Data Science Concepts and Analysis

Week 3: Explore, visually

- Why visualize?
- Elements of statistical graphics
- Principles of effective visualization

This week: data visualization

Objective: introduce the uses, types, anatomy, and construction of statistical graphics.

- **Why visualize?**
    - No one likes reading a table
    - Exploratory graphics: discovery
    - Presentation graphics: communication
- **Statistical graphics**
    - Elements: axes, geometric objects, aesthetic attributes, and text
    - Construction: mapping data to graphical elements
    - Common statistical graphics
- **Principles of effective visualization**
    - Iliinsky: novel; informative; efficient; pleasing.
    - Workflow for developing effective plots
    - Tips and advice

Why visualize?

- No one likes reading a table
- Exploratory graphics: discovery
- Presentation graphics: communication

Notice your reaction

Table 1. Mean Achievement Estimates by Gender, Subject, Grade, and Year

	ELA							Math						
	2009	2010	2011	2012	2013	2014	2015	2009	2010	2011	2012	2013	2014	2015
							Male							
3	-0.02	-0.03	-0.03	-0.03	-0.05	-0.03	-0.05	0.08	0.08	0.07	0.07	0.07	0.08	0.09
4	-0.03	-0.03	-0.04	-0.05	-0.06	-0.05	-0.07	0.06	0.06	0.05	0.05	0.05	0.07	0.06
5	-0.02	-0.05	-0.05	-0.05	-0.06	-0.06	-0.09	0.05	0.05	0.04	0.04	0.03	0.04	0.03
6	-0.03	-0.04	-0.04	-0.05	-0.06	-0.07	-0.10	0.06	0.05	0.04	0.04	0.03	0.03	0.02
7	-0.05	-0.05	-0.06	-0.06	-0.08	-0.08	-0.11	0.05	0.05	0.04	0.02	0.01	0.01	0.00
8	-0.06	-0.06	-0.06	-0.06	-0.08	-0.10	-0.11	0.07	0.05	0.05	0.02	0.01	0.01	0.00
							Female							
3	0.18	0.16	0.16	0.17	0.16	0.17	0.18	0.05	0.04	0.03	0.04	0.04	0.05	0.07
4	0.17	0.15	0.15	0.16	0.15	0.16	0.17	0.03	0.03	0.03	0.05	0.03	0.04	0.05
5	0.17	0.16	0.15	0.16	0.15	0.15	0.17	0.03	0.03	0.02	0.04	0.03	0.04	0.06
6	0.19	0.19	0.18	0.19	0.17	0.17	0.20	0.07	0.06	0.06	0.05	0.06	0.07	0.08
7	0.20	0.20	0.20	0.20	0.18	0.18	0.22	0.08	0.07	0.06	0.06	0.06	0.07	0.06
8	0.22	0.21	0.21	0.20	0.19	0.19	0.23	0.08	0.07	0.06	0.05	0.06	0.07	0.09
							Male-Female							
3	-0.19	-0.19	-0.20	-0.20	-0.21	-0.20	-0.22	0.03	0.04	0.03	0.03	0.03	0.03	0.02
4	-0.20	-0.18	-0.20	-0.21	-0.21	-0.21	-0.24	0.03	0.03	0.02	0.01	0.02	0.03	0.01
5	-0.19	-0.21	-0.20	-0.20	-0.21	-0.21	-0.26	0.02	0.02	0.01	0.00	0.00	-0.01	-0.03
6	-0.22	-0.22	-0.22	-0.23	-0.24	-0.25	-0.29	-0.01	-0.01	-0.02	-0.01	-0.03	-0.05	-0.06
7	-0.25	-0.25	-0.26	-0.26	-0.26	-0.26	-0.33	-0.03	-0.02	-0.03	-0.04	-0.04	-0.06	-0.06
8	-0.27	-0.27	-0.27	-0.27	-0.27	-0.29	-0.35	-0.01	-0.02	-0.02	-0.03	-0.05	-0.06	-0.10

Notes: Table is based on the mean achievement estimates, standardized to the National NAEP distribution within subject, grade and year. To account for the fact that the data are unbalanced (not all districts have estimates in each grade, year, and subject), we obtain the estimated average test score means for each subject, grade, and year by gender from a model regressing the average test scores in a gender-subject-grade-year-district cell on a set of district and gender-subject-grade-year fixed effects. The averages reported in each cell in Table 1 are the estimated coefficients from the gender-subject-grade-year dummy variables in this model. Note that these averages are not weighted by sample size, and thus reflect the mean test score for the average district in each subject, grade, year, and gender.

(There's a reaason that tables are usually put in appendices.)

Uses of graphics

Graphics can be used for one of two main purposes.

1. Discovery
2. Communication of findings

We will focus on using graphics for discovery.

Types of graphics

There is a broad distinction between:

- **exploratory graphics**, which are intended to be seen only by analysts; and
- **presentation graphics**, which are intended to be seen by an audience.

Exploratory graphics are made quickly in large volumes, and usually not formatted too carefully. Think of them like the pages of a sketchbook.

Presentation graphics are made slowly with great attention to detail. Think of them as exhibition artworks.

The two are not mutually exclusive: an especially helpful exploratory graphic is often worth developing as a presentation graphic to help an audience understand 'what the data look like'.

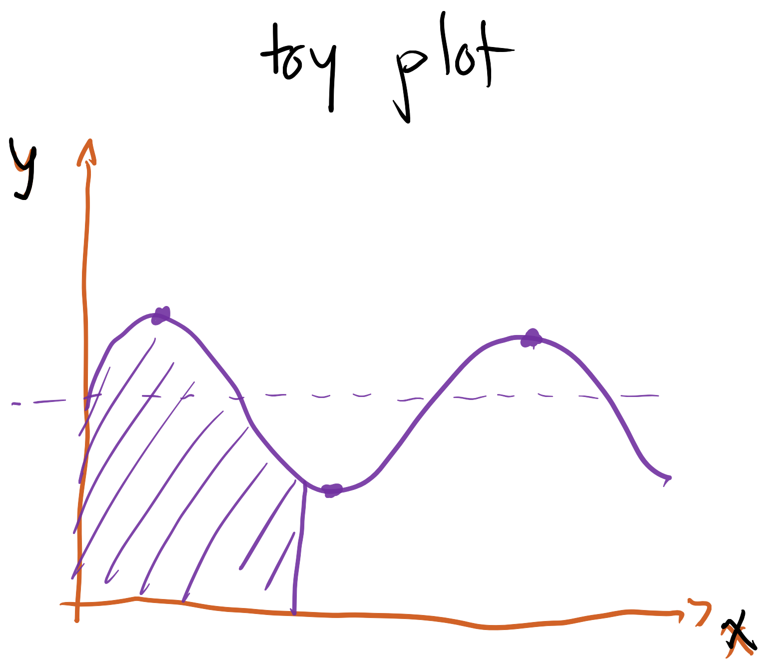
Statistical graphics

- Elements: axes, geometric objects, aesthetic attributes, and text
- Construction: mapping data to graphical elements
- Common statistical graphics

Elements of statistical graphics

Statistical graphics are actually quite simple. They consist of the following four elements:

1. **Axes**
    - References for all other graphical elements.
2. **Geometric objects**
    - Points, lines, curves, filled regions, etc.
3. **Aesthetic attributes**
    - Color, shape, size, opacity/transparency.
4. **Text**
    - Labels, legends, and titles.



geometric objects in purple!
- located relative to axes

text

Axes

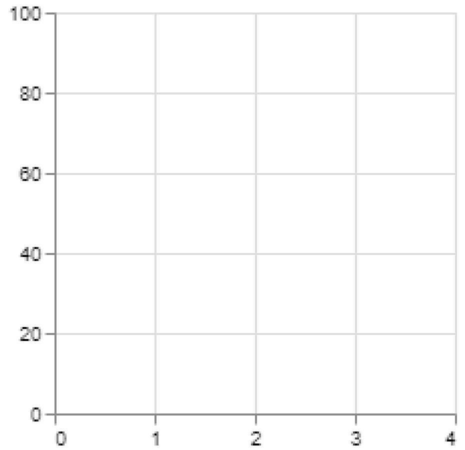
Axes

We are all familiar with axes. The word *axis* literally means axle: an axis is an object that other things turn around.
In graphics, axes establish references for locating any geometric object -- line, point, polygon -- on the graphic.

In [3]:

```
axes
```

Out[3]:



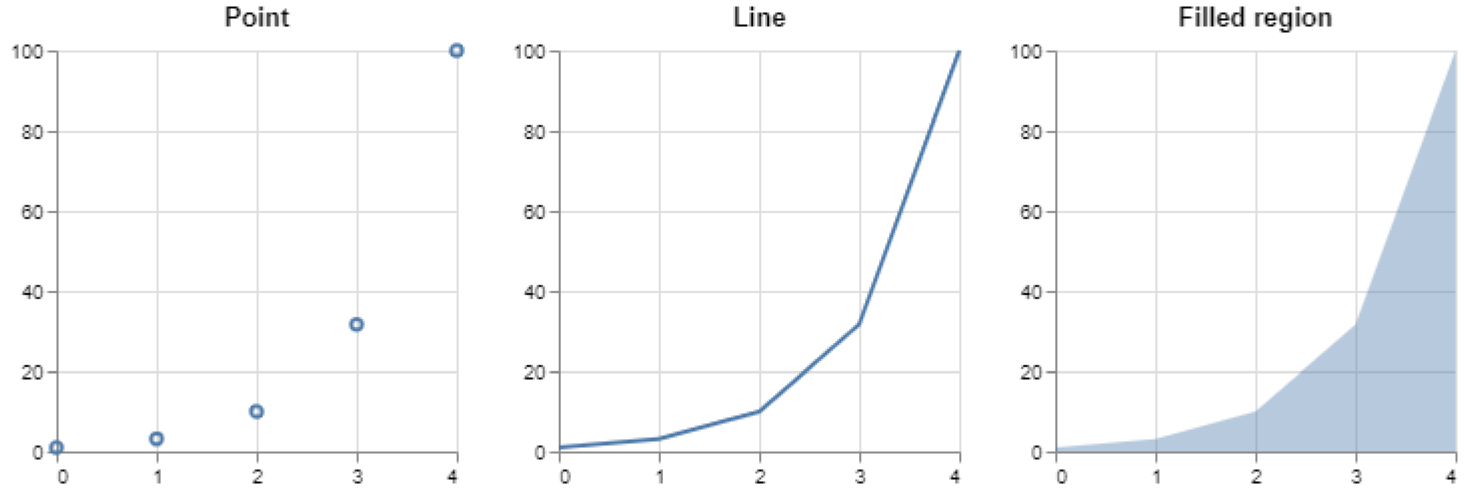
Geometric objects

Geometric objects are the things depicted on a plot, whatever those may be.
Typically, they are points, lines, paths, shapes, and areas.

In [4]:

```
pointgeom | linegeom | polygeom
```

Out[4]:



Aesthetic attributes

For us, aesthetic attributes (or 'aesthetics' for short) will mean qualities of geometric objects, like color.

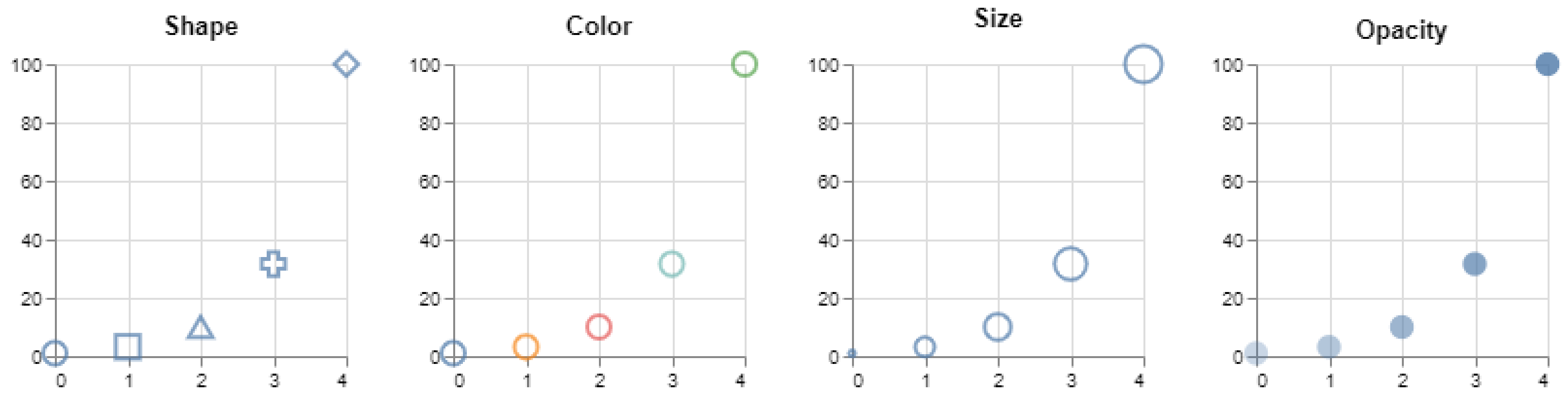
Primary aesthetics in graphics are:

- Shape (for points)
- Color
- Size
- Opacity

In [5]:

```
shapes | colors | sizes | opacities
```

Out[5]:



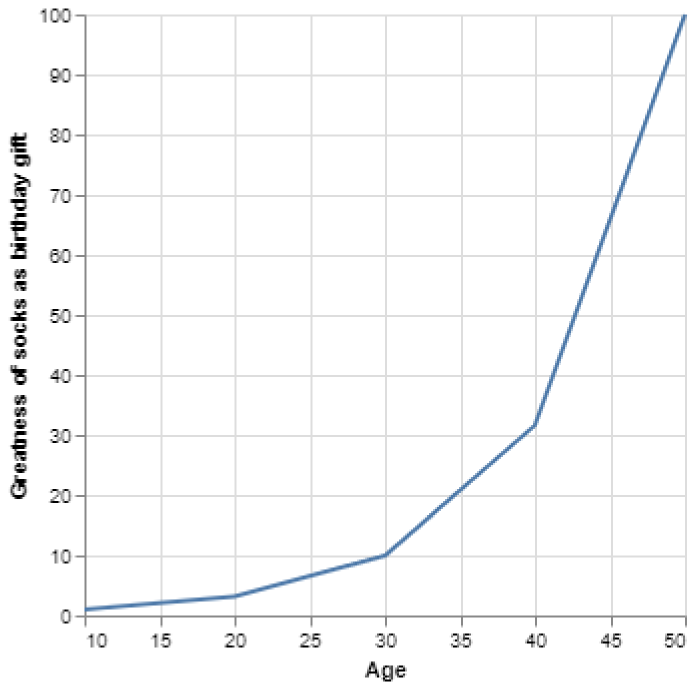
Text

Text is used to label axes, objects, legends, and specify titles.
Text may seem innocuous, but it is what creates story -- text gives a plot its plot!

In [7]:

```
labels
```

Out[7]:



Statistical graphics are mappings

Statistical graphics are **mappings** of dataframe columns to geometric objects and aesthetic attributes. For a simple example, consider the following time series of Cuba's population by year:

In [9]:

```
pop[pop['Country Code'] == 'CUB'].head(2)
```

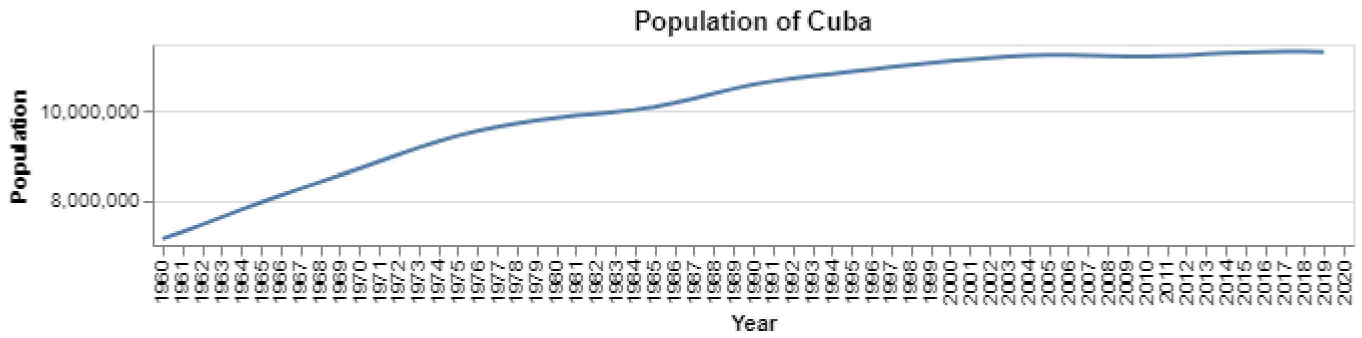
Out[9]:

	Country Name	Country Code	Year	Population
48	Cuba	CUB	1960	7141250.0
312	Cuba	CUB	1961	7291200.0

In [11]:

```
pop_plot
```

Out[11]:



In the plot:

- population $\longrightarrow$ y-coordinate;
- year $\longrightarrow$ x-coordinate;
- the line connects the rows of the dataframe.

Mapping columns to aesthetics

Now consider aggregated populations by global region and year:

In [13]:

```
popregion.head(2)
```

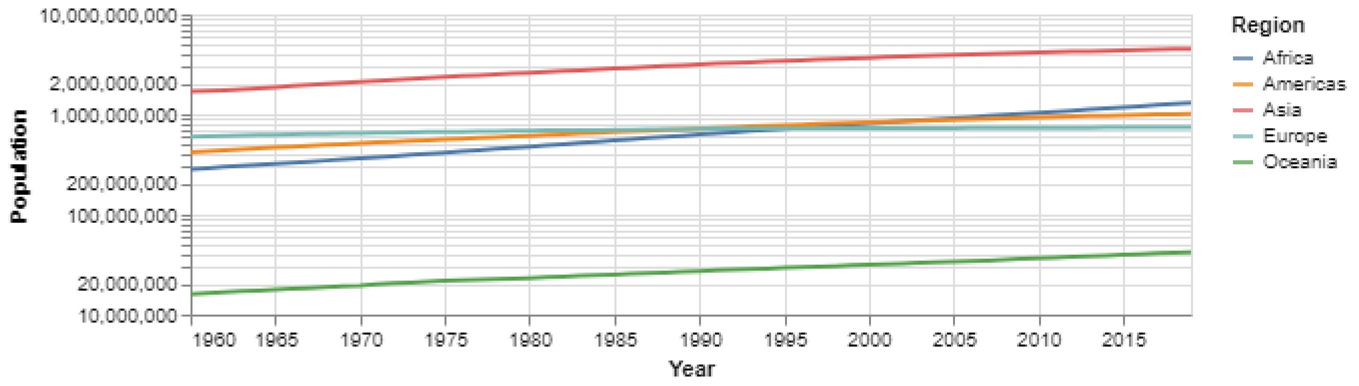
Out[13]:

	Region	Year	Population
0	Africa	1960	282962801.0
1	Africa	1961	289804906.0

In [14]:

```
popregion_plot
```

Out[14]:



In this plot:

- population $\longrightarrow$ y
- year $\longrightarrow$ x
- region $\longrightarrow$ color

Construction of statistical graphics

This may seem like an overly theoretical framework for describing simple plots.

But the ability to map variables to the elements of a graphic is essential because *it means we can display more than two variables at a time by leveraging aesthetic attributes*.

- Creates the possibility to visualize complex information.

You will explore how to leverage this in Lab 3. Here's a preview.

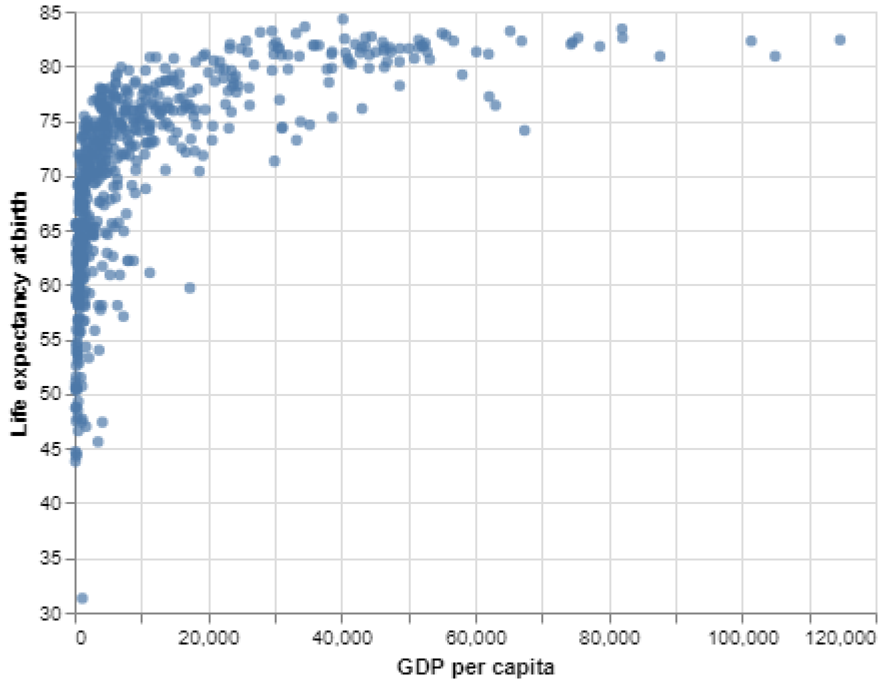
Displaying complex information using aesthetics

In lab 3, you'll begin with this scatterplot:

In [16]:

```
firstplot
```

Out[16]:



Each point represents a country in a particular year. The graphic shows that life expectancy increases with GDP per capita.

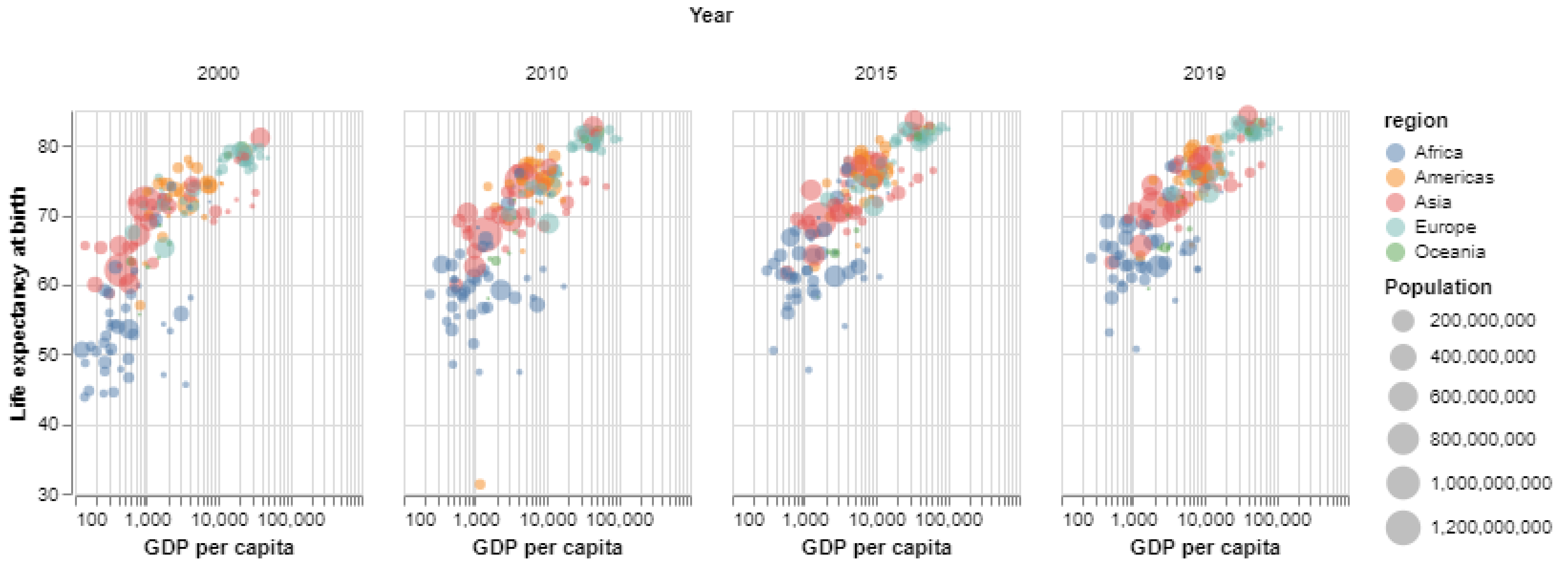
Displaying complex information using aesthetics

And you'll add aesthetic mappings step by step until arriving at this plot:

In [17]:

```
finalplot1
```

Out[17]:



In this plot, each point represents a country and:

- GDP per capita $\longrightarrow$ x
- Life expectancy at birth $\longrightarrow$ y
- Global region $\longrightarrow$ color
- Population $\longrightarrow$ size
- Year $\longrightarrow$ facets (panels)

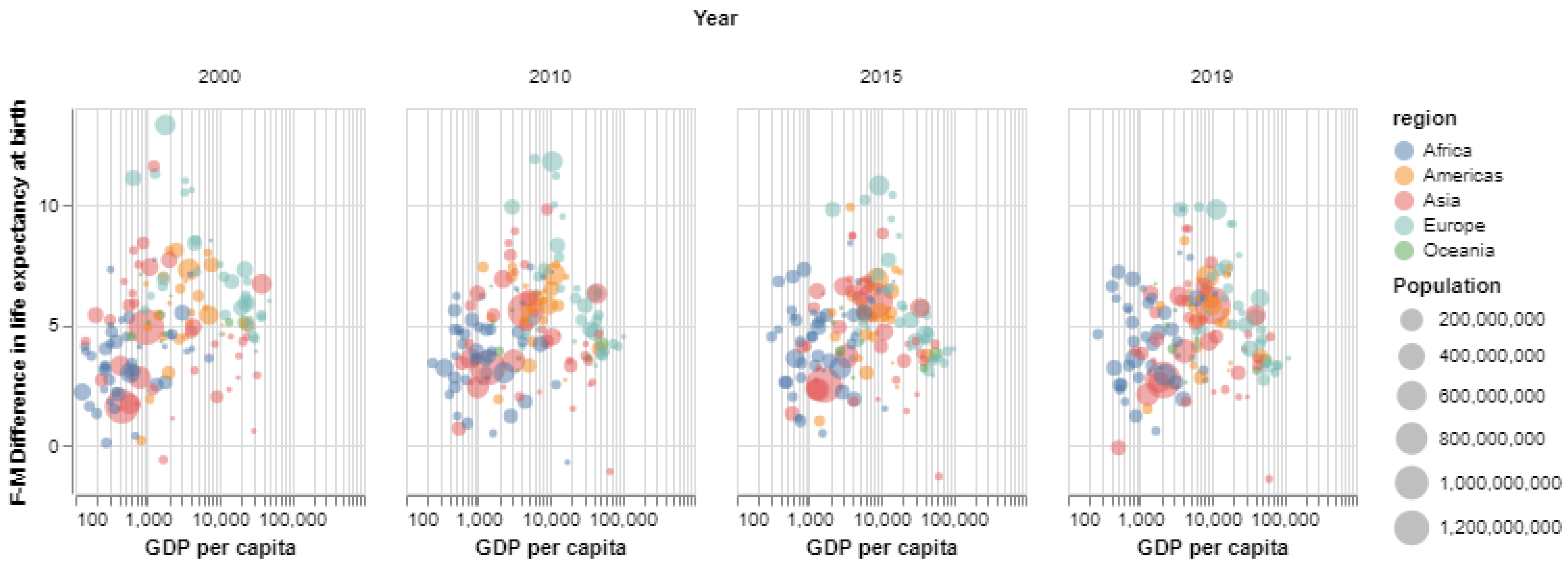
Displaying complex information using aesthetics

You'll then examine life expectancies by sex and show that increases in GDP per capita are associated with differential changes in life expectancy for men and women:

In [18]:

```
finalplot2
```

Out[18]:



The only change here is that the difference in life expectancy between women and men (a simple *transformation* of the data) is mapped to the y axis instead of overall life expectancy.

- Notice that the gap increases with GDP in general, but Oceania exhibits the opposite trend!

Altair

Altair, a python library, creates graphics exactly as described above: mapping columns of a dataframe to graphical elements.

It has a somewhat idiosyncratic syntactical pattern involving a "chart", "marks", and "encodings":

Altair syntax	Example handle	Operation
Chart	`alt.Chart(df)`	Coerces a dataframe `df` to a chart object
Mark	`mark_point()`	Specifies a geometric object
Encoding	`encode(x = ..., y = ..., color = ...)`	Maps columns of `df` to objects and aesthetics

These are chained together to make a graphic.

Region	Year	Population
Africa	1960	260,312,512
⋮	⋮	⋮

In [19]:

```
alt.Chart( # basis for graphic
    popregion
).mark_line( # geometric object
).encode( # mapping
    x = 'Year',
    y = 'Population',
    color = 'Region'
).properties(width = 500, height = 100) # resize
```

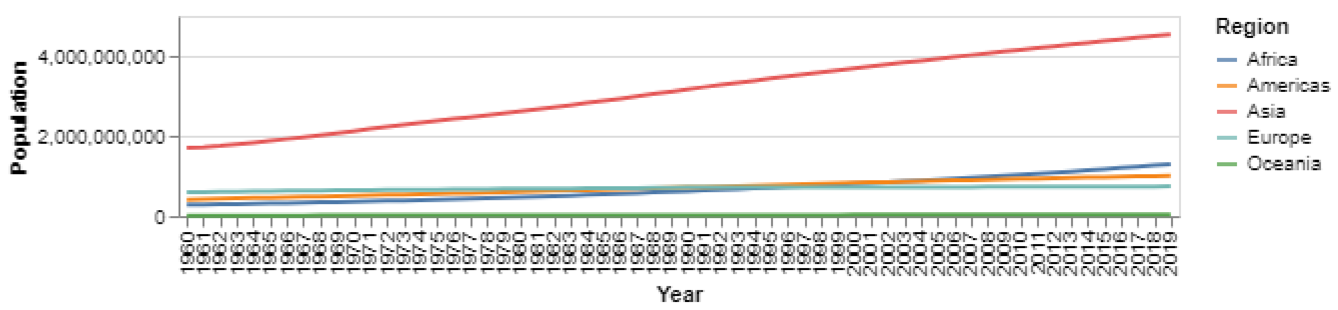
name of a dataframe

is there a col. named year?

year → x
pop → y
region → color

** The names appearing in the encoding need to match the names of columns in the dataframe used to create the chart.

Out[19]:



Altair

Earlier, the y-axis scale was adjusted so that the lines were a bit more spread out in the graphic, and the x-axis had a less congested appearance. This was done by modifying the *encodings*:

- specify data type for year as time
- specify display scale for population

":T" means coerce to time

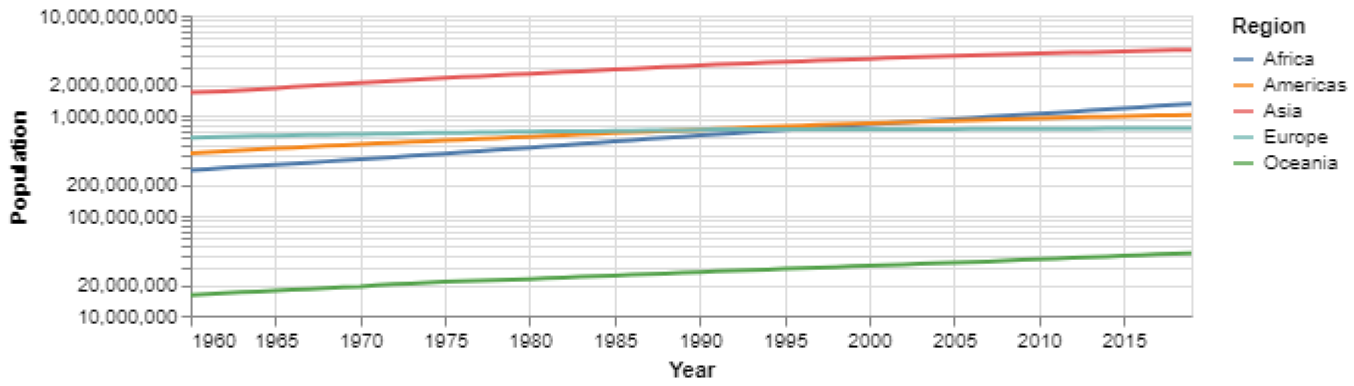
:N nominal / categorical
:Q quantitative

In [20]:

```
alt.Chart(
    popregion
).mark_line(
).encode(
    x = 'Year:T', # change here -- data type specification
    y = alt.Y('Population', scale = alt.Scale(type = 'log')), # change here -- axis scale
    color = 'Region'
).properties(width = 500, height = 150)
```

alt.Y(...) changes all aspects of encodings

Out[20]:



Common statistical graphics and their uses

Broadly, the most common statistical graphics can be divided according to the number of variables that form their primary display. The uses listed below are not exclusive, just some of the most common.

- **Single-variable graphics** are used to visualize distributions.
  - Box plot
  - Histogram

- **Two-variable graphics** are used to visualize relationships.
  - Scatterplot
  - Bar plot
  - Line plot

- **Three-variable graphics** are used to visualize spatial data, matrices, and a collection of other data types.
  - Heatmaps
  - Contour plots

Single-variable graphics: boxplot

Single-variable graphics are used to display the distribution of values of a single variable.
Boxplots show the spread, center, and skewness of values.

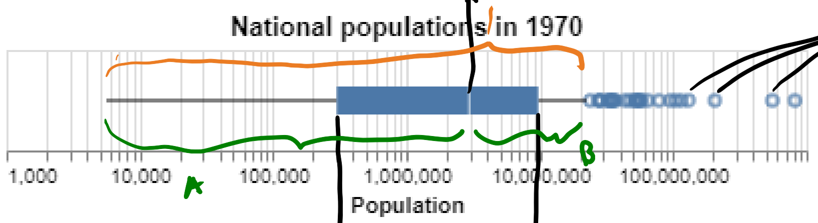
A > B => left skewed

In [22]:

```
boxplot
```

Out[22]:

0.5 quantile, i.e., median



outliers (usually > 2.5 SD)

0.25 quantile

0.75 quantile

means 25% ~~10%~~ of values are smaller

means 75% of values are greater

25% ~~10%~~ of countries in 1970 had populations under 200,000

population
10,375
120,512
10,191
⋮

Single-variable graphics: boxplot

The Altair syntax for boxplot construction is shown below:

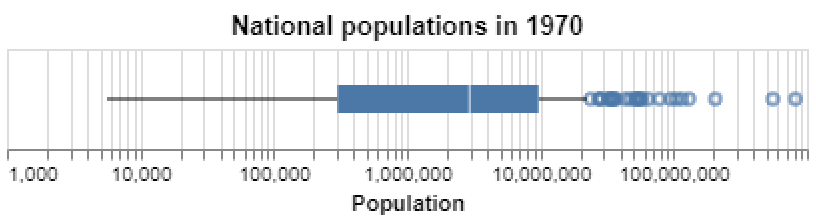
In [23]:

```
boxplot = alt.Chart(
    popcountry.loc["1970"]
).mark_boxplot().encode(
    x = alt.X('Population',
              scale = alt.Scale(type = 'log'))
).properties(
    height = 50,
    title = 'National populations in 1970'
)

boxplot
```

geometric object

variable of interest

Out[23]:



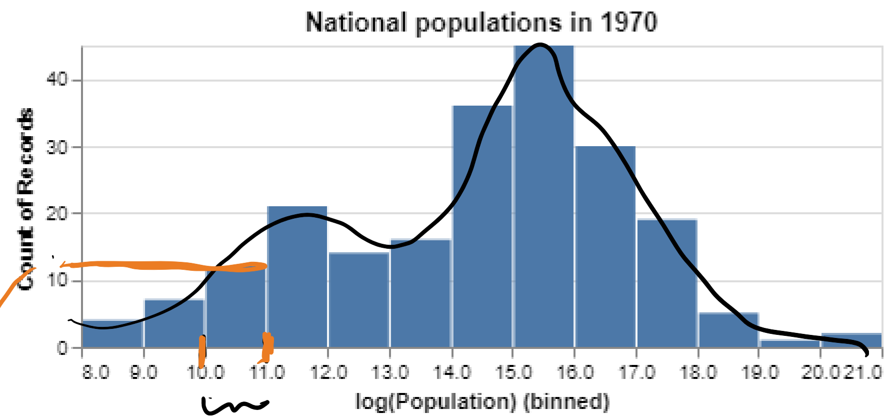
Single-variable graphics: histogram

Histograms show the relative frequencies of values of a single variable.
One can see spread, center, skewness, and outliers, but ***also shape***.

In [24]:

```
histogram
```

Out[24]:



* Bimodal!

a "bin" of values : between 10 & 11

# of rows whose values fall in the bin

here, 12 countries w/ pop $\in (e^{10}, e^{11})$

Single-variable graphics: histogram

The Altair syntax for histogram constuction is as follows:

In [25]:

same as boxplot

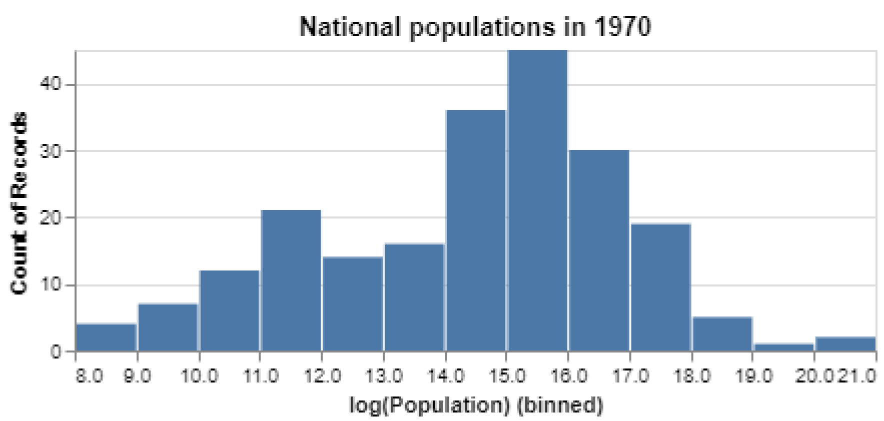
```
histogram = alt.Chart(
    popcountry.loc["1970"]
).mark_bar().encode(
    x = alt.X('log(Population)',
        bin = alt.Bin(maxbins = 20)),
    y = 'count()'
).properties(
    height = 150,
    title = 'National populations in 1970'
)

histogram
```

geometric object: bar

variable of interest

bin x axis into 20 or fewer bins

Out[25]:



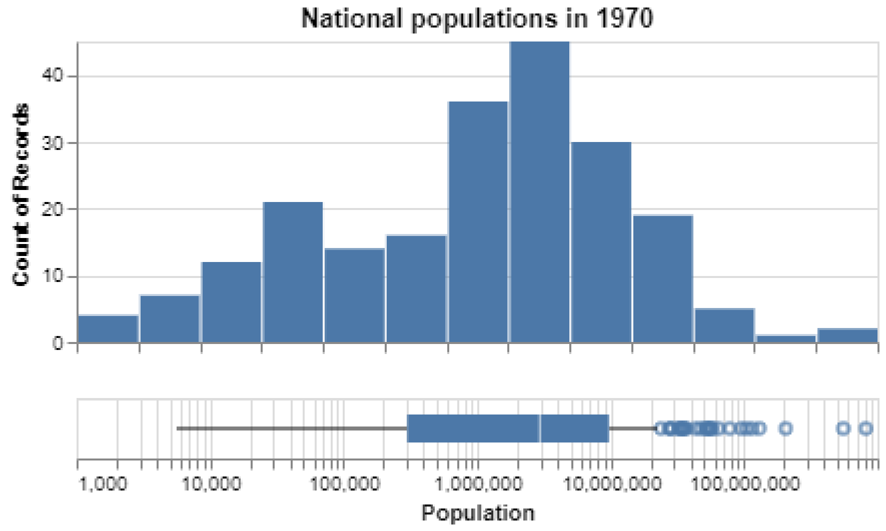
Single-variable graphics

These single-variable graphics show similar information, but a little differently.

In [27]:

```
histogram & boxplot
```

Out[27]:



Notice that you can see two modes in the histogram, but only the primary mode in the boxplot.

Notice also that Altair allows stacking of charts: `chart-top & chart-bottom` aligns charts vertically!

Single-variable graphics

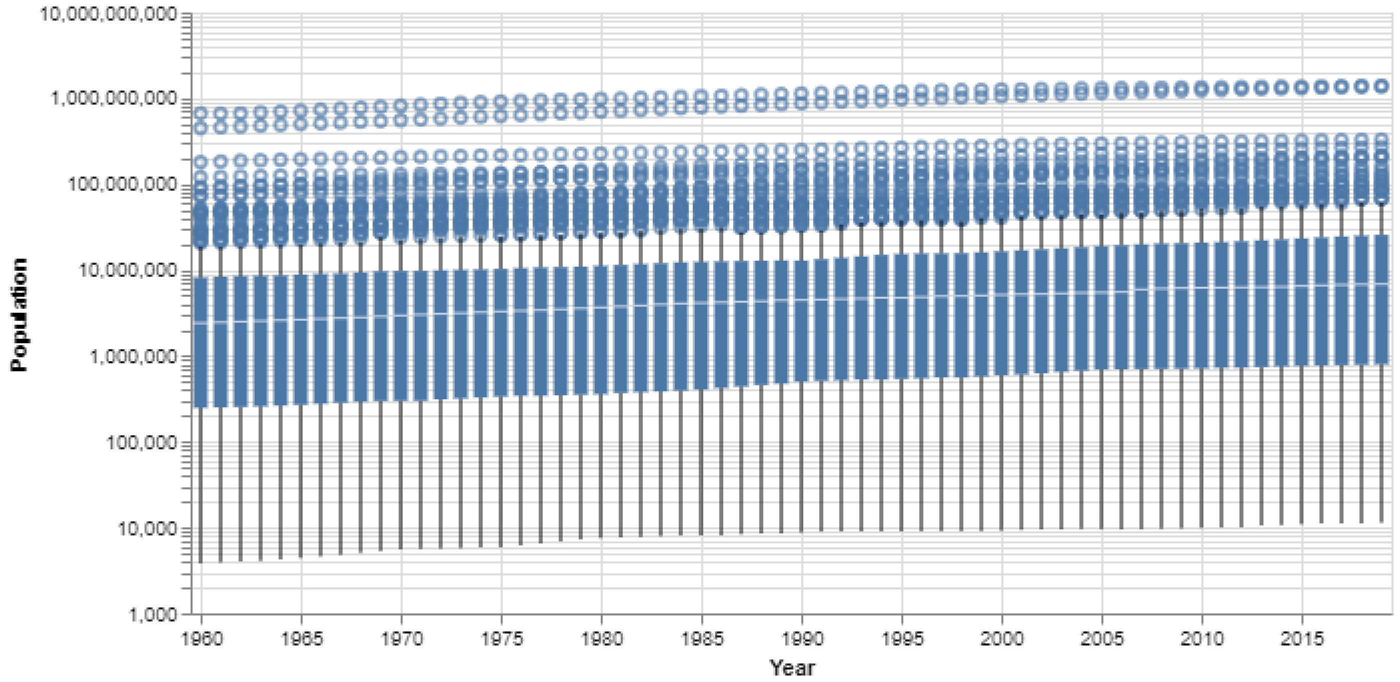
Single-variable graphics are not necessarily limited to univariate data.

For example, we could make one boxplot for each year and show the distributions of national populations for each year:

In [28]:

```
alt.Chart(popcountry.reset_index()).mark_boxplot(outliers = True, size = 7).encode(
    x = 'Year:T',
    y = alt.Y('Population', scale = alt.Scale(type = 'log'))
).properties(width = 600)
```

Out[28]:



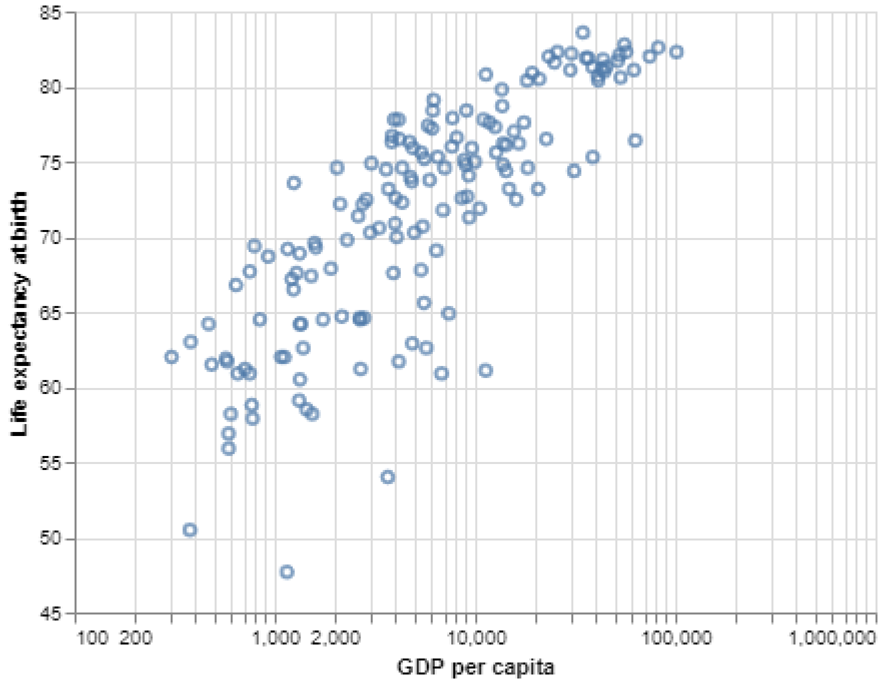
Two-variable graphics: scatterplot

Scatterplots display relationships between two variables.

In [30]:

```
scatter
```

Out[30]:



The pattern of scatter shows that life expectancy generally increases with GDP per capita.

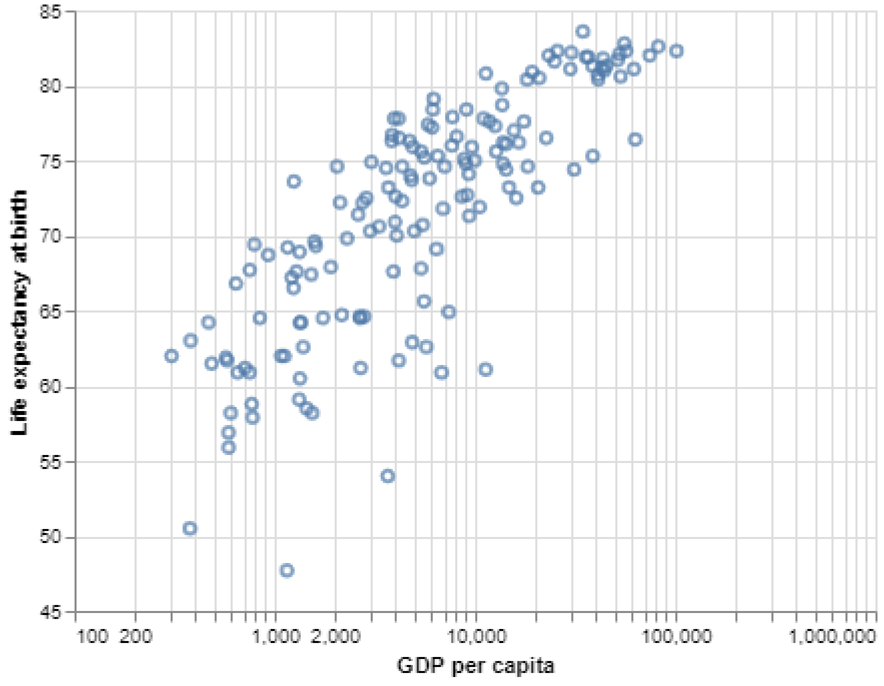
Two-variable graphics: scatterplot

The Altair syntax for this plot is as follows:

In [31]:

```
alt.Chart(
    lifegdp.loc[2015]
).mark_point().encode(
    x = alt.X('GDP per capita',
              scale = alt.Scale(type = 'log')),
    y = alt.Y('All',
              scale = alt.Scale(zero = False),
              title = 'Life expectancy at birth')
)
```

Out[31]:



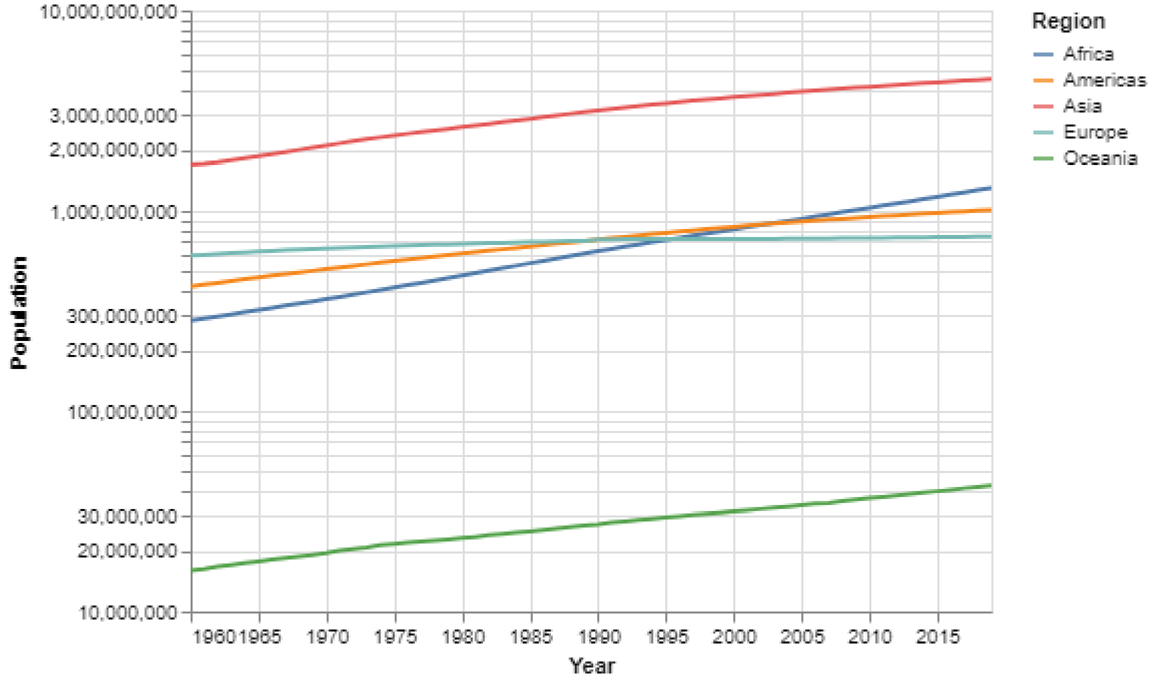
Two-variable graphics: line plot

Line plots display trajectories by connecting rows in a dataframe.
These can represent trends, time courses, or paths traveled.

In [32]:

```
line
```

Out[32]:



Two-variable graphics: line plot

The Altair syntax for this plot is as follows:

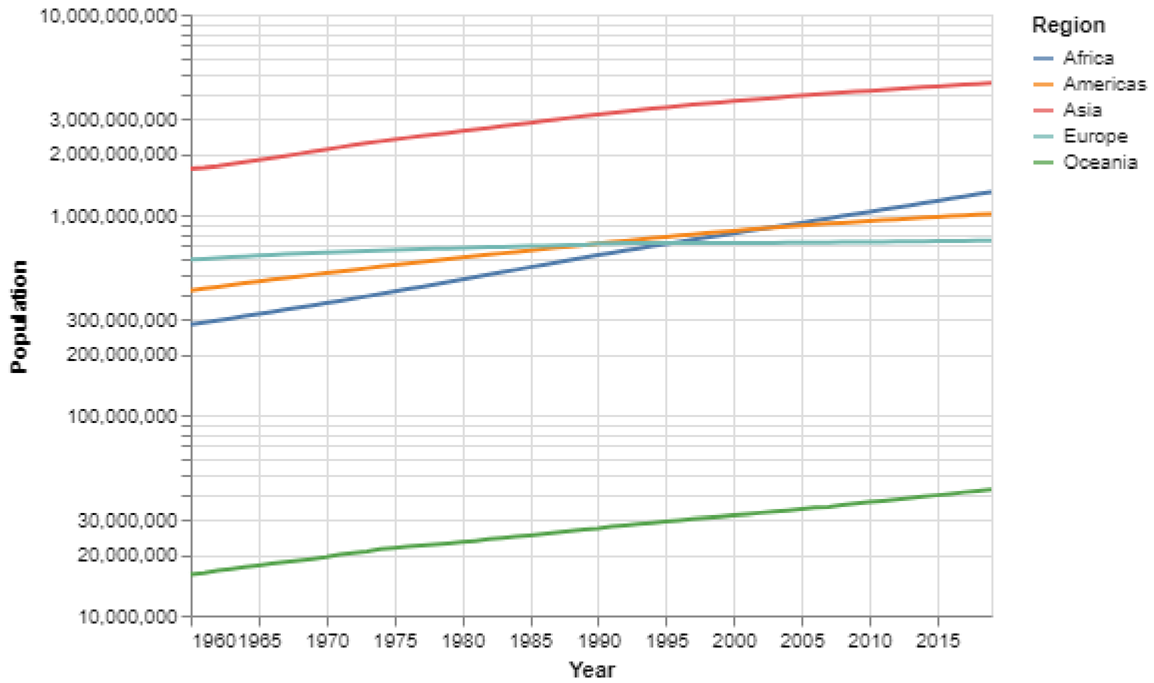
:T means time (data type)

popregion = { Year Population Region
1960 17M Oceania
⋮ ⋮ ⋮

In [33]:

```
alt.Chart(popregion).mark_line().encode(
    x = 'Year:T',
    y = alt.Y('Population', scale = alt.Scale(type = 'log')),
    color = 'Region'
)
```

Out[33]:



Two-variable graphics: bar plot

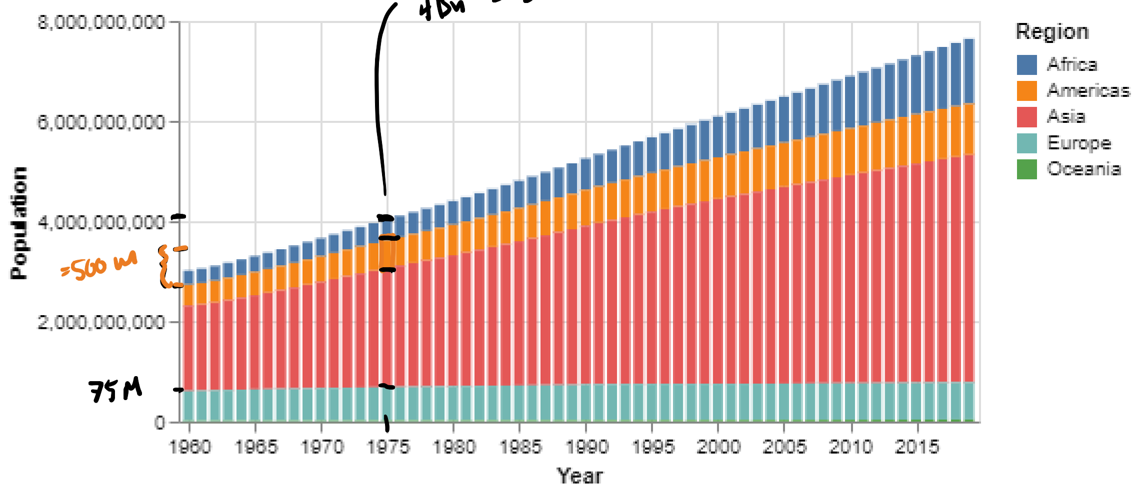
Bar plots usually depict the relationship between the magnitude of one variable and another. For instance:

In [34]:

```
alt.Chart(popregion).mark_bar().encode(
    x = 'Year:T',
    y = alt.Y('Population:Q'),
    color = 'Region'
).properties(height = 200)
```

total height: 4Bn = sum (pop in each region)

Out[34]:



} 500M = pop Americas

} 750 M = pop in Europe

The graphic indicates in particular that:

- global population growth (total height of bar) is increasing linearly;
- the share of the global population in Asia is increasing over time.

Principles of effective visualization

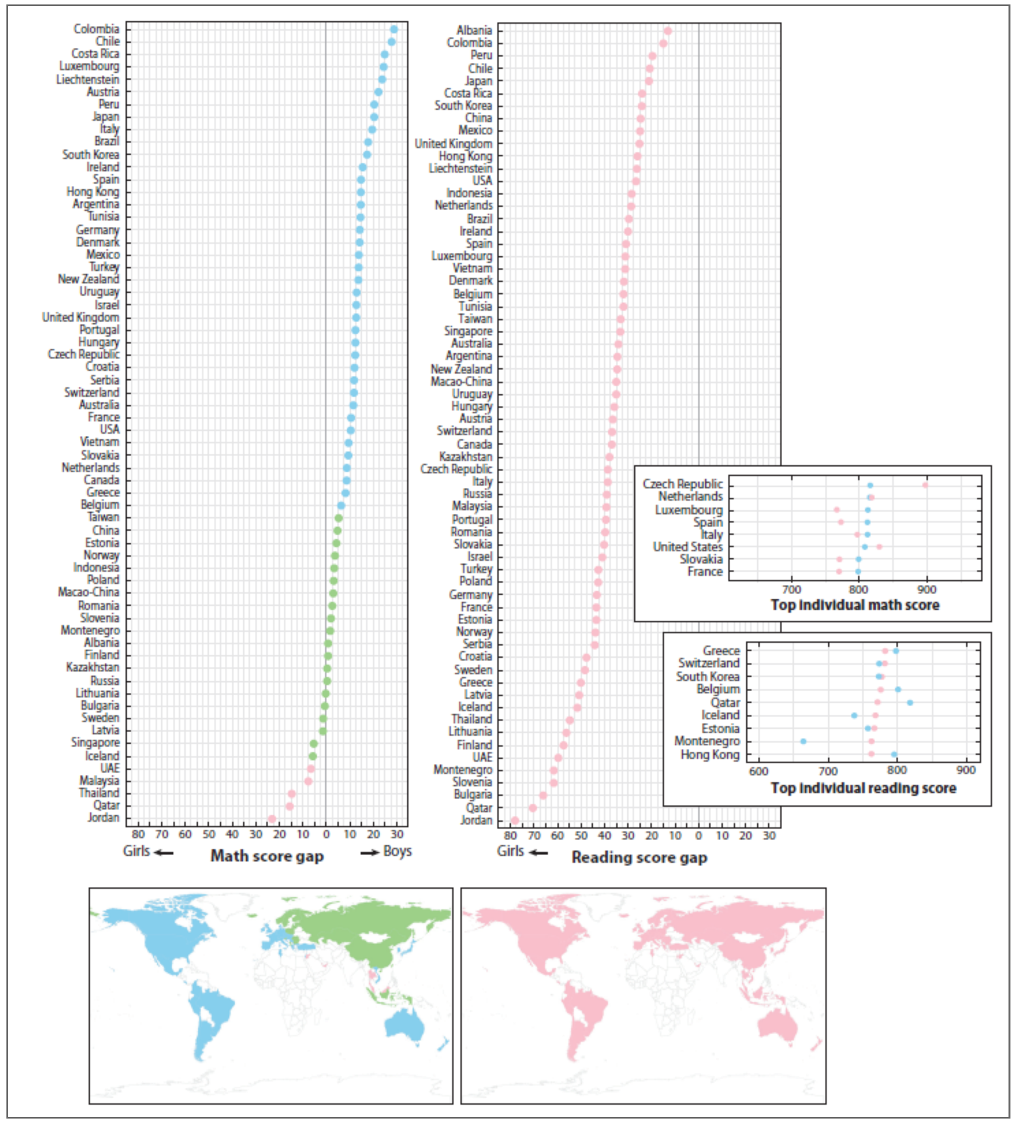
- Desiderata: novel; informative; efficient; pleasing.
- Workflow for developing effective plots
- Tips and advice

What makes visualizations effective?

Iliinsky introduces some *desiderata* for data visualizations.
The article focuses on going beyond conventional 'formats' (scatterplot, bar chart, etc.), but I think we can equally well apply the criteria to standard statistical graphics.

- **Novel.** Novel visuals don't need to elicit superlative reactions, but they should (if only subtly) surprise and spark interest to some extent.
    - Were *you* excited or interested when you made the plot?
- **Informative**. Informative visuals make information apparent. In a way they are *unambiguous*.
    - Did you learn something when you made the plot?
    - Did you have to think very hard to understand what the plot has to say?
- **Efficient**. Efficient visuals have an accessible message. They use space economically but without becoming overly complicated.
    - Include exactly what you want to show -- nothing more, nothing less.
    - Did you add anything that doesn't really need to be there?
- **Pleasant**. Visuals should be nice to look at!
    - Do you smile a little when you open up that plot?

Example


Colombia
Chile
Costa Rica
Luxembourg
Liechtenstein
Austria
Peru
Japan
Italy
Brazil
South Korea
Ireland
Spain
Hong Kong
Argentina
Tunisia
Germany
Denmark
Mexico
Turkey
New Zealand
Uruguay
Israel
United Kingdom
Portugal
Hungary
Czech Republic
Croatia
Serbia
Switzerland
Australia
France
USA
Vietnam
Slovakia
Netherlands
Canada
Greece
Belgium
Taiwan
China
Estonia
Norway
Indonesia
Poland
Macao-China
Romania
Slovenia
Montenegro
Albania
Finland
Kazakhstan
Russia
Lithuania
Bulgaria
Sweden
Latvia
Singapore
Iceland
UAE
Malaysia
Thailand
Qatar
Jordan
80 70 60 50 40 30 20 10 0 10 20 30
Girls
Math score gap
Boys
Albania
Colombia
Peru
Chile
Japan
Costa Rica
South Korea
China
Mexico
United Kingdom
Hong Kong
Liechtenstein
USA
Indonesia
Netherlands
Brazil
Ireland
Spain
Luxembourg
Vietnam
Denmark
Belgium
Tunisia
Taiwan
Singapore
Australia
Argentina
New Zealand
Macao-China
Uruguay
Hungary
Austria
Switzerland
Canada
Kazakhstan
Czech Republic
Italy
Russia
Malaysia
Portugal
Romania
Slovakia
Israel
Turkey
Poland
Germany
France
Estonia
Norway
Serbia
Croatia
Sweden
Greece
Latvia
Iceland
Thailand
Lithuania
Finland
UAE
Montenegro
Slovenia
Bulgaria
Qatar
Jordan
80 70 60 50 40 30 20 10 0 10 20 30
Girls
Reading score gap
Czech Republic
Netherlands
Luxembourg
Spain
Italy
United States
Slovakia
France
700 800 900
Top individual math score
Greece
Switzerland
South Korea
Belgium
Qatar
Iceland
Estonia
Montenegro
Hong Kong
600 700 800 900
Top individual reading score

Workflow tips and advice

Developing graphics is an iterative trial-and-error process:

- make a plot, see how it looks, think about what to add or change, make another and repeat.

This process is modeled in Lab 3. When you're on your own, here are some general guidelines:

- **Hone in on an essential question or relationship.**
    - This might require making a few crude plots to help you decide where to focus.
- **Start simple.**
    - Begin with the most basic plot you can.
- **Add complexity as you go.**
    - Decide what to add based on *your own questions* rather than an envisioned endpoint.
- **Change one thing at a time, and keep copies.**
    - You might well need to backtrack.
- **Don't be afraid of starting over.**
    - If an idea doesn't pan out, no harm done!

Summary

This has been a primer on visualization basics. We touched on the following points:

- The elements of statistical graphics are axes, geometric objects, aesthetics, and text
- Statistical graphics are constructed by mapping dataframe columns to geometric objects and aesthetics
    - Allows for display of complex information
    - Specified in Altair as marks and encodings on a chart
- The basic statistical graphics are histograms, boxplots, scatterplots, line plots, and barplots.
    - Think of these as building blocks you can use to develop more sophisticated visualizations
- Effective data visualizations are novel, informative, efficient/economical, and pleasing.
- Developing visualizations is an iterative process
    - Start simple, add complexity, and experiment!

You will refine your understanding as you use these tools in context, and see many examples throughout the course.