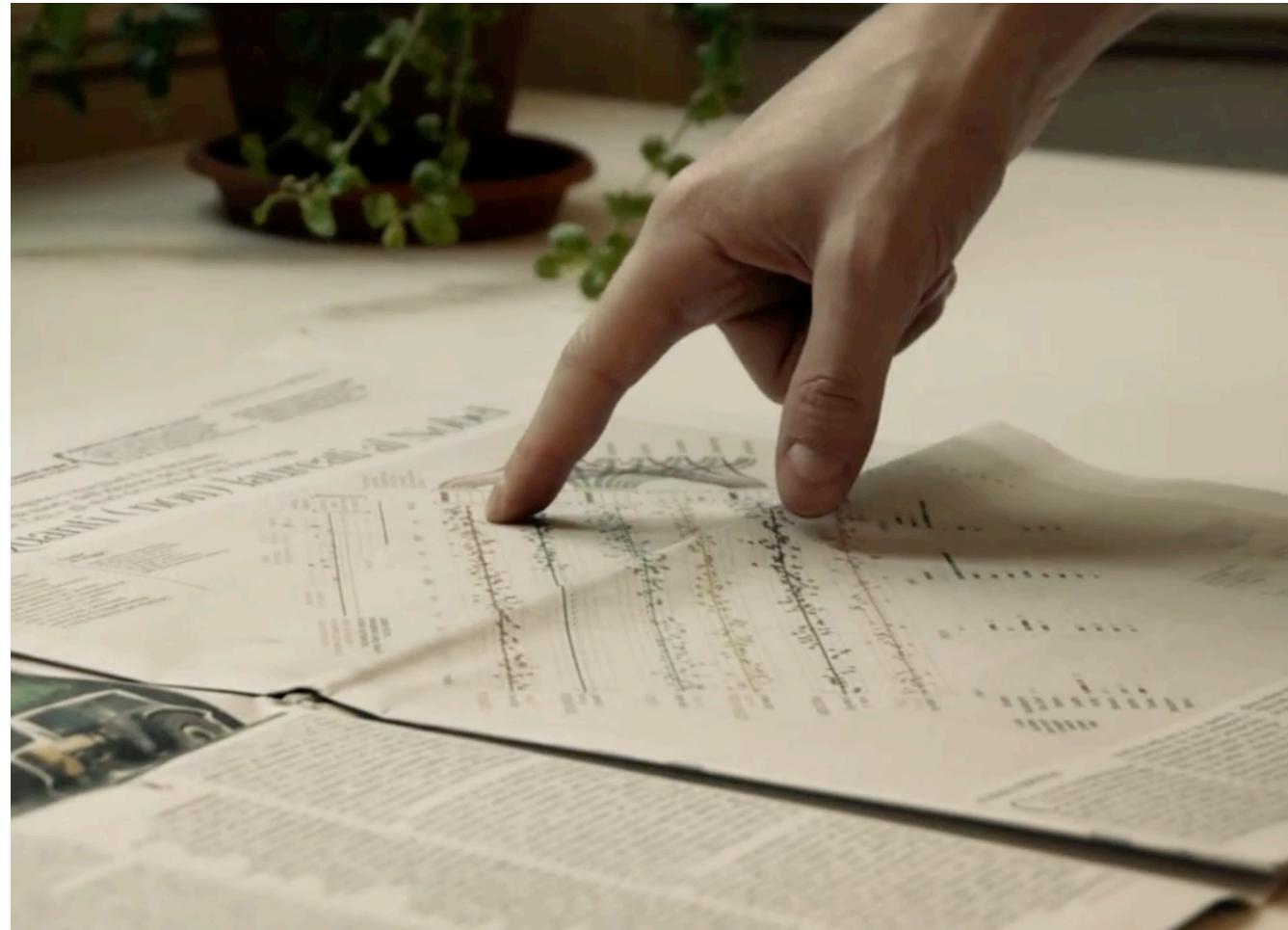
(4) registered at secondary school
percentage of students enrolled in secondary school. If part students belong to different age range, the percentage is above 100.



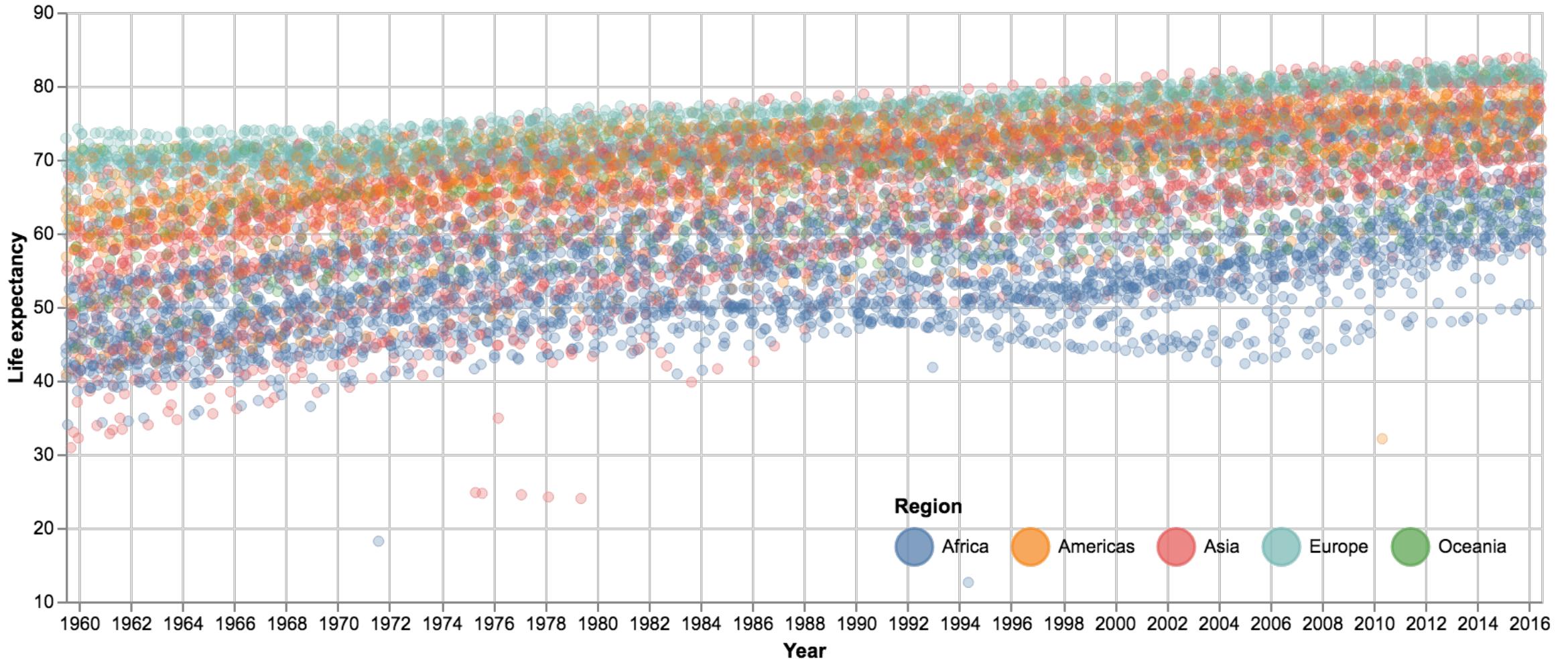
Interactive graphics with Altair

Agenda

- Visualization types
- Interactivity in EDA
 - First steps in Altair
- Vega ecosystem



Life expectancy at birth over time



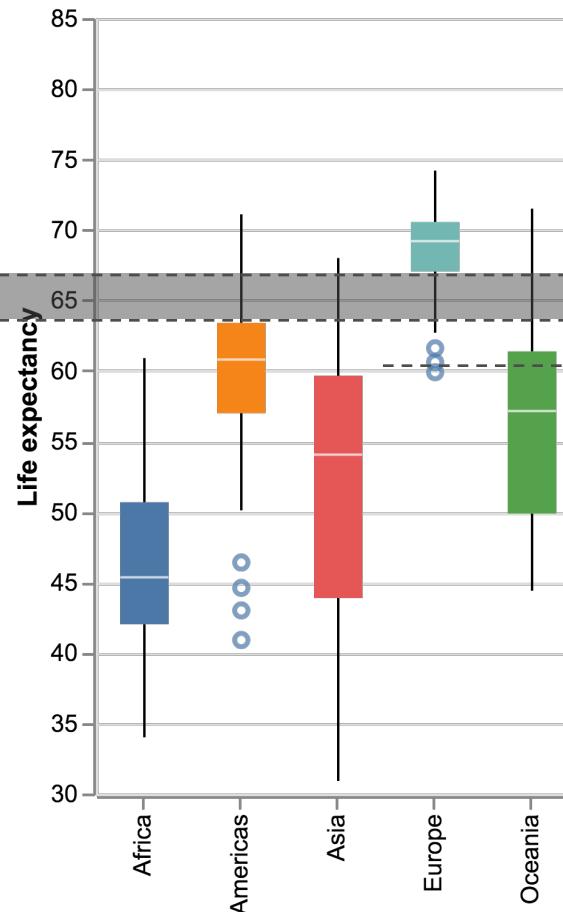
We are one world now

Europe

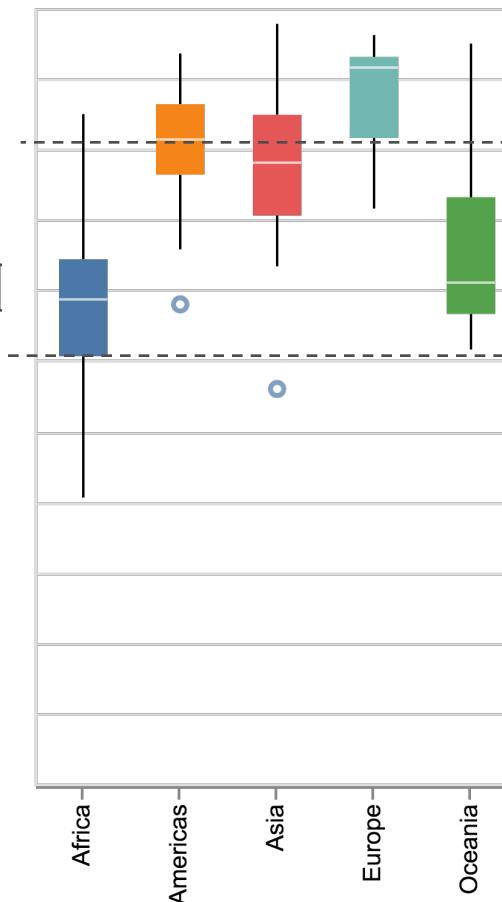
the gap

Other world

1960



2016



One world

no gap

Africa now

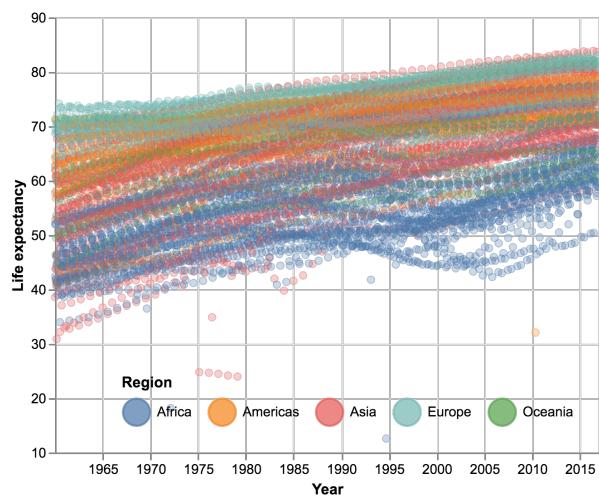
Longer than Moldova

Visualization specter

Question

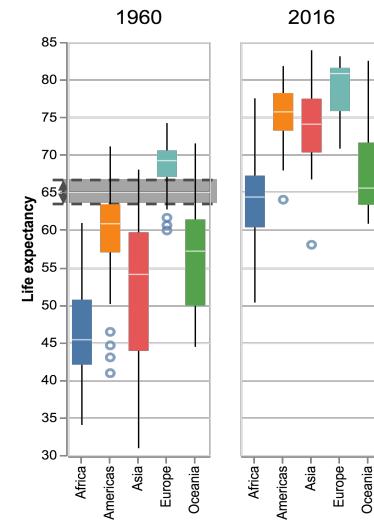


Exploration Data Analyses



Answer

Speaker slides

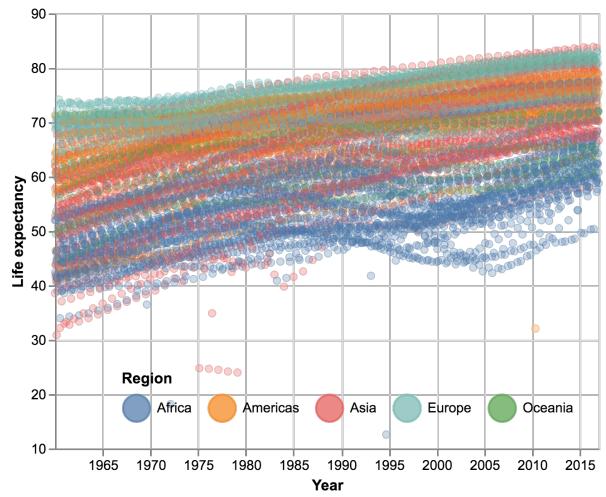


Visualization specter

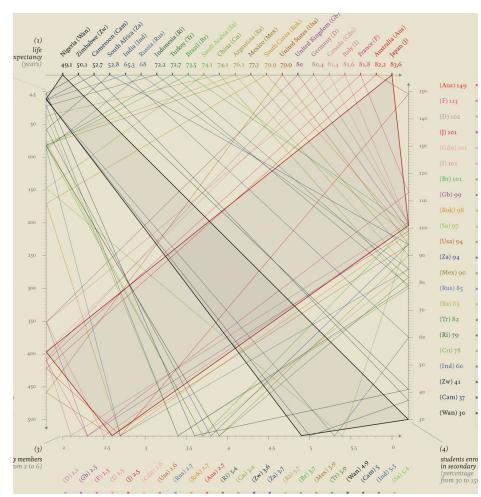
Question



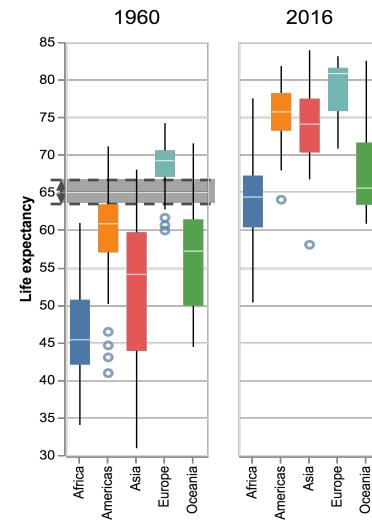
Exploration Data Analyses



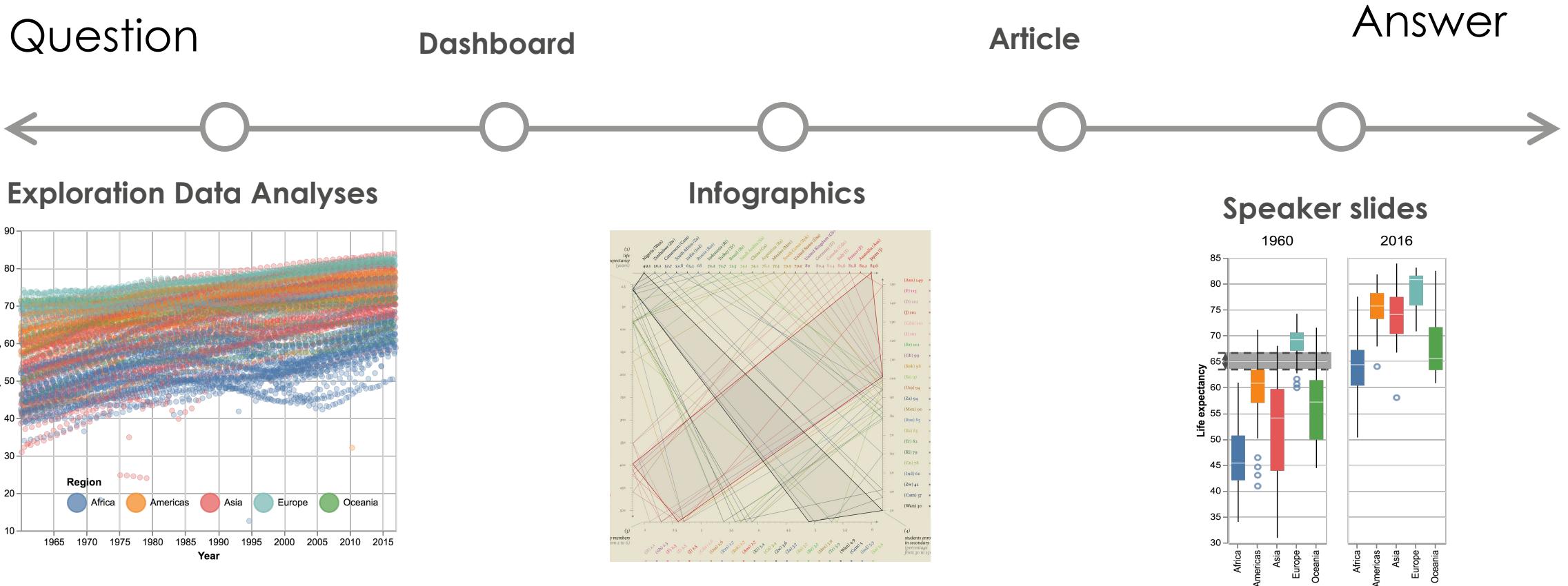
Infographics



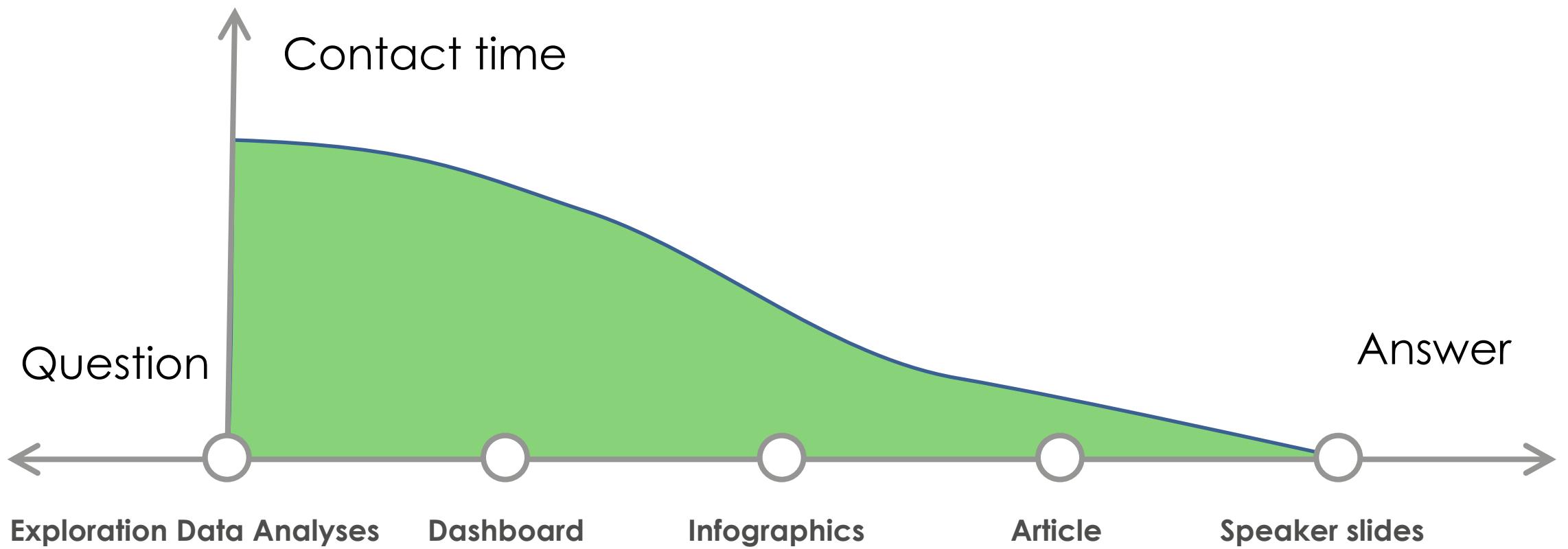
Speaker slides



Visualization specter



Time matters

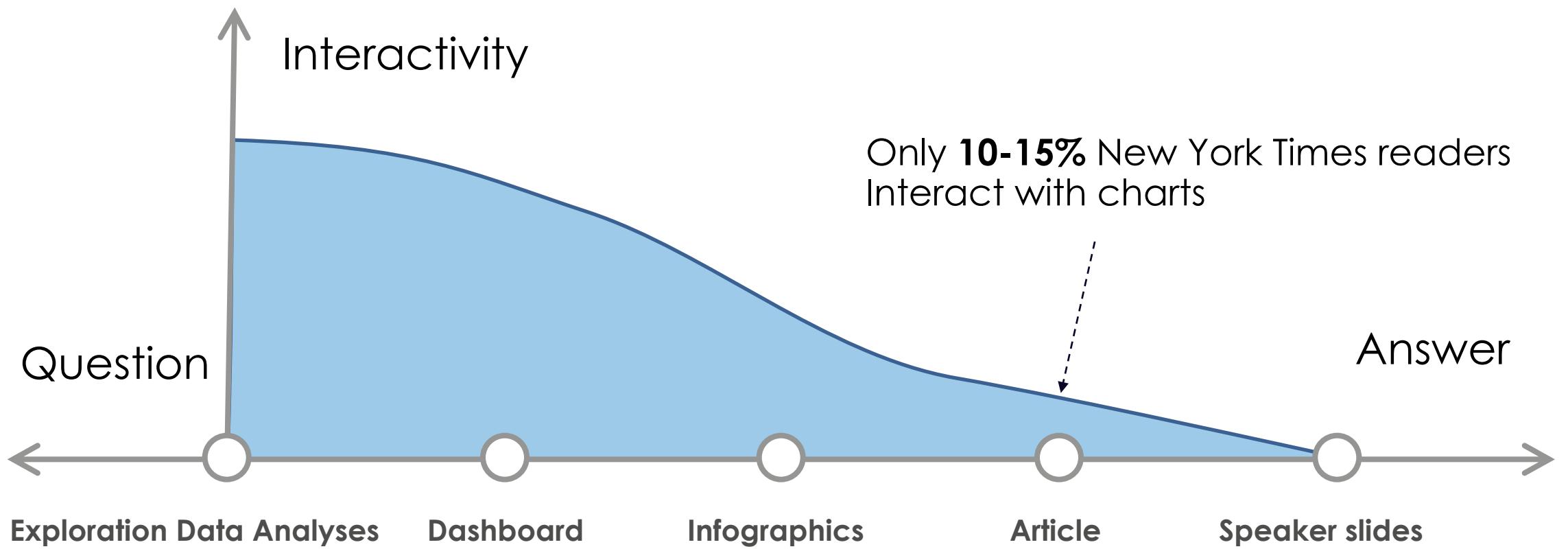


No interaction for speaker slides

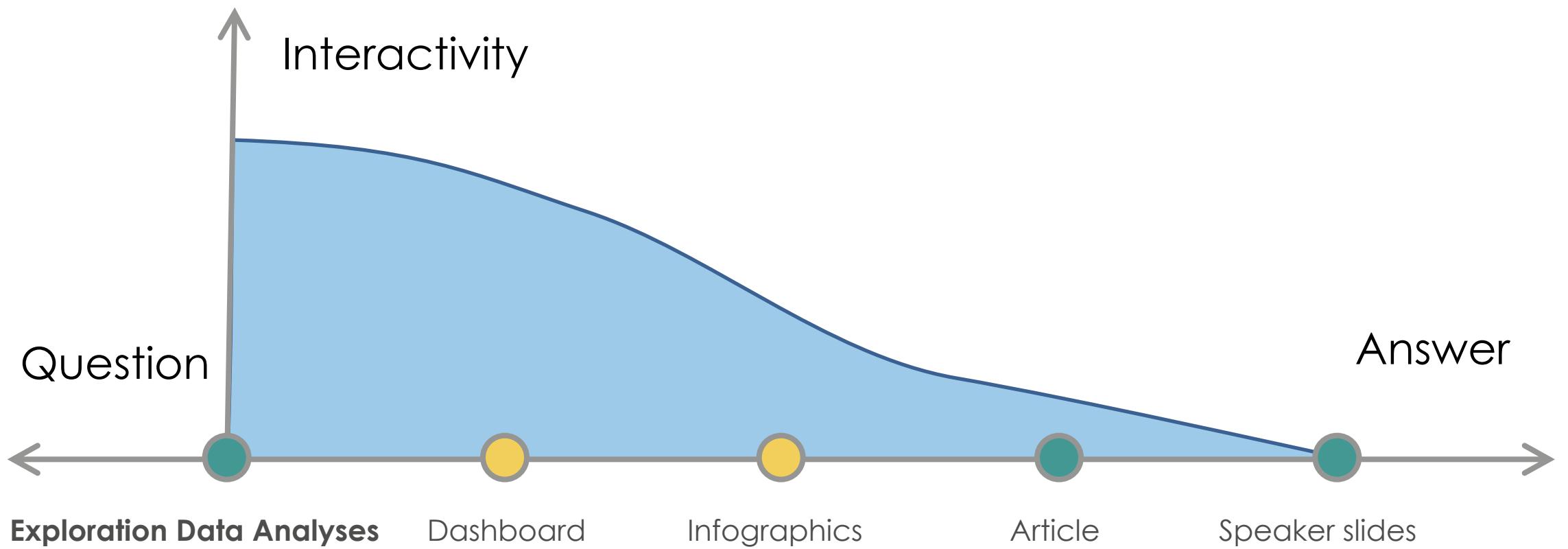
Hans Rosling several myths about world development [TED 2006](#)



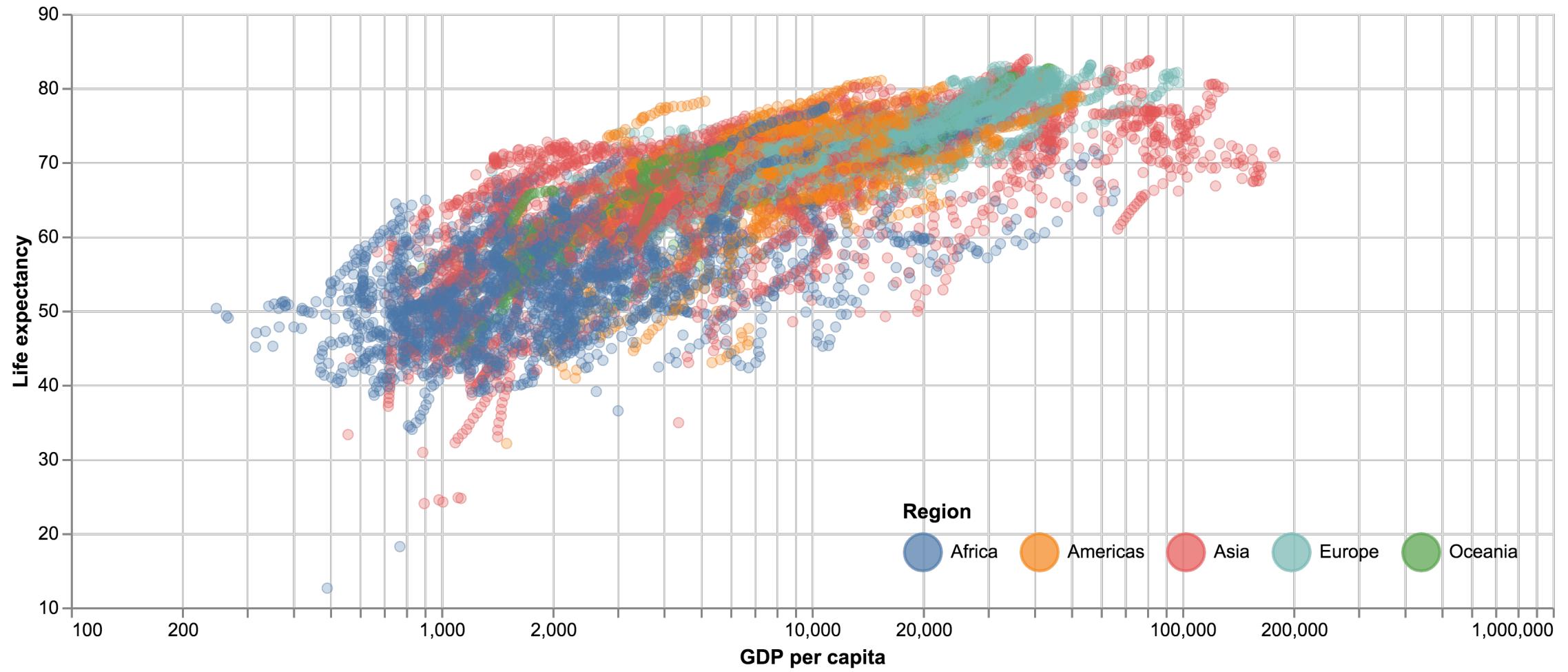
Interactivity application



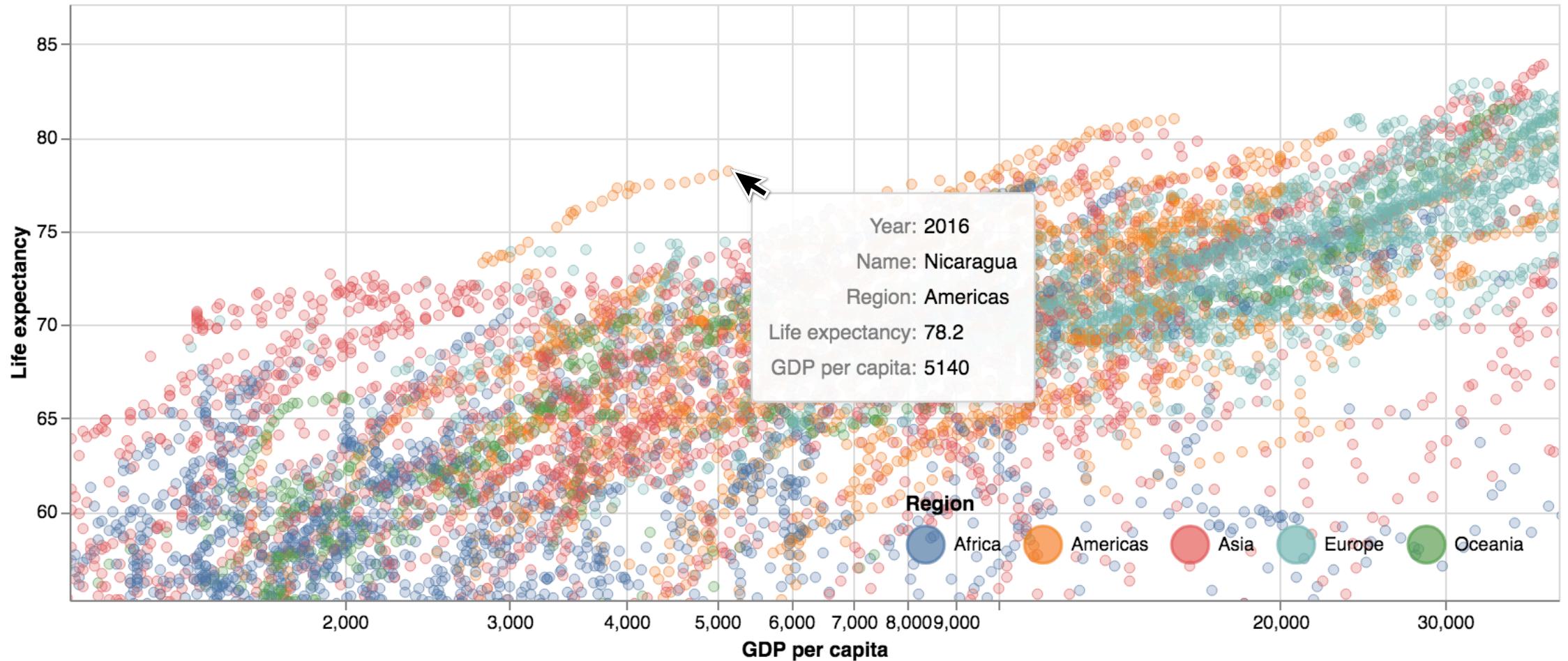
Altair application



EDA: Life expectancy at birth GDP per capita



EDA: Zoom and data point info



Data source: IHME through www.gapminder.org

First Step into Altair API

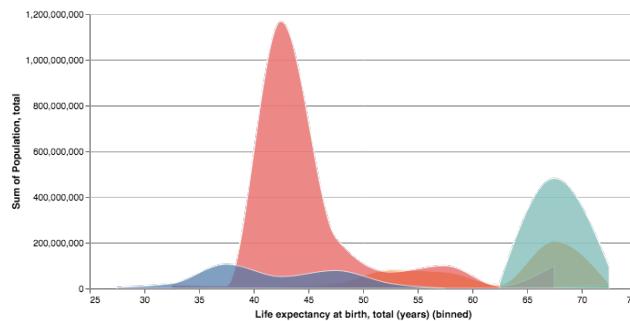
```
1 import pandas as pd  
2  
3 df_data.vgplot.scatter(  
4     x="GDP per capita",  
5     y="Life expectancy",  
6     c="Region"  
7 )
```

VS

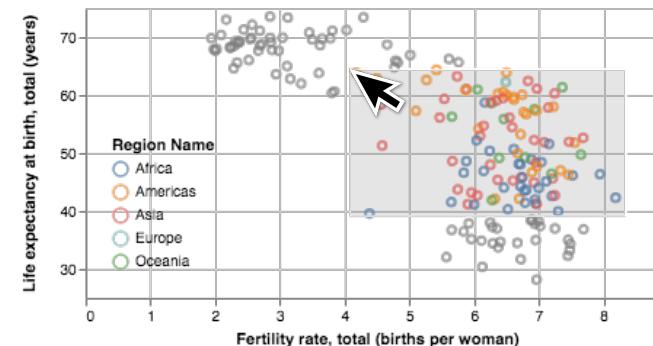
```
1 import altair as alt  
2  
3 (  
4     alt.Chart(data=df_data)  
5     .mark_point()  
6     .encode(  
7         x="GDP per capita",  
8         y="Life expectancy",  
9         color="Region"  
10    )  
11 )
```

Altair: Python API Grammar of Interactive Graphics

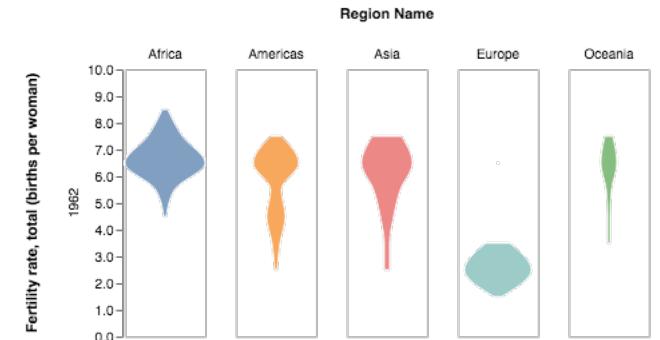
Grammar of Graphics



Grammar of interaction



View Composition Algebra

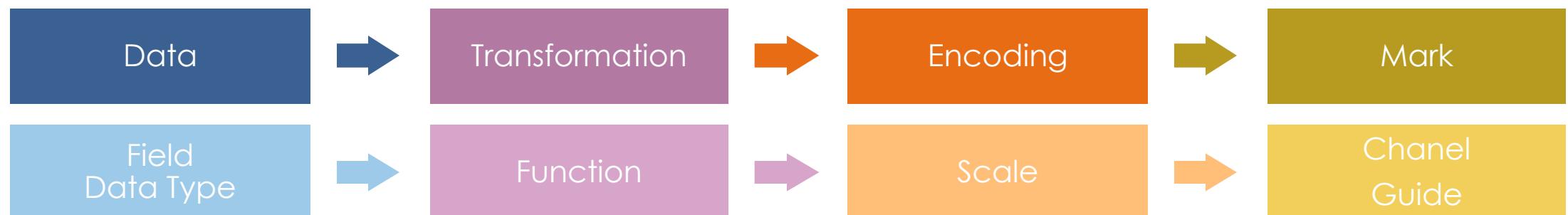


Vega-Lite presented by authors :

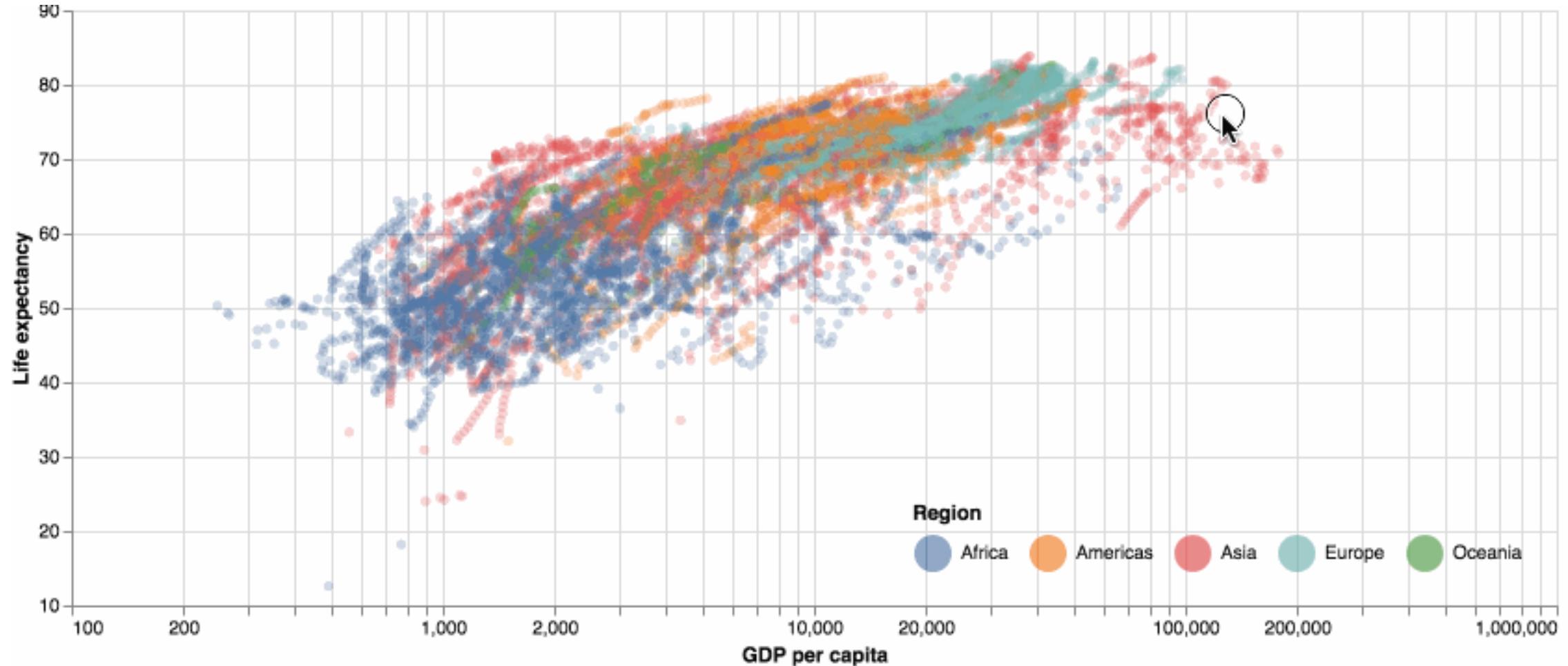
Vega Lite: A Grammar of Interactive Graphics - Wongsuphasawat, Moritz, and Satyanarayan [Open Viz Conf 2017](#)

Grammar of Graphics

```
1 chart = (
2     alt.Chart(data = df_data)
3     .mark_circle(fillOpacity=0.4)
4     .encode(
5         alt.X("GDP per capita:Q", scale=alt.Scale(type="log")),
6         alt.Y("Life expectancy:Q", scale=alt.Scale(zero=False)),
7         alt.Color("Region:N"),
8         tooltip=["Year:N", "Name:N", "Life expectancy:Q", "GDP per capita:Q"],
9     )
10    .interactive()
11 )
12 chart
```

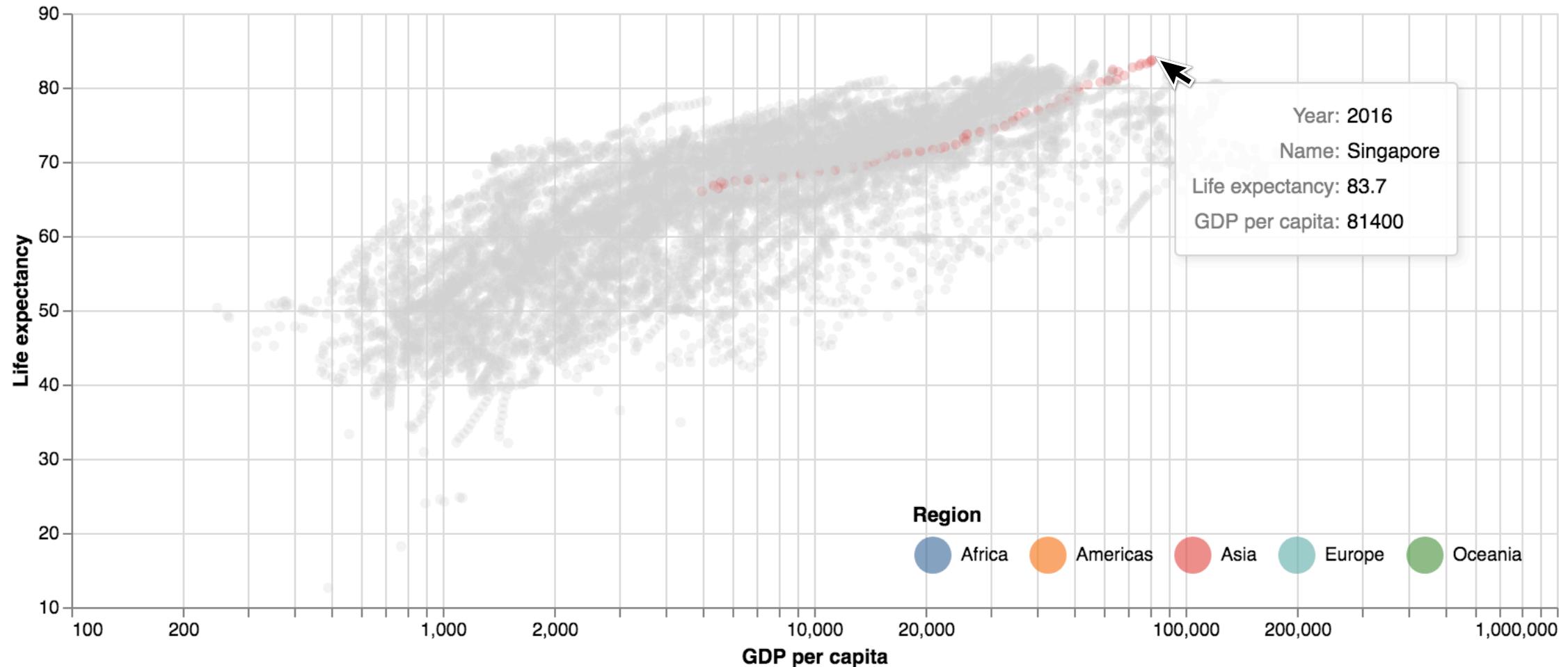


EDA: Zoom and pan + data point info



Data source: [IHME](#) through [www.gapminder.org](#)

EDA: Highlight a hidden class

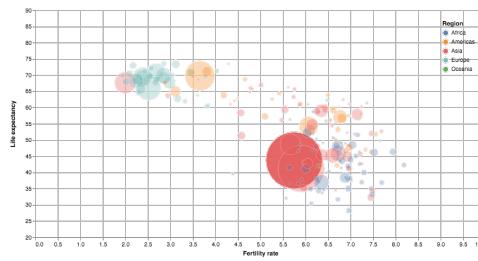


Data source: [IHME](#) through [www.gapminder.org](#)

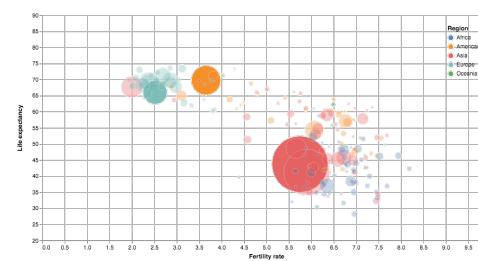
Grammar of interaction

Selection Types

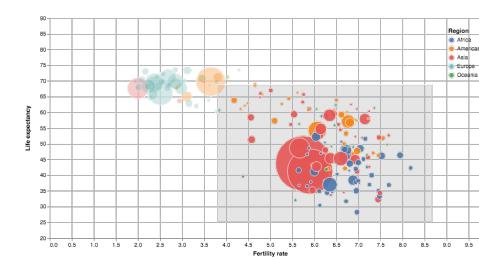
Single



Multi



Interval



Event

Encoding

Fields

Selection

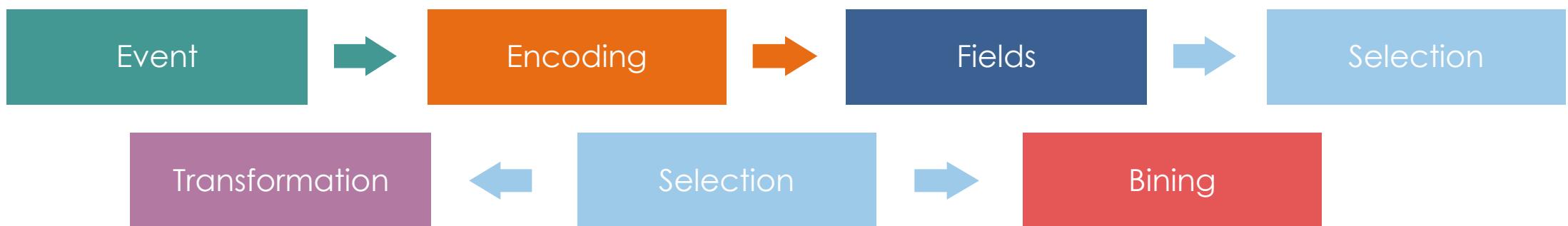
Transformation

Selection

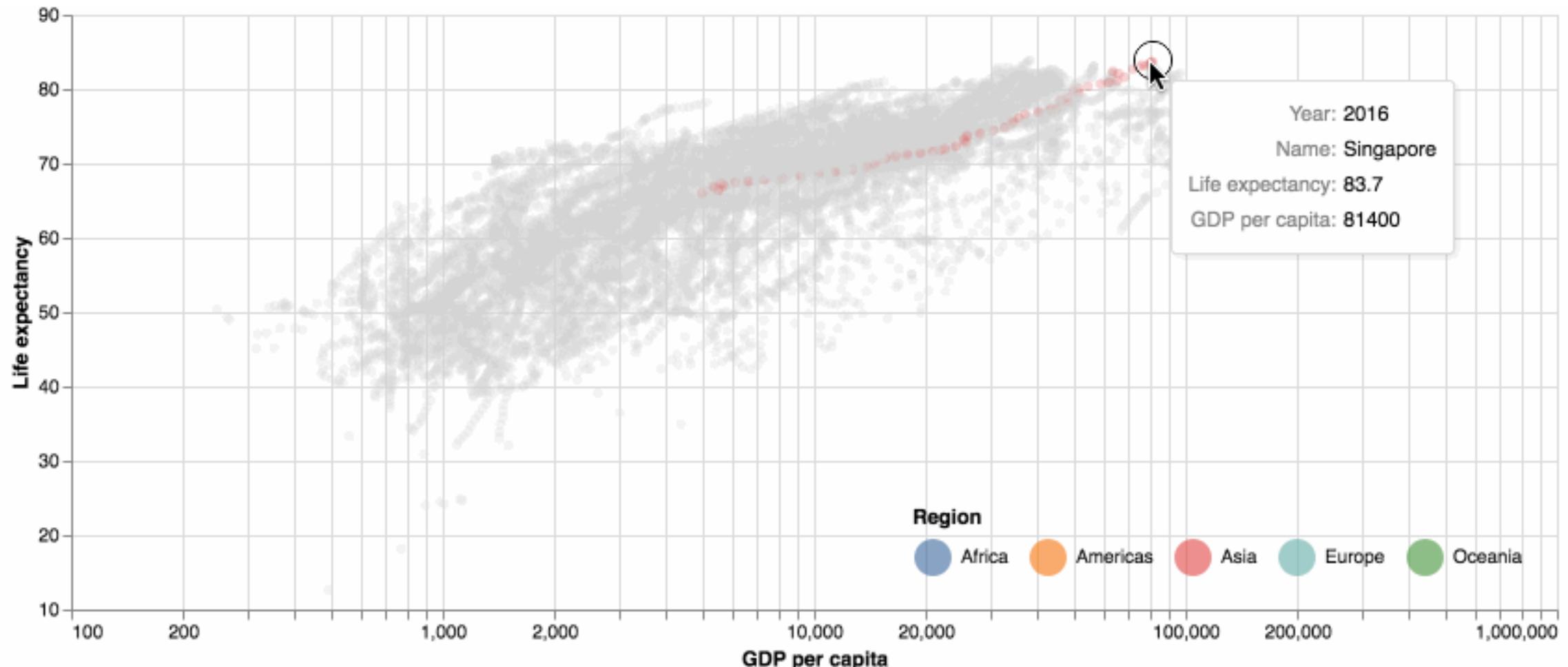
Bining

Grammar of interaction

```
1 sel_country = alt.selection_multi(fields=["Name"])
2 country_chart = (
3     chart
4     .add_selection(sel_country)
5     .encode(
6         color=alt.condition(sel_country, "Region:N", alt.value("lightgray")),
7     )
8
9 )
10
11 country_chart
12
```

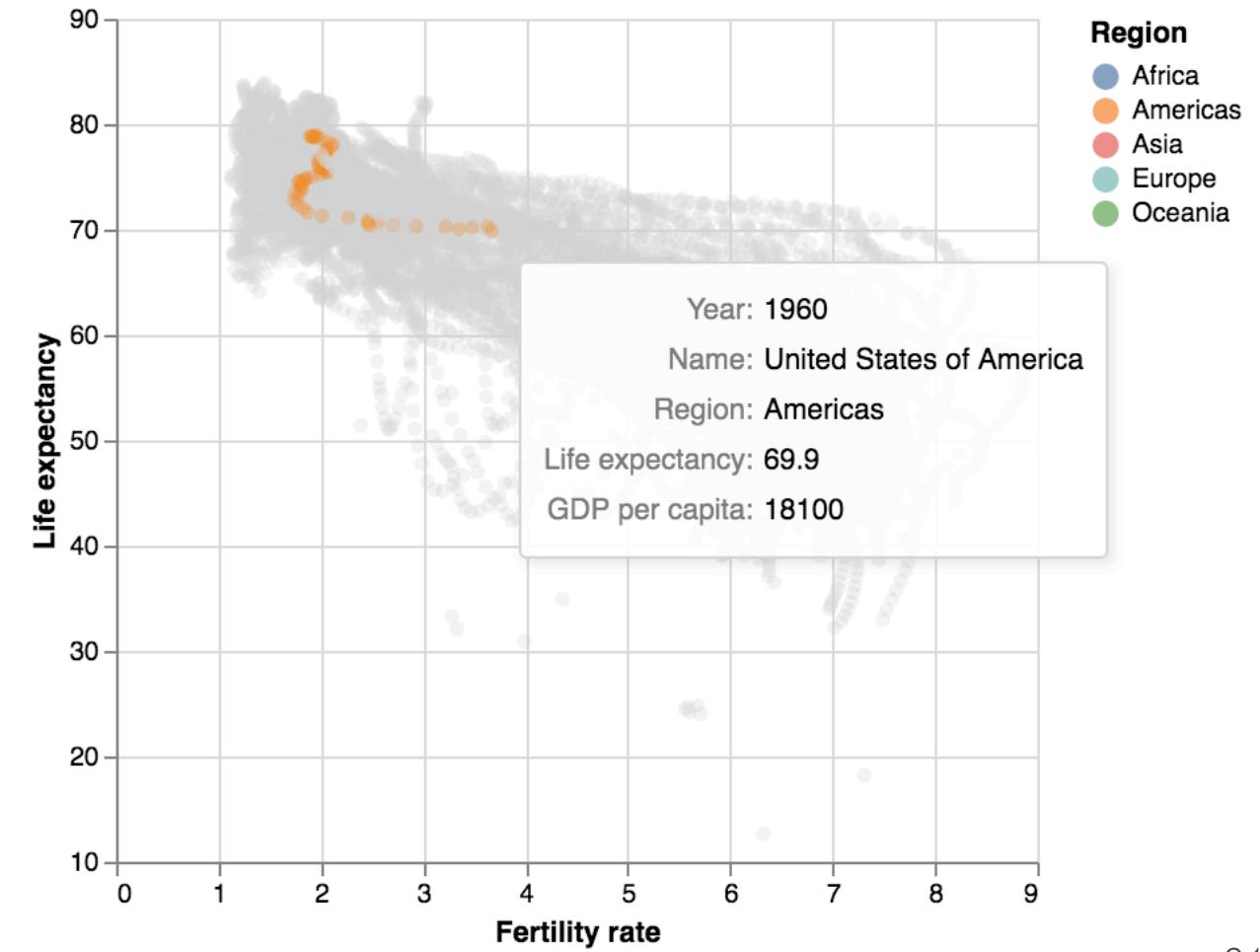
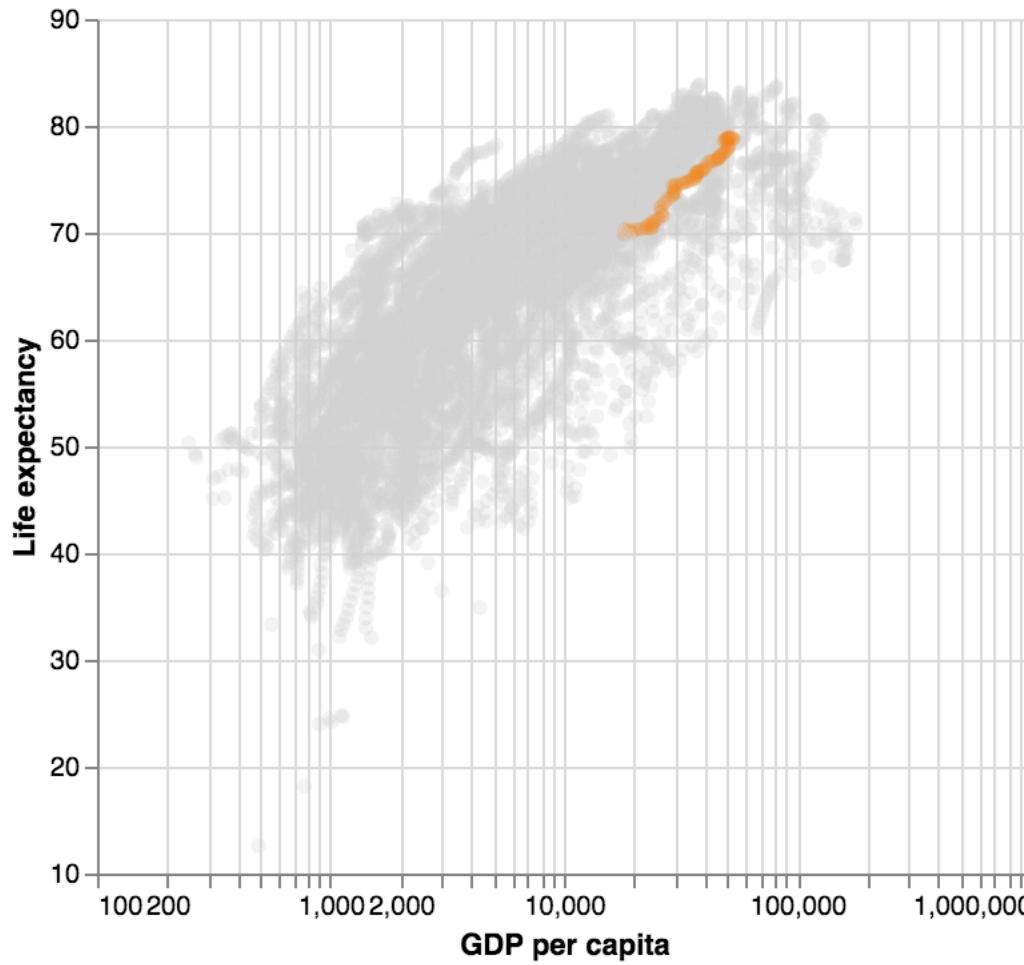


EDA: Highlight a hidden class



Data source: [IHME](#) through [www.gapminder.org](#)

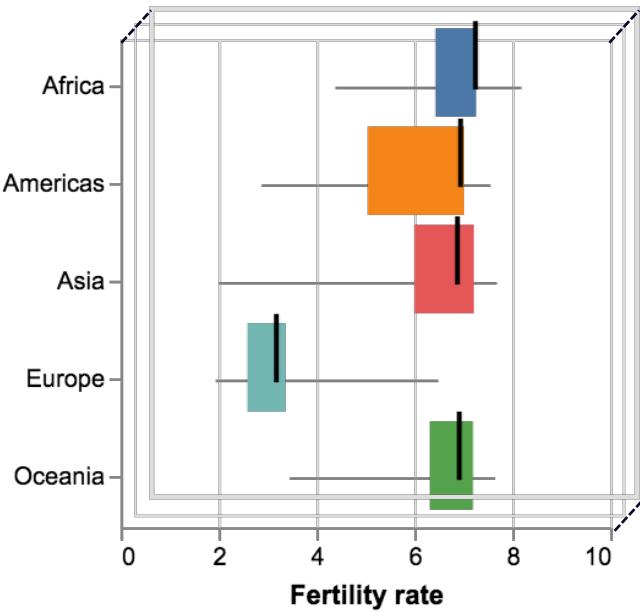
EDA: Multifactor analyses



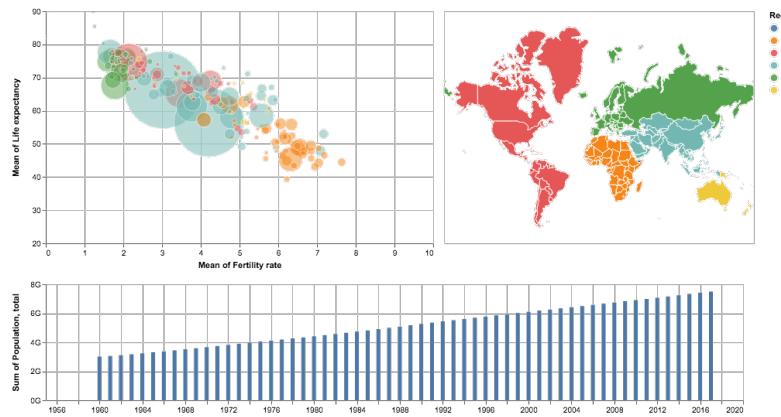
Data source: [IHME](#) through [www.gapminder.org](#)

View Composition Algebra

Layer

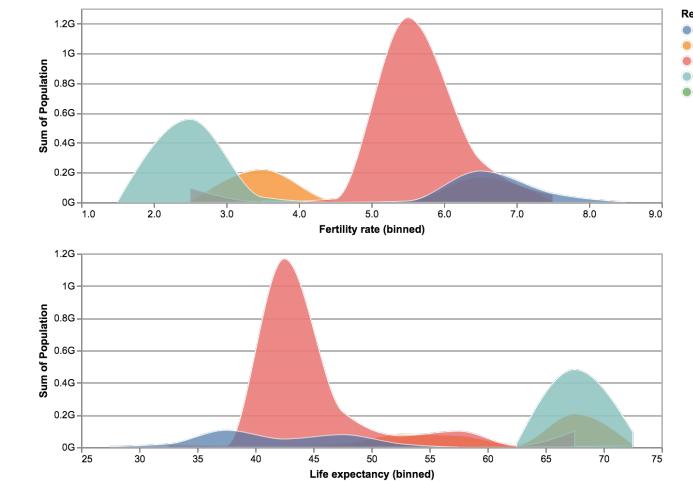


Concat



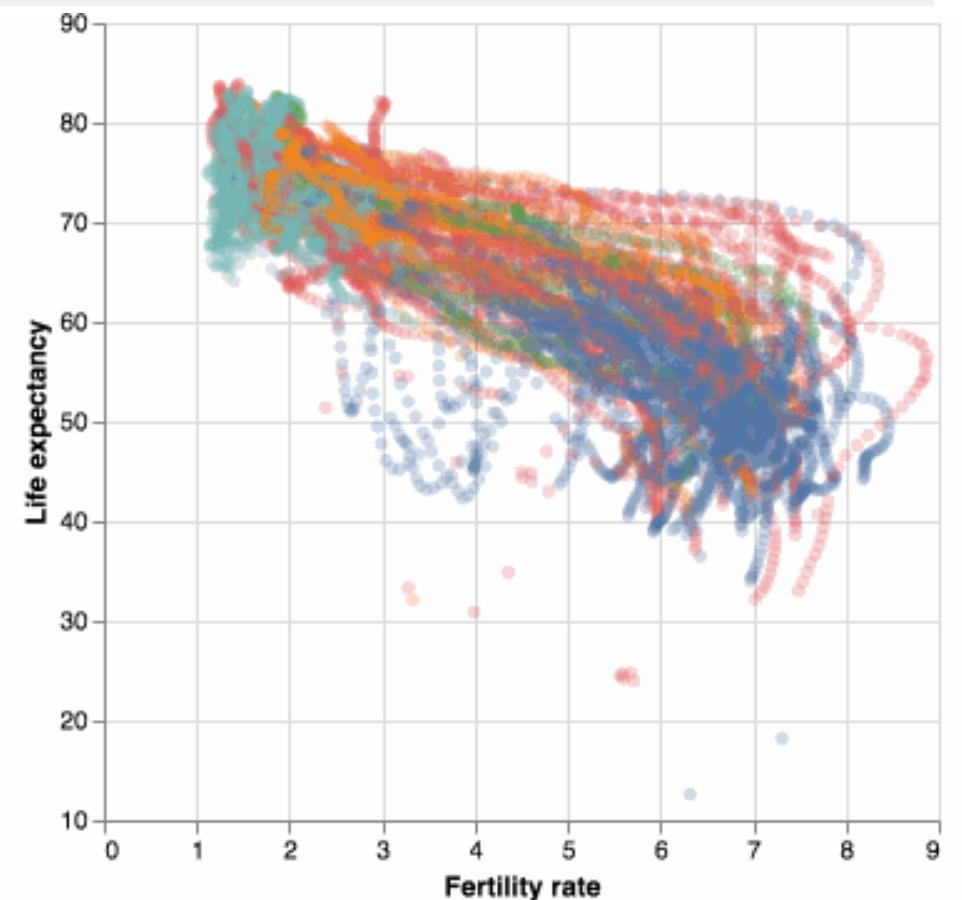
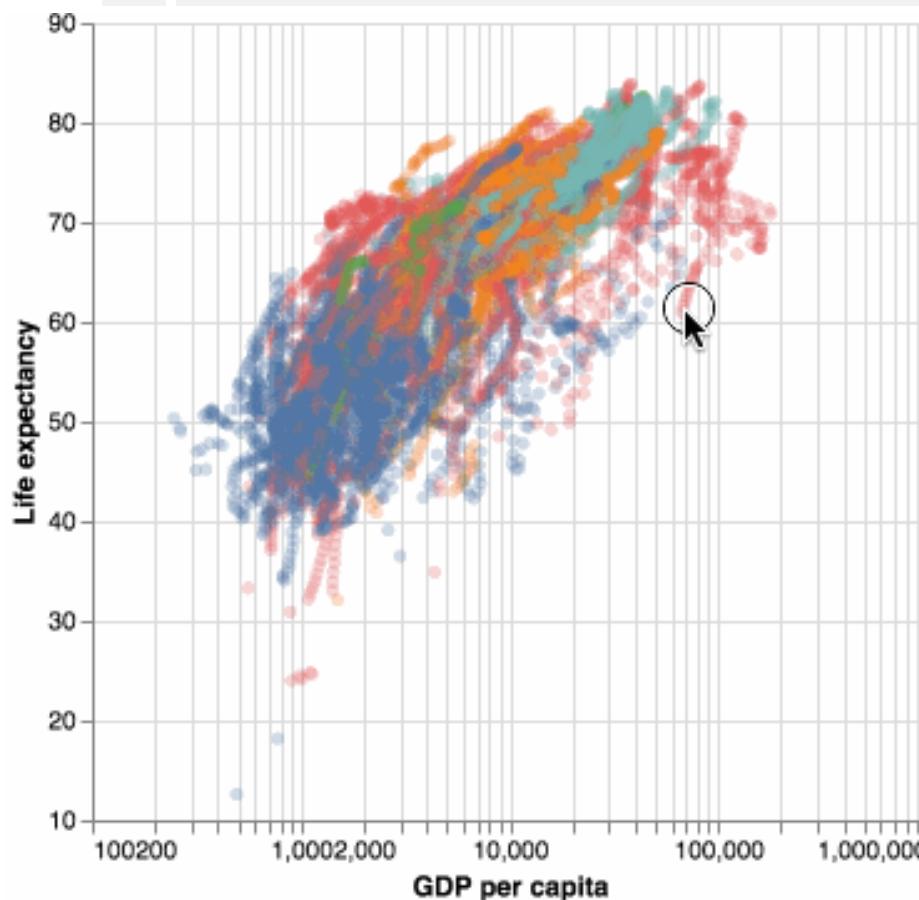
Resolve

Facet/Repeat



EDA: Multifactor analyses

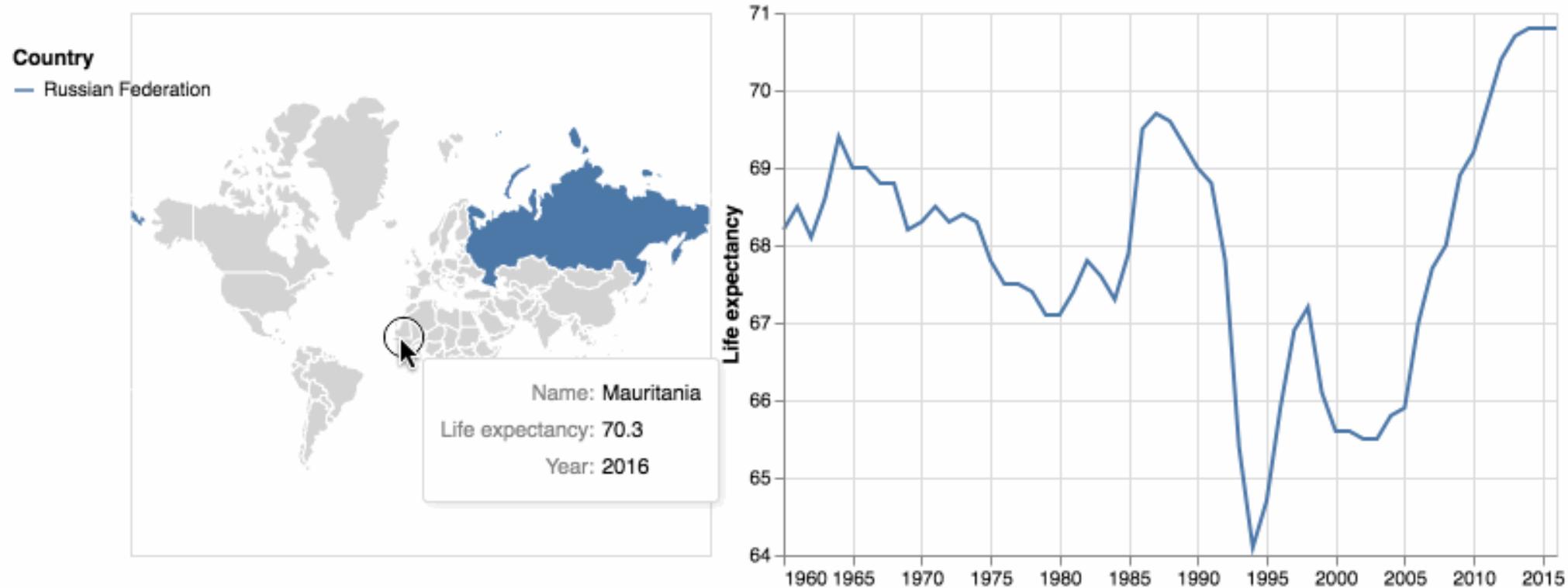
```
1 country_chart = country_chart.properties(width=335, height=320)  
2 country_chart | country_chart.encode(x="Fertility rate:Q")
```



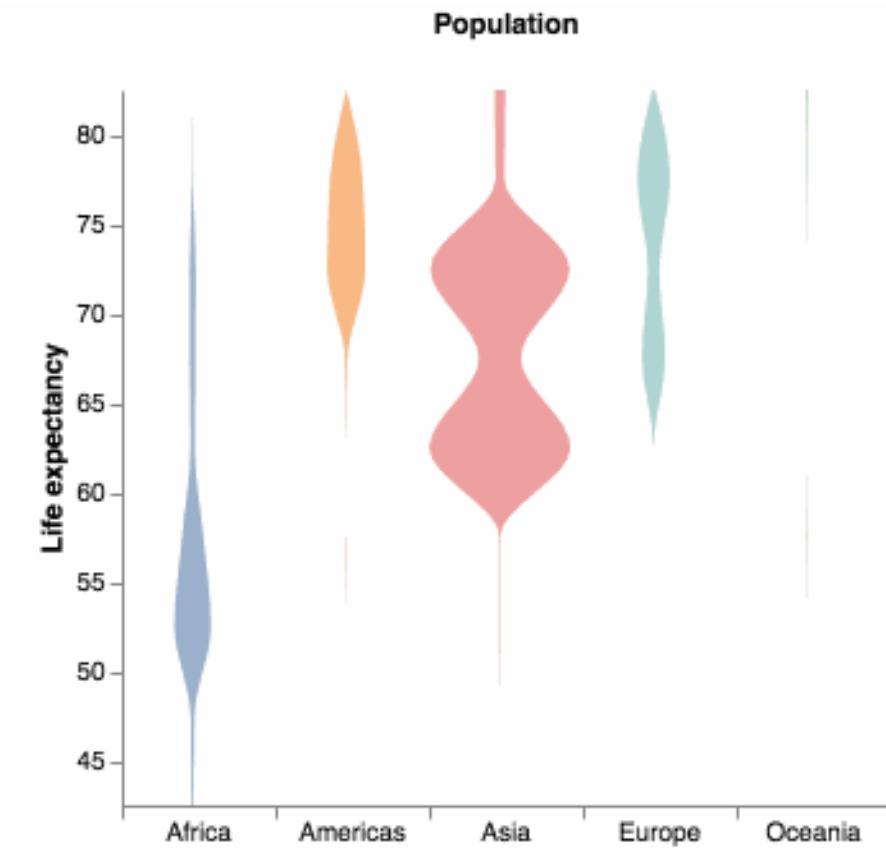
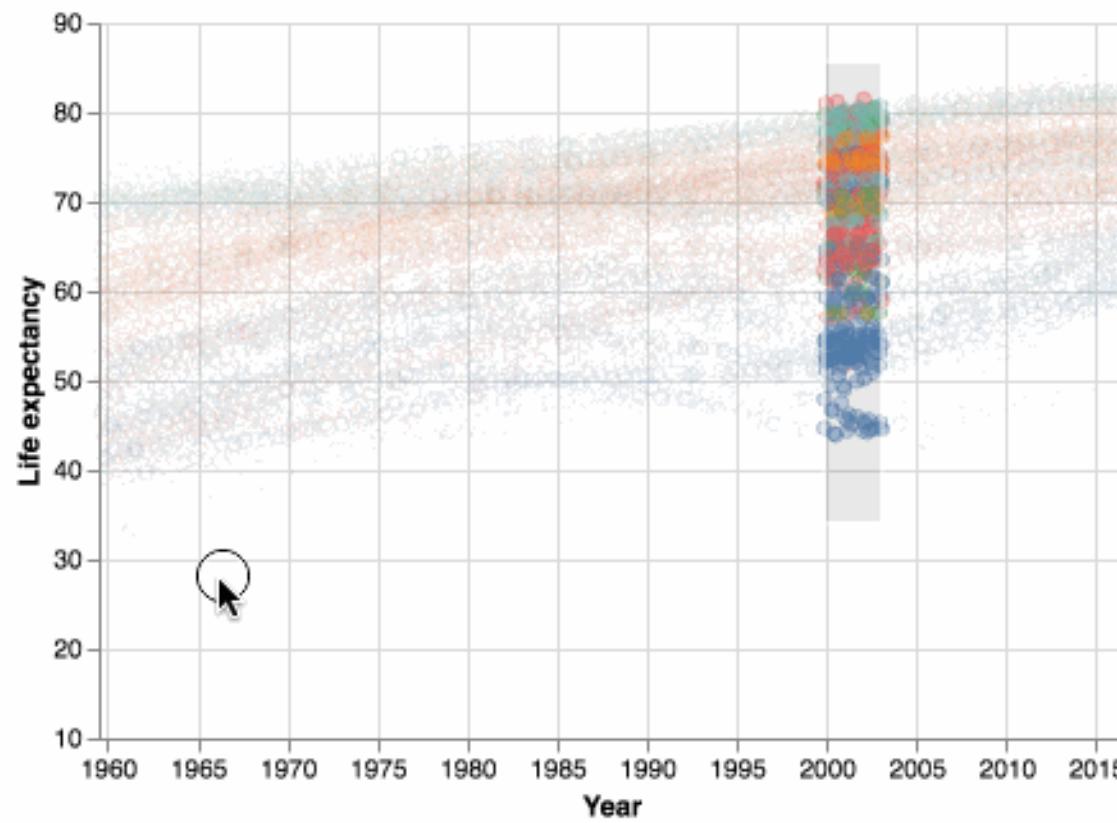
Region

- Africa
- Americas
- Asia
- Europe
- Oceania

EDA: Details View

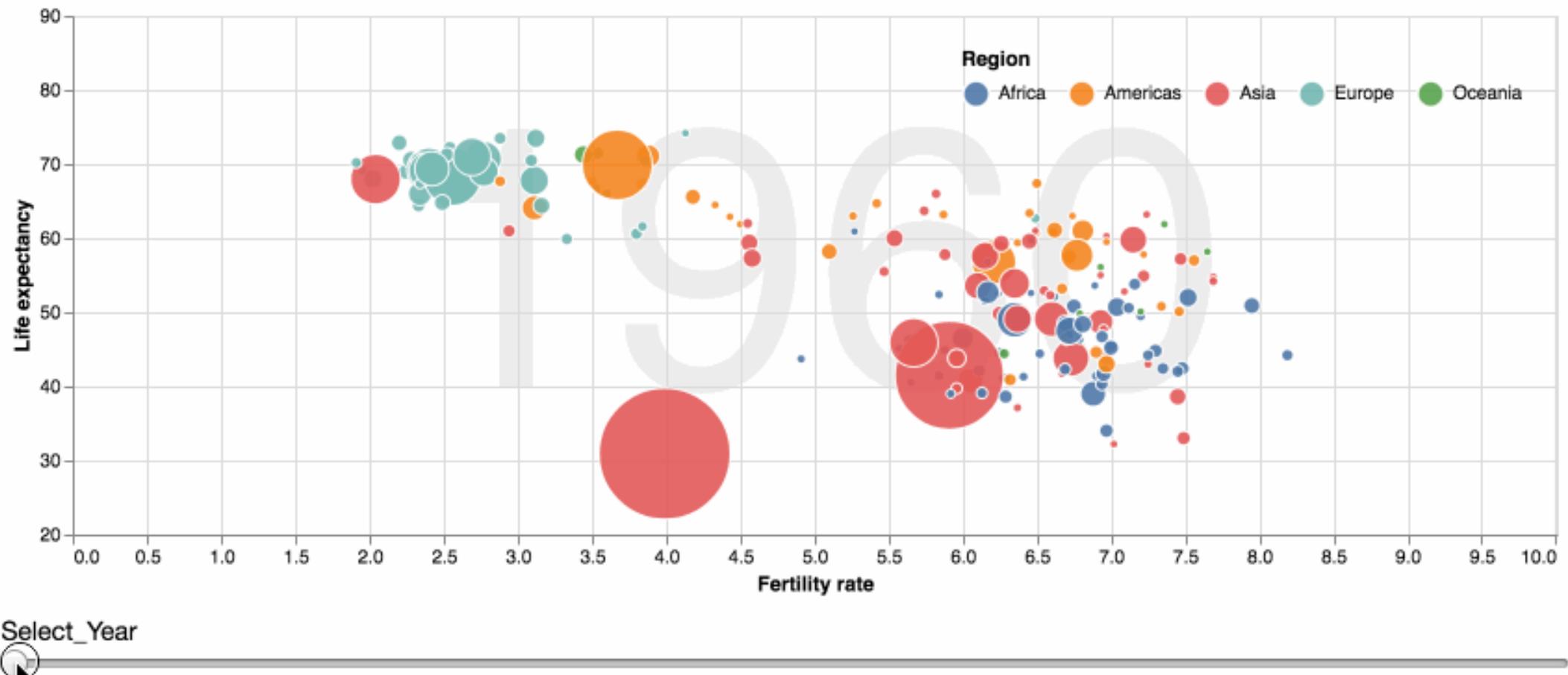


EDA: Aggregate



Data source: IHME through www.gapminder.org

EDA: Dynamic filtering

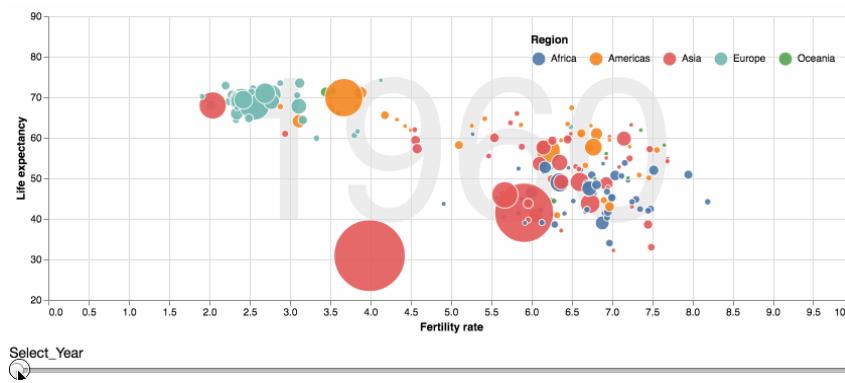
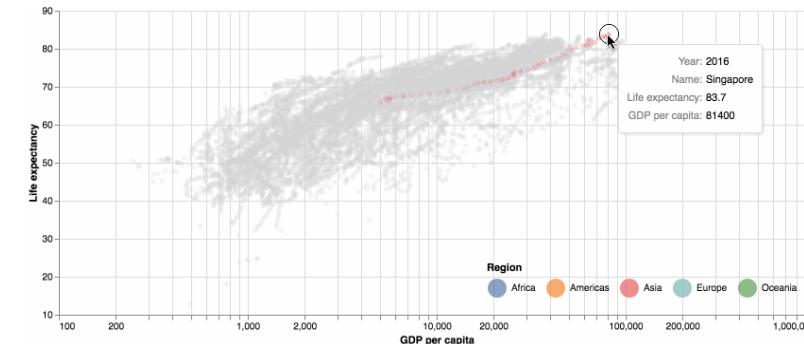


Reconstruction of Hans Rosling's chart on TED 2006

Data source: [IHME](#) through [www.gapminder.org](#)

Basic tools

1. Zoom and pan
2. Data point info
3. Highlight a hidden class



Advanced exploration

1. Details
2. Aggregate
3. Dynamic filtering

Altair: first difficulties

```
1 import altair as alt
2
3 df_data = pd.read_csv("gapmider_data.csv")
4 (
5     alt.Chart(data=df_data)
6     .mark_point()
7     .encode(x="GDP per capita", y="Life expectancy", color="Region")
8 )
```

`MaxRowsError: The number of rows in your dataset is greater than the maximum allowed (5000).
For information on how to plot larger datasets in Altair, see the documentation`

```
Chart({
  data:      Year  Child mortality  Fertility rate  GDP per capita  Life expectancy  \
M49
  4    1960            364.0          7.45        1210.0          38.6
  4    1961            358.0          7.45        1200.0          39.4
  4    1962            352.0          7.45        1200.0          40.1
```

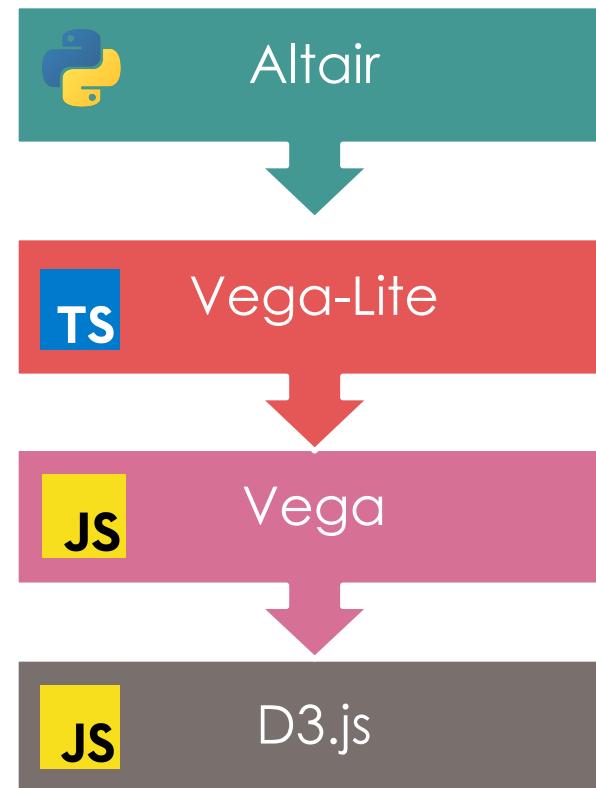
How it works?

Declarative statistical visualization library for Python

High-level grammar for interactive graphics

Visualization grammar
higher-level visualization specification language

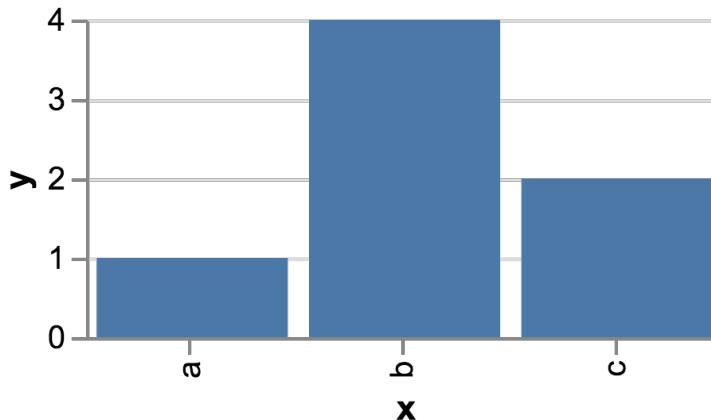
JavaScript library for manipulating Data-Driven Documents



```

1 df = pd.DataFrame({
2     "x": ["a", "b", "c"],
3     "y": [1, 4, 2]
4 })
5 chart = (
6     alt.Chart(data=df)
7     .mark_bar()
8     .encode(
9         x="x",
10        y="y"
11    )
12 )
13 chart

```



Vega – lite grammar for interactive graphics (JSON)

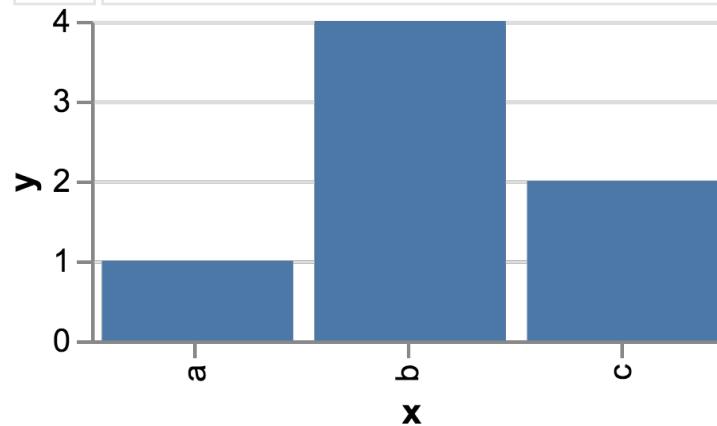
```

1 {
2     "config": {"view": {"width": 400, "height": 200}},
3     "data": {"name": "data-9334"},
4     "mark": "bar",
5     "encoding": {
6         "x": {"field": "x", "type": "nominal"},
7         "y": {"field": "y", "type": "quantitative"}
8     },
9     "datasets": {
10        "data-9334": [
11            {"x": "a", "y": 1},
12            {"x": "b", "y": 4},
13            {"x": "c", "y": 2}
14        ]
15    }
16 },
17 "$schema":
18 "https://vega.github.io/schema/vega-lite/v3.2.0.json"

```

How it works?

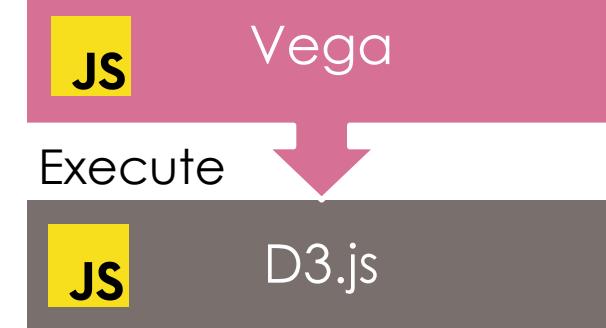
```
1 df = pd.DataFrame({  
2     "x": ["a", "b", "c"],  
3     "y": [1, 4, 2]  
4 })  
5 chart = (  
6     alt.Chart(data=df)  
7     .mark_bar()  
8     .encode(  
9         x="x",  
10        y="y"  
11    )  
12 )  
13 chart
```



13 lines Python code



IPython Kernel

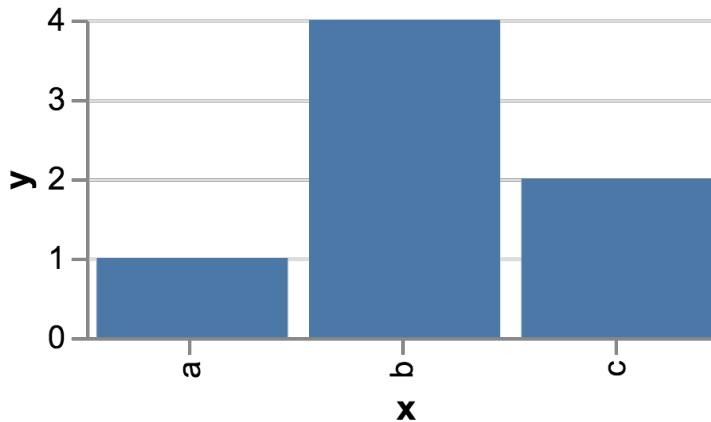


kaggle™



Data is processed on JS side

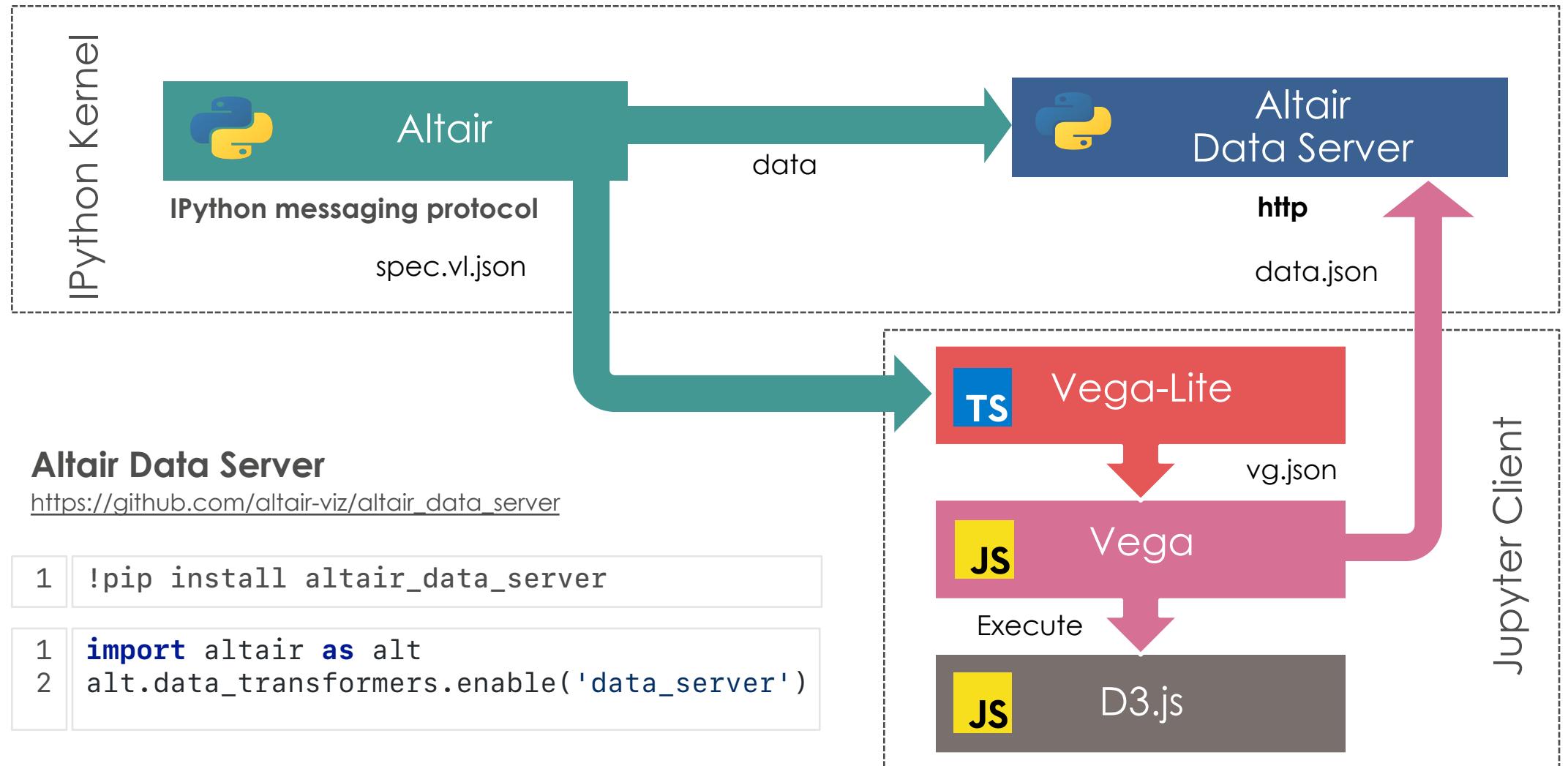
```
1 df = pd.DataFrame({  
2     "x": ["a", "b", "c"],  
3     "y": [1, 4, 2]  
4 })  
5 chart = (  
6     alt.Chart(data=df)  
7     .mark_bar()  
8     .encode(  
9         x="x",  
10        y="y"  
11    )  
12 )  
13 chart
```



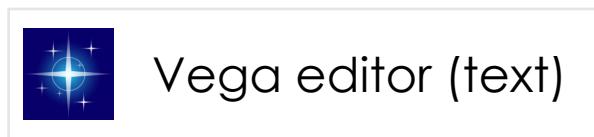
Vega – lite grammar for interactive graphics (JSON)

```
1 {  
2     "config": {"view": {"width": 400, "height": 200}},  
3     "data": {"name": "data-9334"},  
4     "mark": "bar",  
5     "encoding": {  
6         "x": {"field": "x", "type": "nominal"},  
7         "y": {"field": "y", "type": "quantitative"}  
8     },  
9     "datasets": {  
10        "data-9334": [  
11            {"x": "a", "y": 1},  
12            {"x": "b", "y": 4},  
13            {"x": "c", "y": 2}  
14        ]  
15    }  
16 },  
17 "$schema":  
18 "https://vega.github.io/schema/vega-lite/v3.2.0.json"
```

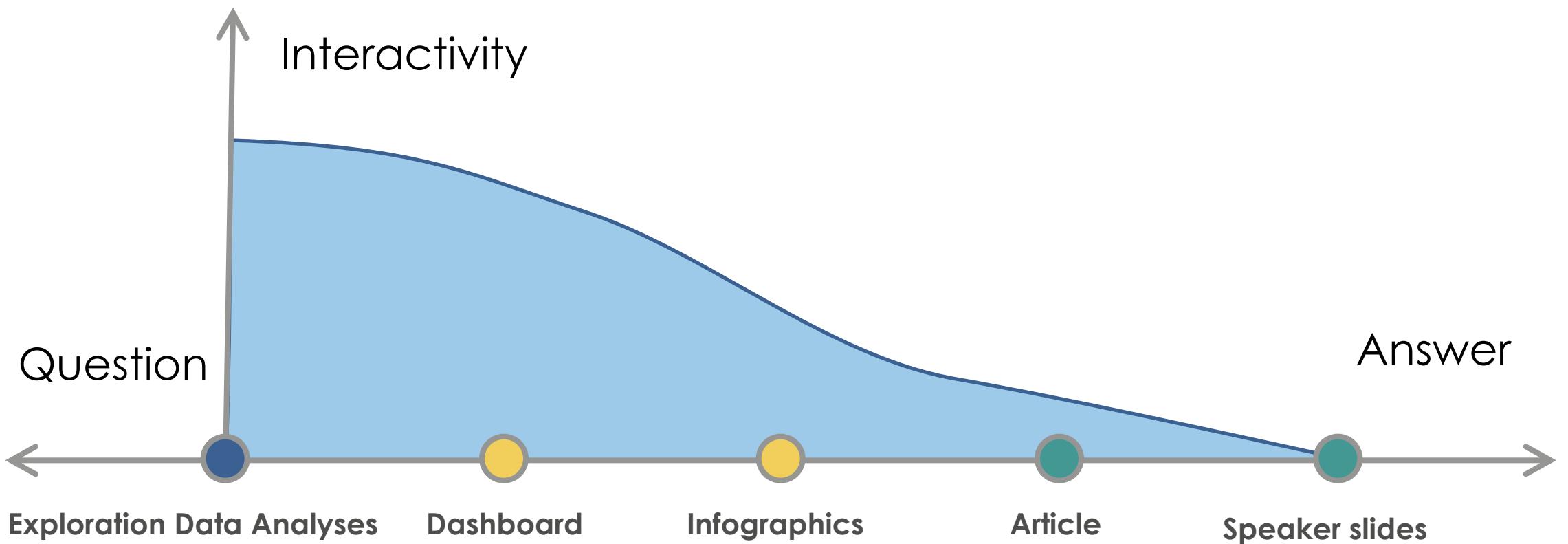
Transfer data over http



Vega-lite (Vega) ecosystem



Recap



Supplementary materials (notebook, slides, links):
<https://iliatimofeev.github.io/DataFest2019Altair/>

Questions?

