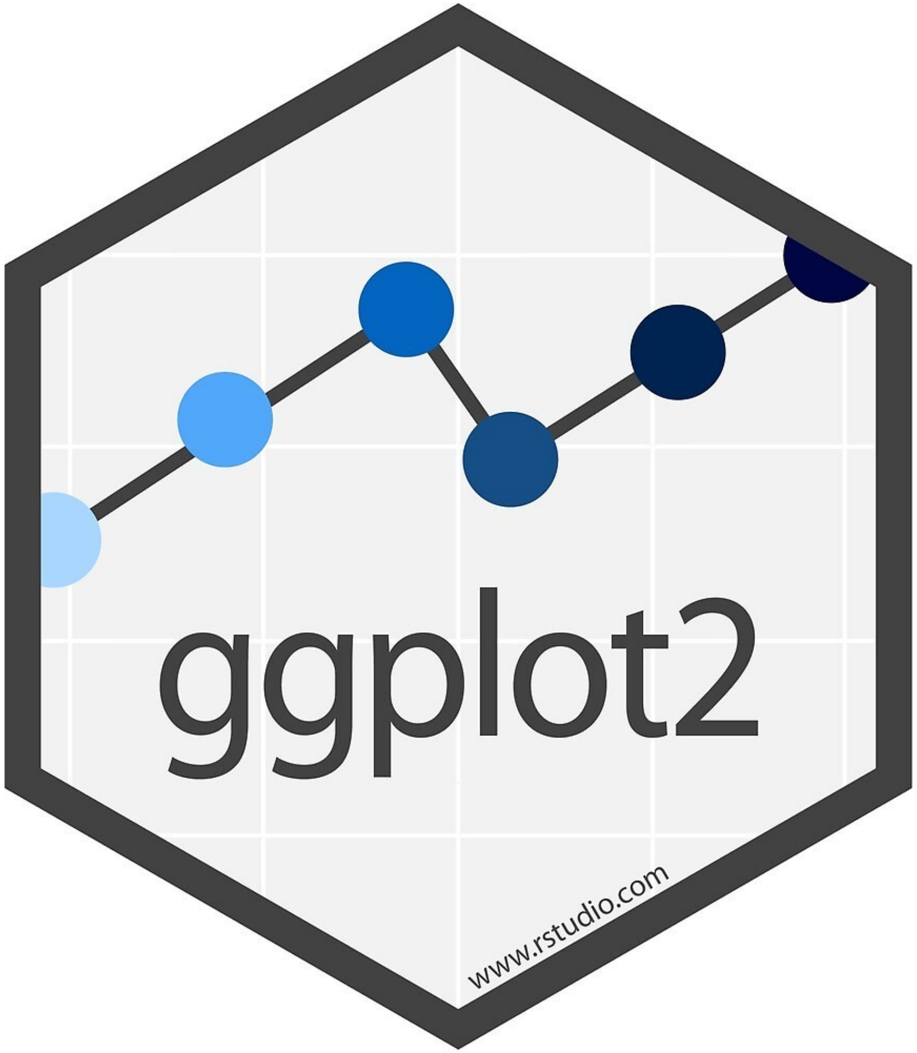
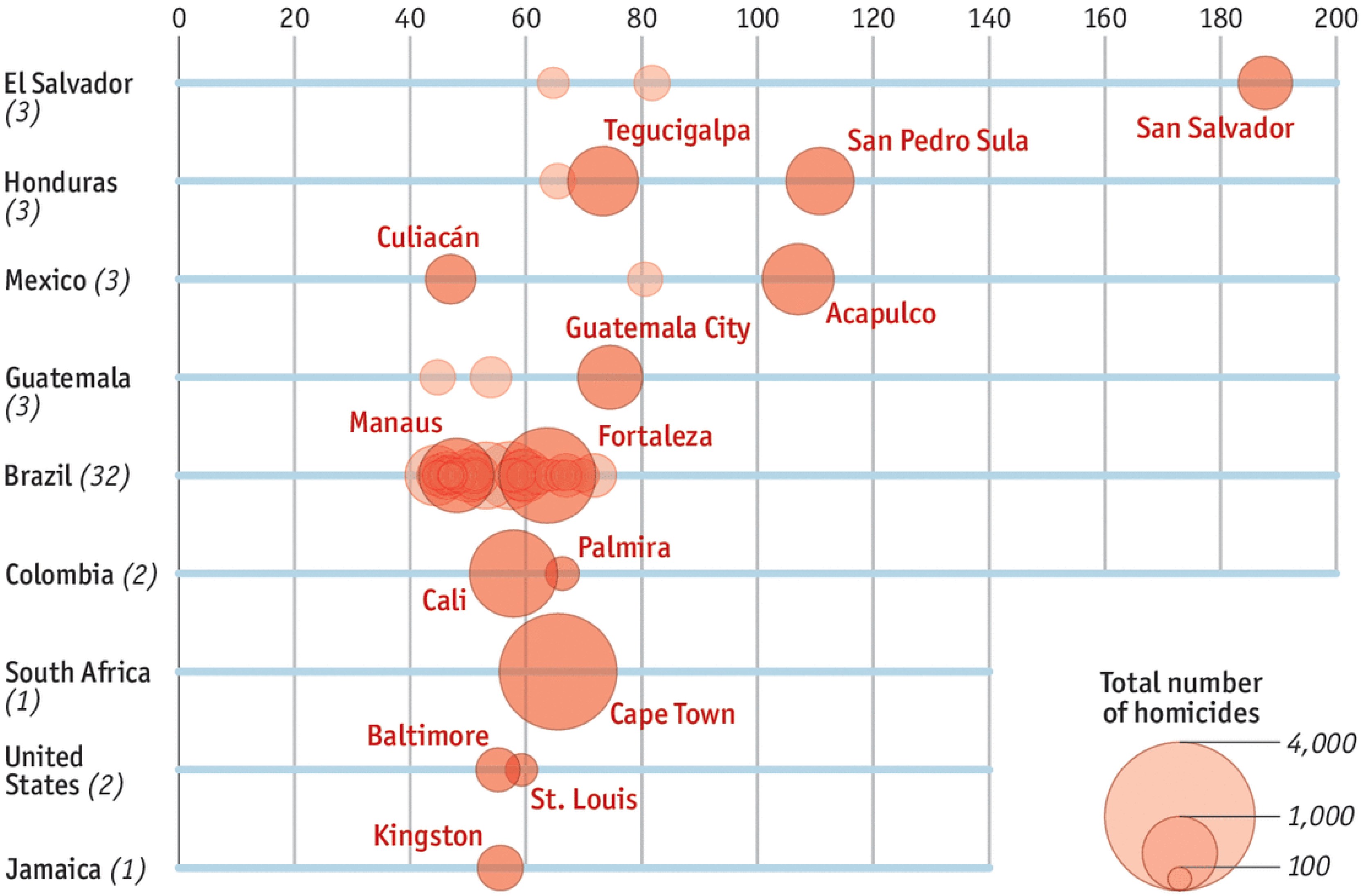


The Grammar of Graphics

The world’s most murderous metropolises (re-ranked)

Homicides per 100,000 population, 50 worst cities*, 2015 or latest available

(Number of cities listed per country)



Sources: Igarapé Institute; press reports; *The Economist*

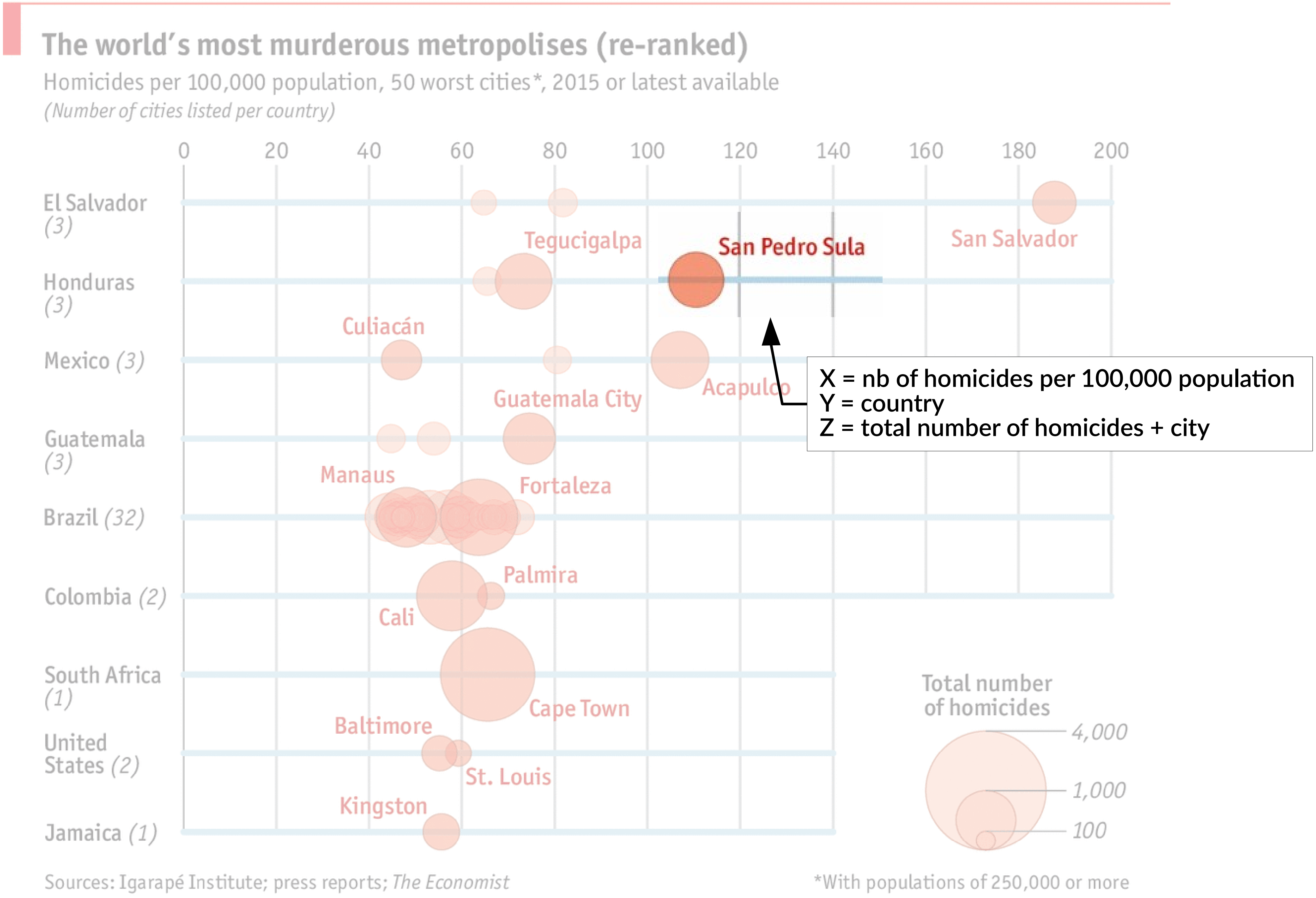
*With populations of 250,000 or more

Economist.com

The world's most murderous metropolises (re-ranked)

Homicides per 100,000 population, 50 worst cities*, 2015 or latest available

(Number of cities listed per country)

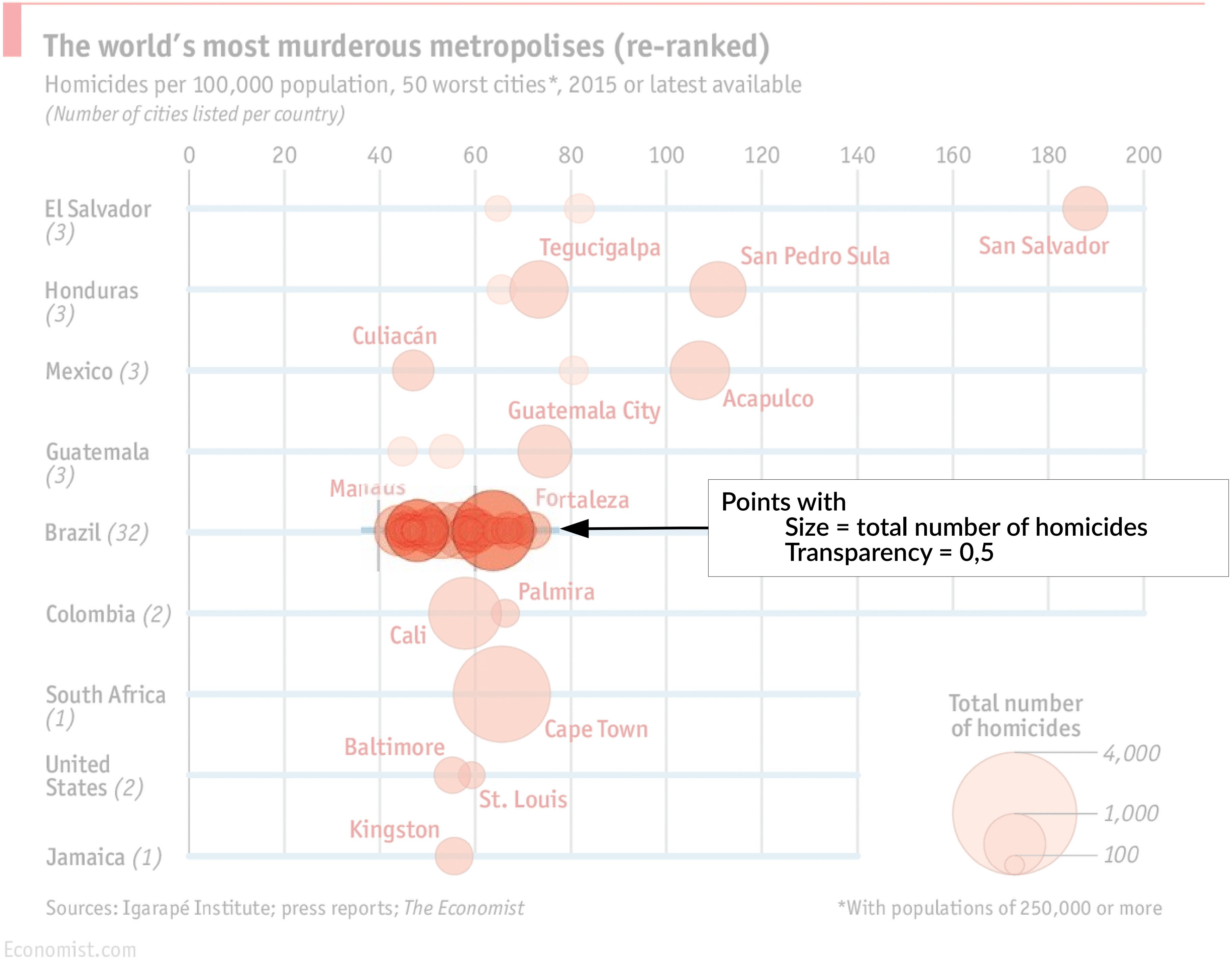


X = nb of homicides per 100,000 population
Y = country
Z = total number of homicides + city

Sources: Igarapé Institute; press reports; *The Economist*

*With populations of 250,000 or more

Economist.com



The world's most murderous metropolises (re-ranked)

Homicides per 100,000 population, 50 worst cities*, 2015 or latest available

(Number of cities listed per country)

Sources: Igarapé Institute; press reports; *The Economist*

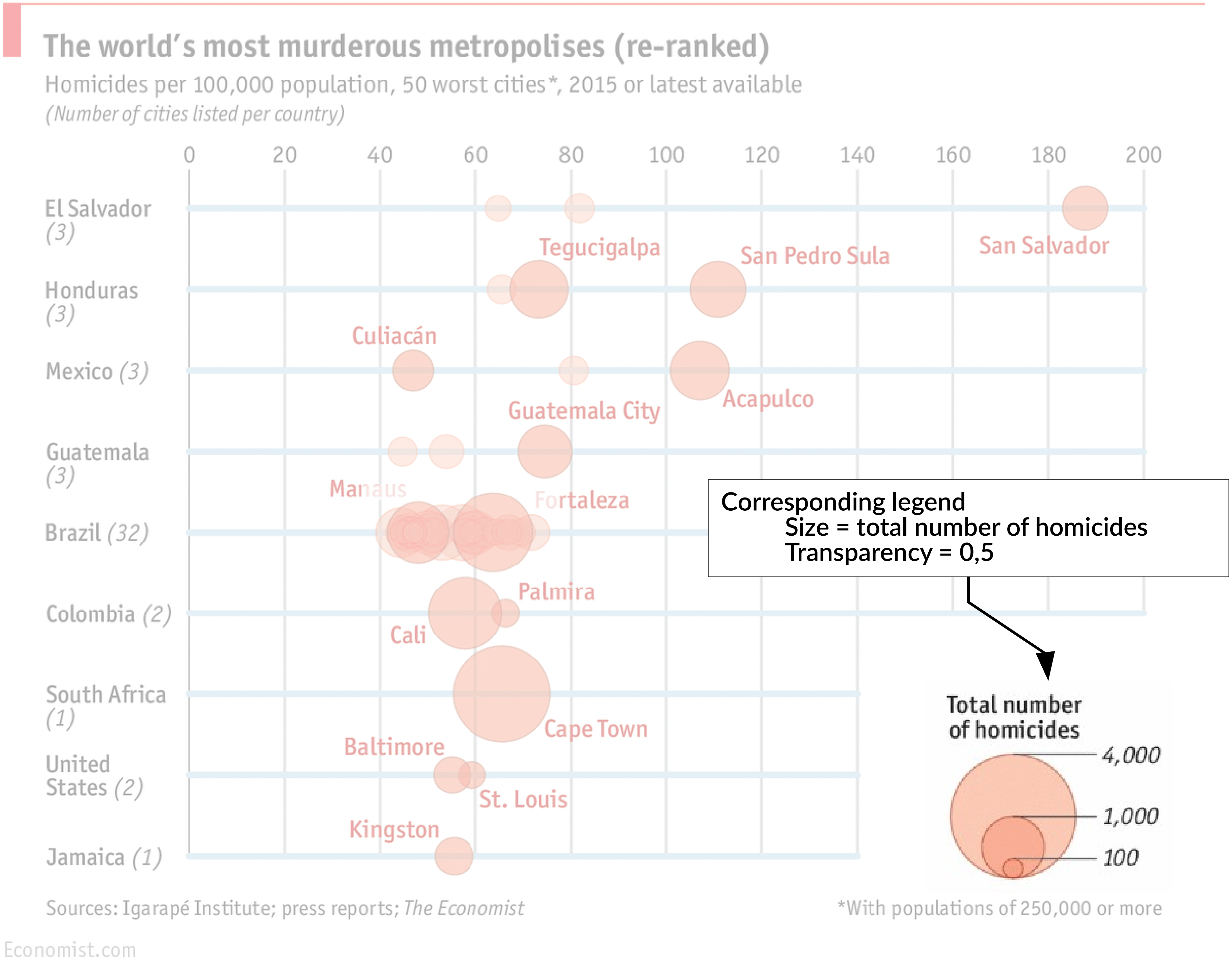
*With populations of 250,000 or more

Economist.com

Points with

Size = total number of homicides

Transparency = 0,5



The world's most murderous metropolises (re-ranked)

Homicides per 100,000 population, 50 worst cities*, 2015 or latest available

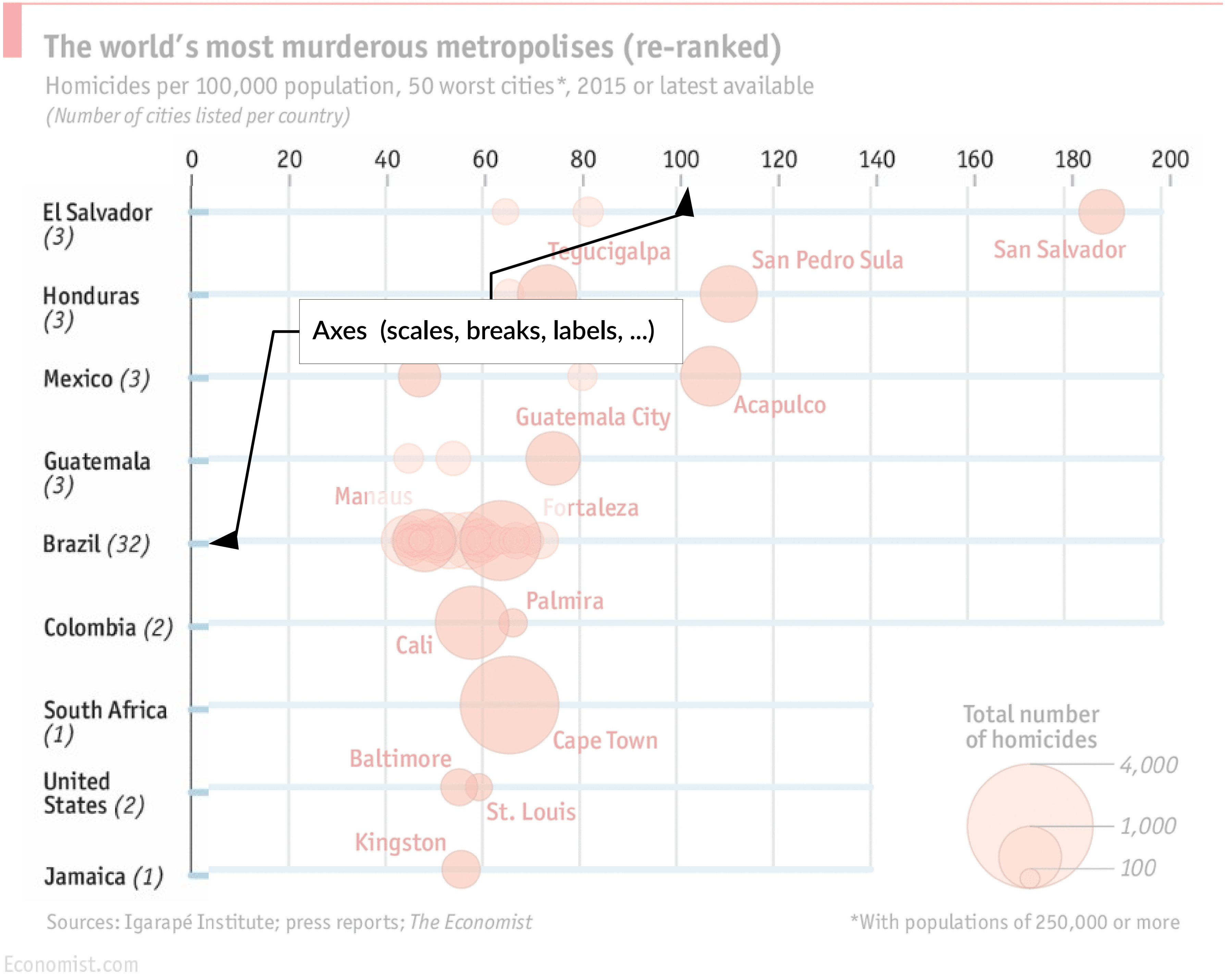
(Number of cities listed per country)

Corresponding legend
Size = total number of homicides
Transparency = 0,5

Sources: Igarapé Institute; press reports; *The Economist*

*With populations of 250,000 or more

Economist.com



The world's most murderous metropolises (re-ranked)

Homicides per 100,000 population, 50 worst cities*, 2015 or latest available

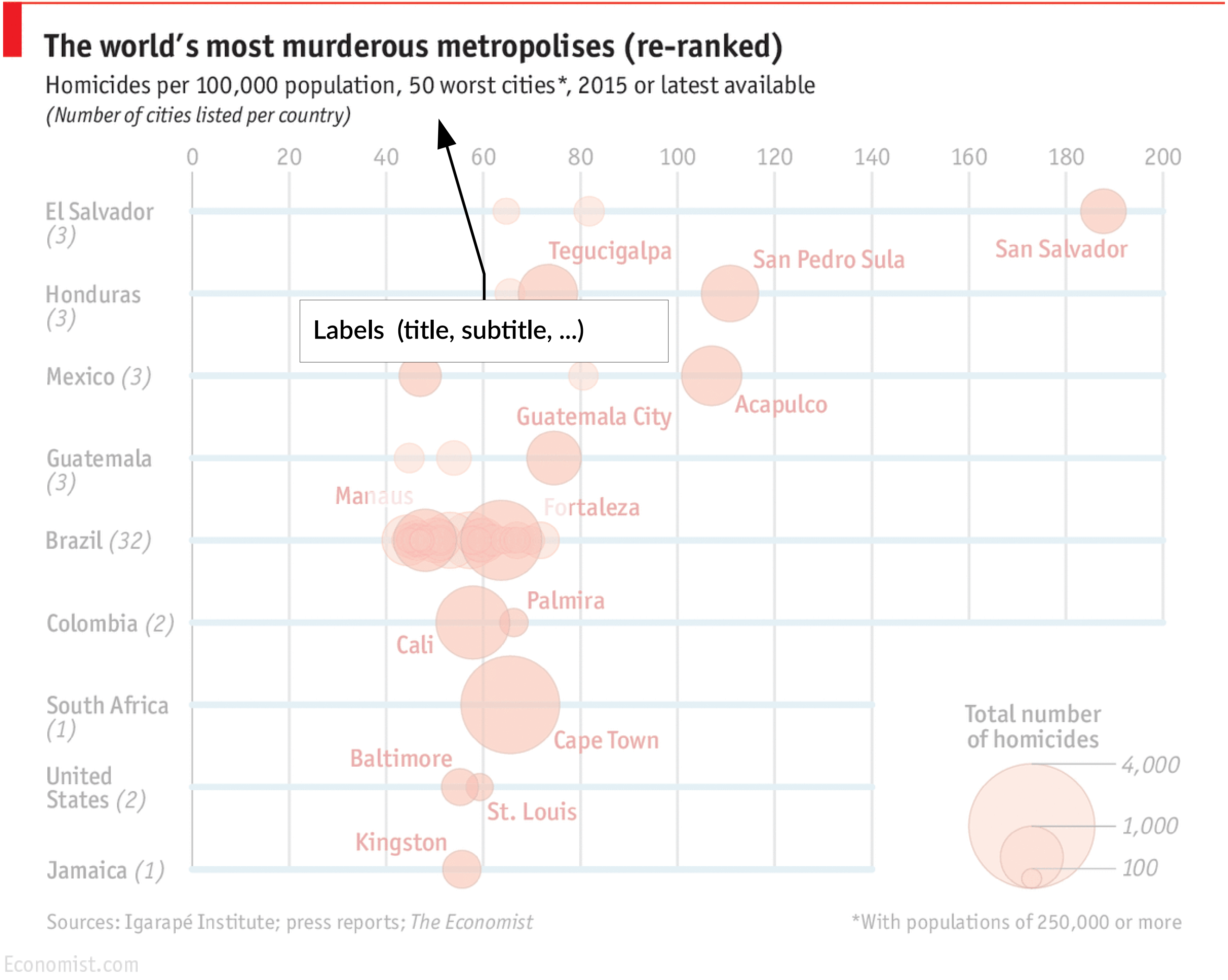
(Number of cities listed per country)

Axes (scales, breaks, labels, ...)

Sources: Igarapé Institute; press reports; *The Economist*

*With populations of 250,000 or more

Economist.com



The world's most murderous metropolises (re-ranked)

Homicides per 100,000 population, 50 worst cities*, 2015 or latest available

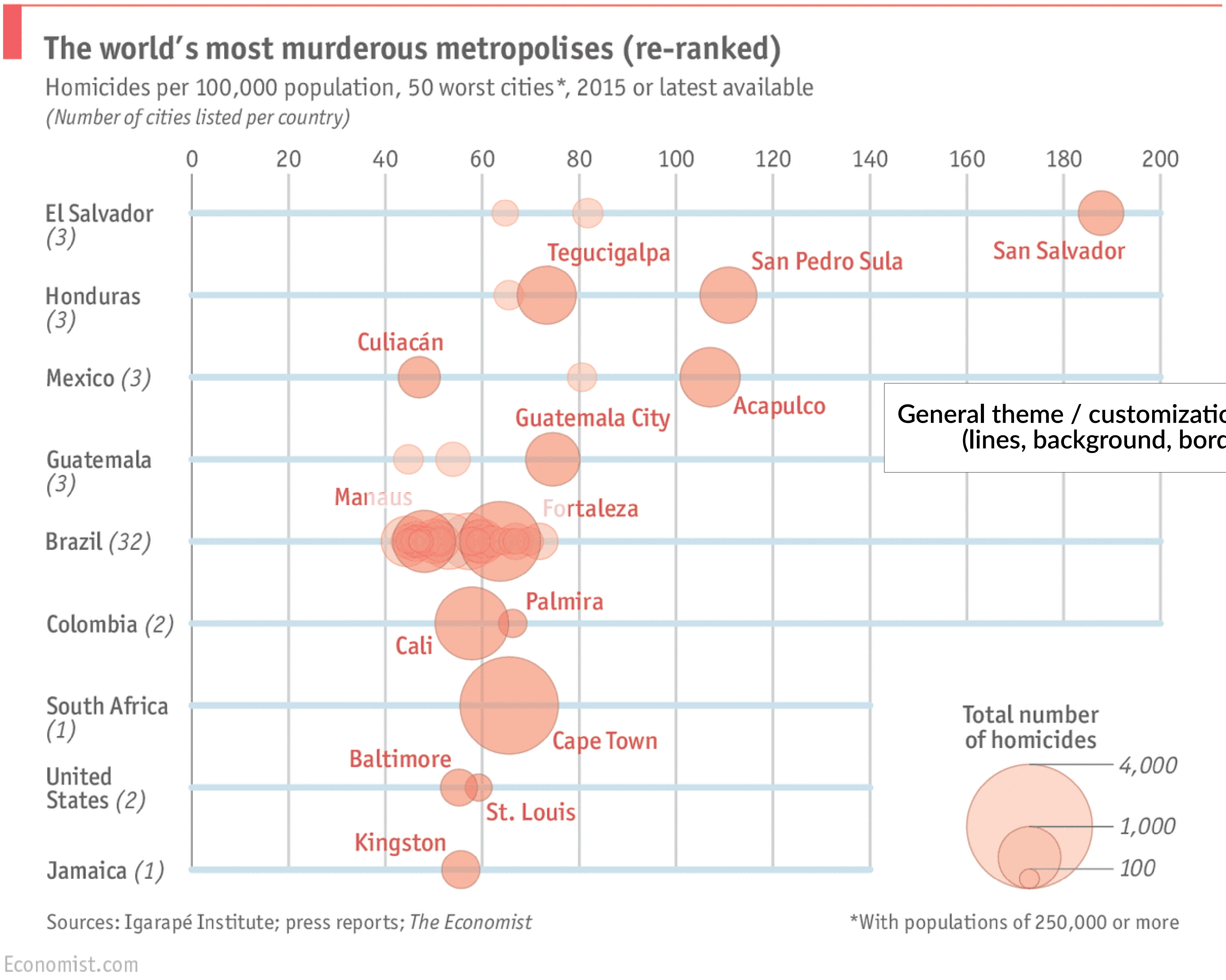
(Number of cities listed per country)

Sources: Igarapé Institute; press reports; *The Economist*

*With populations of 250,000 or more

Economist.com

Labels (title, subtitle, ...)



The world's most murderous metropolises (re-ranked)

Homicides per 100,000 population, 50 worst cities*, 2015 or latest available

(Number of cities listed per country)

Sources: Igarapé Institute; press reports; *The Economist*

*With populations of 250,000 or more

Economist.com

General theme / customization
(lines, background, borders, ...)

Elements of a plot / graphic

1. Data what you find in your data table, your columns, what’s varying

Elements of a plot / graphic

1. Data — what you find in your data table, your columns, what’s varying
2. Representation type — points, lines, boxplot, barplot, …

Elements of a plot / graphic

1. Data	what you find in your data table, your columns, what's varying
2. Representation type	points, lines, boxplot, barplot, …
3. Representation attributes	size, color, shape, transparency, …

Elements of a plot / graphic

1. Data	what you find in your data table, your columns, what’s varying
2. Representation type	points, lines, boxplot, barplot, …
3. Representation attributes	size, color, shape, transparency, …
4. Scales / legends	breaks, labels, transformation, …

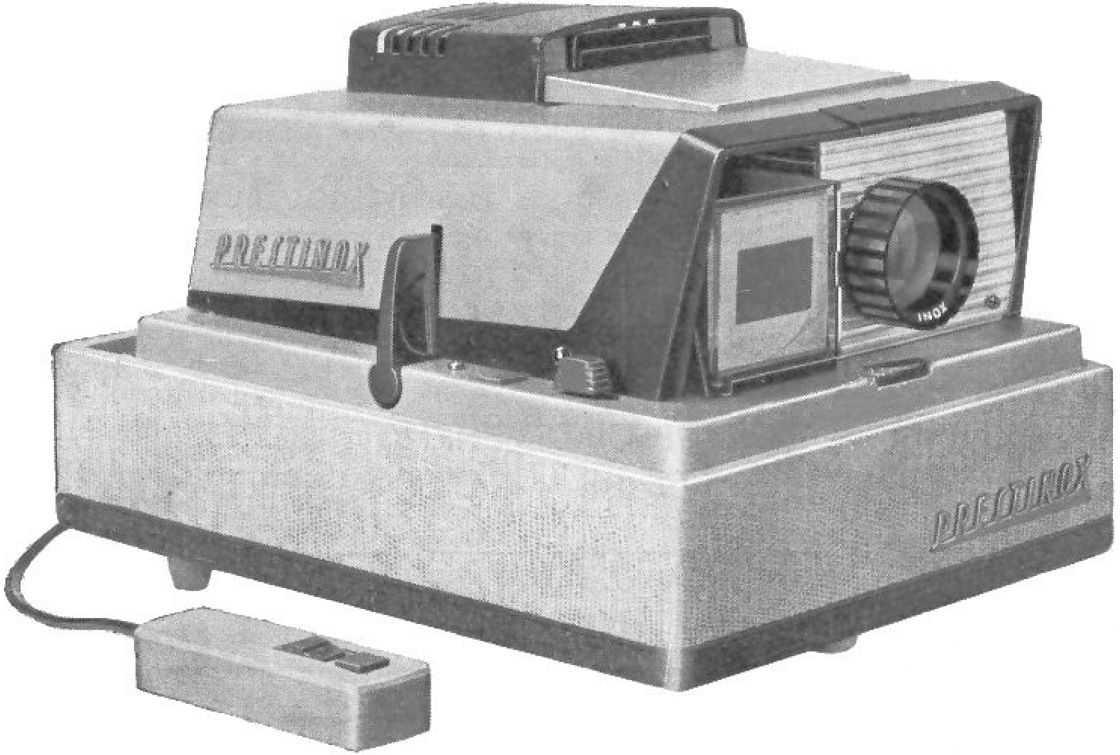
Elements of a plot / graphic

1. Data	what you find in your data table, your columns, what's varying
2. Representation type	points, lines, boxplot, barplot, …
3. Representation attributes	size, color, shape, transparency, …
4. Scales / legends	breaks, labels, transformation, …
5. Global customizing	borders, background, themes, …

Elements of a plot / graphic

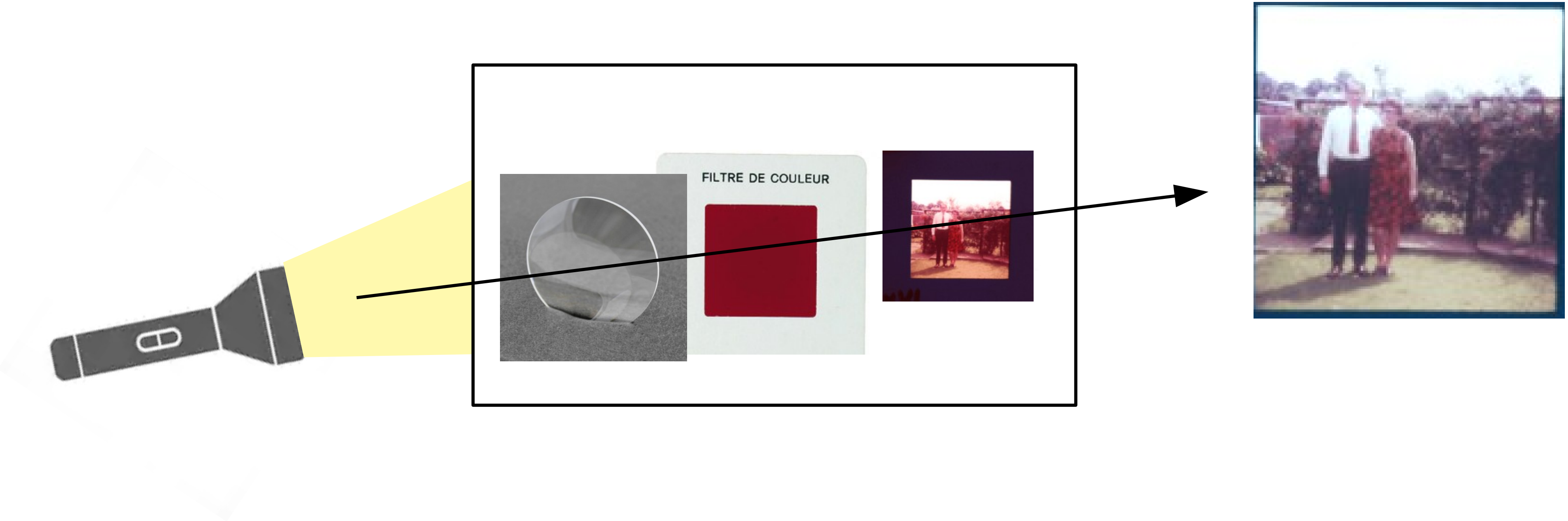
more practically

=



Elements of a plot / graphic

more practically

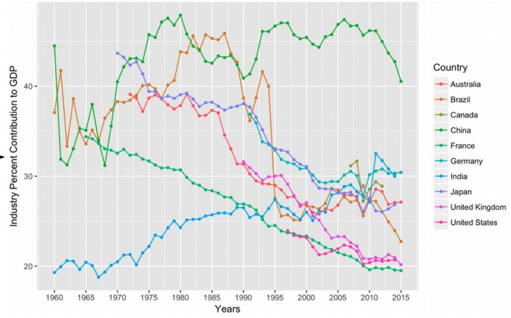


Elements of a plot / graphic

more practically

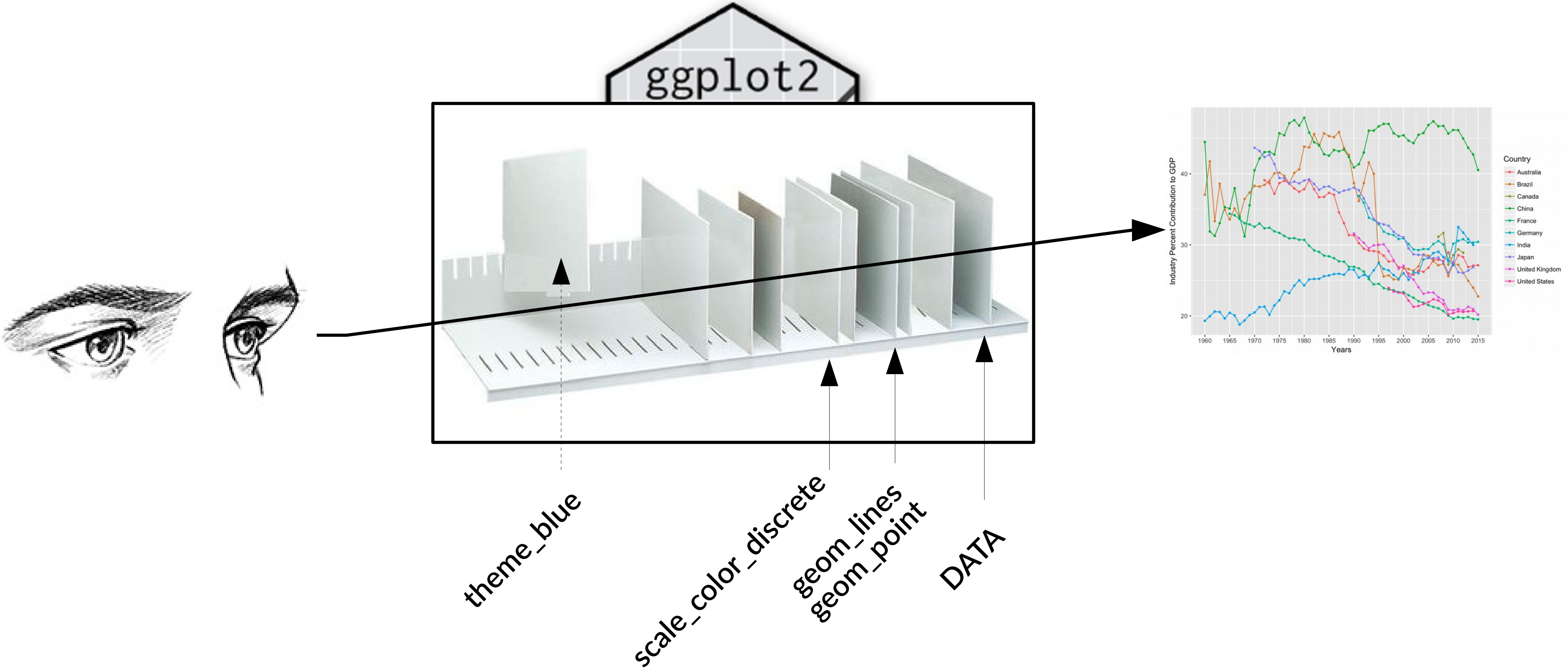
ggplot2

[redacted]



Elements of a plot / graphic

more practically



theme_blue

scale_color_discrete

geom_lines
geom_point

DATA

Elements of a plot / graphic

how to write it with ggplot

```
ggplot(data = TAB, ...) +
     .................. +
     .................. +
     .................. +
     ..................
```

→ your dataset (1.)

1. Data
2. Representation type
3. Representation attributes
4. Scales / legends
5. Global aesthetics

Elements of a plot / graphic

how to write it with ggplot

1. Data
2. Representation type
3. Representation attributes
4. Scales / legends
5. Global aesthetics

```
ggplot(data = TAB, ...) +
     geom_...( ...... ) +
     .................. +
     .................. +
     ..................
```

→ your dataset (1.)

→ at least one geom_ to represent the elements of your dataset (2.)

Elements of a plot / graphic

how to write it with ggplot

1. Data
2. Representation type
3. Representation attributes
4. Scales / legends
5. Global aesthetics

```
ggplot(data = TAB, ...) +
     geom_...( ...... ) +
     .................. +
     scale_color_...( .. ) +
     theme( ......... )
```

→ your dataset (1.)

→ at least one geom_ to represent the elements of your dataset (2.)

→ potentially some other elements for representation (3. 4. and 5.)

Elements of a plot / graphic

aesthetics

Visual properties
of the geom

What you find in
your data table,
your columns,
what's varying

Elements of a plot / graphic

aesthetics

```
ggplot(data = TAB) +
     geom_point(aes(x = Year, y = Production))
```

Visual properties
of the geom

What you find in
your data table,
your columns,
what's varying

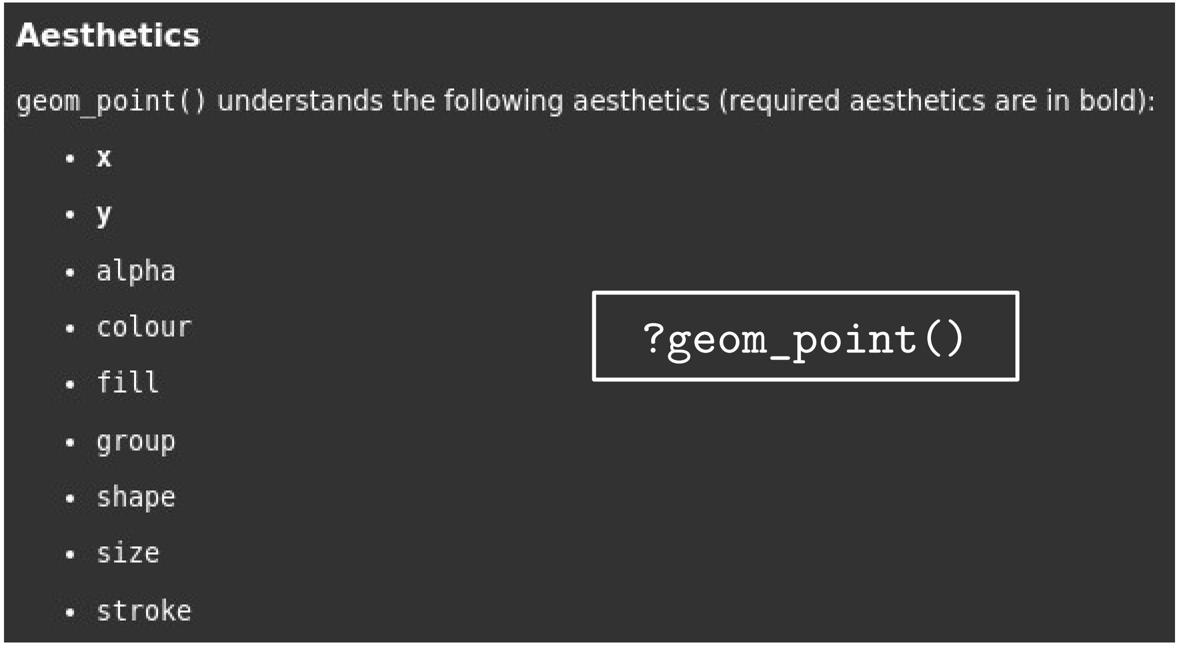
Elements of a plot / graphic

aesthetics

```
ggplot(data = TAB) +
     geom_point(aes(x = Year, y = Production))
```

Visual properties
of the geom

What you find in
your data table,
your columns,
what's varying



Aesthetics

geom_point() understands the following aesthetics (required aesthetics are in bold):

- **x**
- **y**
- alpha
- colour
- fill
- group
- shape
- size
- stroke

```
?geom_point()
```

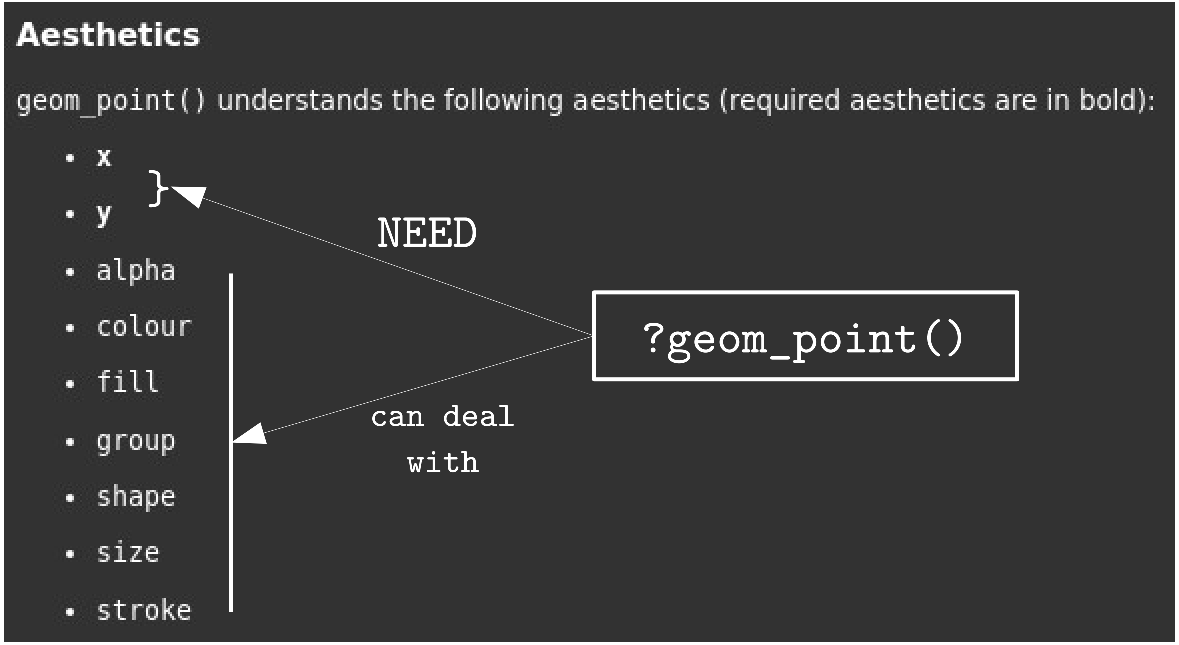
Elements of a plot / graphic

aesthetics

```
ggplot(data = TAB) +
     geom_point(aes(x = Year, y = Production))
```

Visual properties of the geom

What you find in your data table, your columns, what's varying



Aesthetics

geom_point() understands the following aesthetics (required aesthetics are in bold):

- **x**
- **y**
- alpha
- colour
- fill
- group
- shape
- size
- stroke

NEED

?geom_point()

can deal with

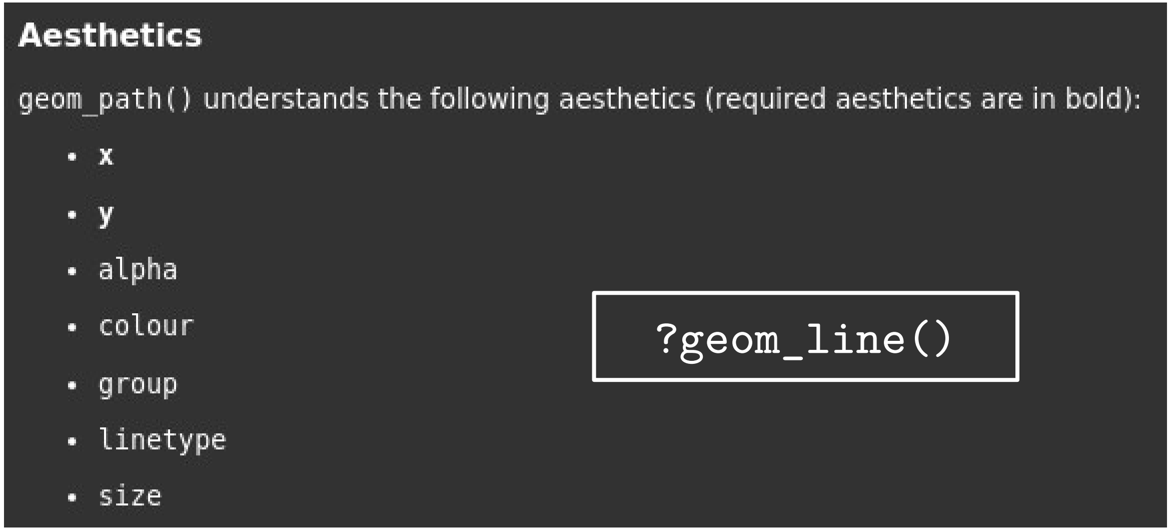
Elements of a plot / graphic

aesthetics

```
ggplot(data = TAB) +
     geom_point(aes(x = Year, y = Production))
```

Visual properties of the geom

What you find in your data table, your columns, what's varying



Aesthetics

geom_path() understands the following aesthetics (required aesthetics are in bold):

- **x**
- **y**
- alpha
- colour
- group
- linetype
- size

```
?geom_line()
```

Elements of a plot / graphic

aesthetics

```
ggplot(data = TAB) +
     geom_point(aes(x = Year, y = Production))
```

Visual properties of the geom

```
ggplot(data = TAB, aes(x = Year, y = Production)) +
     geom_point()
```

What you find in your data table, your columns, what’s varying

Elements of a plot / graphic

aesthetics

```
ggplot(data = TAB) +
     geom_point(aes(x = Year, y = Production))
```

Visual properties of the geom

```
ggplot(data = TAB, aes(x = Year, y = Production)) +
     geom_point()
```

What you find in your data table, your columns, what’s varying

```
ggplot(data = TAB, aes(x = Year, y = Production)) +
     geom_point() +
     geom_line()
```

Elements of a plot / graphic

aesthetics

```
ggplot(data = TAB) +
     geom_point(aes(x = Year, y = Production))
```

Visual properties of the geom

```
ggplot(data = TAB, aes(x = Year, y = Production)) +
     geom_point()
```

What you find in your data table, your columns, what's varying

```
ggplot(data = TAB, aes(x = Year, y = Production)) +
     geom_point() +
     geom_line()
```

```
ggplot(data = TAB) +
     geom_point(aes(x = Year, y = Production)) +
     geom_line() +
```

Elements of a plot / graphic

aesthetics

```
ggplot(data = TAB) +
     geom_point(aes(x = Year, y = Production))
```

Visual properties
of the geom

```
ggplot(data = TAB, aes(x = Year, y = Production)) +
     geom_point()
```

What you find in
your data table,
your columns,
what's varying

```
ggplot(data = TAB, aes(x = Year, y = Production)) +
     geom_point() +
     geom_line()
```

```
~~ggplot(data = TAB) +~~
     ~~geom_point(aes(x = Year, y = Production)) +~~
     ~~geom_line() +~~
```

Elements of a plot / graphic

aesthetics

```
ggplot(data = TAB) +
     geom_point(aes(x = Year, y = Production))
```

Visual properties
of the geom

```
ggplot(data = TAB, aes(x = Year, y = Production)) +
     geom_point()
```

What you find in
your data table,
your columns,
what's varying

```
ggplot(data = TAB, aes(x = Year, y = Production)) +
     geom_point() +
     geom_line()
```

~~ggplot(data = TAB) +~~

~~geom_point(aes(x = Year, y = Production)) +~~

~~geom_line() +~~

```
ggplot(data = TAB, aes(x = Year, y = Production)) +
     geom_point() +
     geom_line(aes(y = Density))
```

Elements of a plot / graphic

aesthetics, melt

Fictitious example :

Year	Murder_rate	Suicide_rate
1992	1,10	7,6
1995	1,30	12,1
1996	1,20	13,8
2000	0,67	5,5
2003	1,10	11,5

Elements of a plot / graphic

aesthetics, melt

Fictitious example :

```
ggplot(data = TAB) +
     geom_point(aes(x = Year, y = Murder_rate))
```

Year	Murder_rate	Suicide_rate
1992	1,10	7,6
1995	1,30	12,1
1996	1,20	13,8
2000	0,67	5,5
2003	1,10	11,5

Elements of a plot / graphic

aesthetics, melt

<u>Fictitious example :</u>

Year	Murder_rate	Suicide_rate
1992	1,10	7,6
1995	1,30	12,1
1996	1,20	13,8
2000	0,67	5,5
2003	1,10	11,5

```
ggplot(data = TAB) +
     geom_point(aes(x = Year, y = Murder_rate))

ggplot(data = TAB, aes(x = Year, y = Murder_rate)) +
     geom_point()
```

Elements of a plot / graphic

aesthetics, melt

Fictitious example :

Year	Murder_rate	Suicide_rate
1992	1,10	7,6
1995	1,30	12,1
1996	1,20	13,8
2000	0,67	5,5
2003	1,10	11,5

```
ggplot(data = TAB) +
     geom_point(aes(x = Year, y = Murder_rate))

ggplot(data = TAB, aes(x = Year, y = Murder_rate)) +
     geom_point()

ggplot(data = TAB, aes(x = Year)) +
     geom_point(aes(y = Murder_rate), color = ‘blue’) +
     geom_point(aes(y = Suicide_rate), color = ‘orange’)
```

Elements of a plot / graphic

aesthetics, melt

Fictitious example :

Year	Murder_rate	Suicide_rate
1992	1,10	7,6
1995	1,30	12,1
1996	1,20	13,8
2000	0,67	5,5
2003	1,10	11,5

↓ melt

Year	event	rate
1992	murder	1,10
1992	suicide	7,6
1995	murder	1,30
1995	suicide	12,1
1996	murder	1,20
1996	suicide	13,8
2000	murder	0,67
2000	suicide	5,5
2003	murder	1,10
2003	suicide	11,5

```
ggplot(data = TAB) +
     geom_point(aes(x = Year, y = Murder_rate))

ggplot(data = TAB, aes(x = Year, y = Murder_rate)) +
     geom_point()

ggplot(data = TAB, aes(x = Year)) +
     geom_point(aes(y = Murder_rate), color = ‘blue’) +
     geom_point(aes(y = Suicide_rate), color = ‘orange’)
```

Elements of a plot / graphic

aesthetics, melt

Fictitious example :

Year	Murder_rate	Suicide_rate
1992	1,10	7,6
1995	1,30	12,1
1996	1,20	13,8
2000	0,67	5,5
2003	1,10	11,5

↓ melt

SAME SCALES

Year	event	rate
1992	murder	1,10
1992	suicide	7,6
1995	murder	1,30
1995	suicide	12,1
1996	murder	1,20
1996	suicide	13,8
2000	murder	0,67
2000	suicide	5,5
2003	murder	1,10
2003	suicide	11,5

```
ggplot(data = TAB) +
     geom_point(aes(x = Year, y = Murder_rate))

ggplot(data = TAB, aes(x = Year, y = Murder_rate)) +
     geom_point()

ggplot(data = TAB, aes(x = Year)) +
     geom_point(aes(y = Murder_rate), color = ‘blue’) +
     geom_point(aes(y = Suicide_rate), color = ‘orange’)
```

Elements of a plot / graphic

aesthetics, melt

Fictitious example :

Year	Murder_rate	Suicide_rate
1992	1,10	7,6
1995	1,30	12,1
1996	1,20	13,8
2000	0,67	5,5
2003	1,10	11,5

↓ melt

column names
= new factor

Year	event	rate
1992	murder	1,10
1992	suicide	7,6
1995	murder	1,30
1995	suicide	12,1
1996	murder	1,20
1996	suicide	13,8
2000	murder	0,67
2000	suicide	5,5
2003	murder	1,10
2003	suicide	11,5

```
ggplot(data = TAB) +
     geom_point(aes(x = Year, y = Murder_rate))

ggplot(data = TAB, aes(x = Year, y = Murder_rate)) +
     geom_point()

ggplot(data = TAB, aes(x = Year)) +
     geom_point(aes(y = Murder_rate), color = ‘blue’) +
     geom_point(aes(y = Suicide_rate), color = ‘orange’)
```

Elements of a plot / graphic

aesthetics, melt

Fictitious example :

Year	Murder_rate	Suicide_rate
1992	1,10	7,6
1995	1,30	12,1
1996	1,20	13,8
2000	0,67	5,5
2003	1,10	11,5

↓ melt

multiplication
of rows
(* nb of columns)

Year	event	rate
1992	murder	1,10
1992	suicide	7,6
1995	murder	1,30
1995	suicide	12,1
1996	murder	1,20
1996	suicide	13,8
2000	murder	0,67
2000	suicide	5,5
2003	murder	1,10
2003	suicide	11,5

```
ggplot(data = TAB) +
     geom_point(aes(x = Year, y = Murder_rate))

ggplot(data = TAB, aes(x = Year, y = Murder_rate)) +
     geom_point()

ggplot(data = TAB, aes(x = Year)) +
     geom_point(aes(y = Murder_rate), color = ‘blue’) +
     geom_point(aes(y = Suicide_rate), color = ‘orange’)
```

Elements of a plot / graphic

aesthetics, melt

<u>Fictitious example :</u>

Year	Murder_rate	Suicide_rate
1992	1,10	7,6
1995	1,30	12,1
1996	1,20	13,8
2000	0,67	5,5
2003	1,10	11,5

↓ melt

Year	event	rate
1992	murder	1,10
1992	suicide	7,6
1995	murder	1,30
1995	suicide	12,1
1996	murder	1,20
1996	suicide	13,8
2000	murder	0,67
2000	suicide	5,5
2003	murder	1,10
2003	suicide	11,5

```
ggplot(data = TAB) +
     geom_point(aes(x = Year, y = Murder_rate))
```

```
ggplot(data = TAB, aes(x = Year, y = Murder_rate)) +
     geom_point()
```

```
ggplot(data = TAB, aes(x = Year)) +
     geom_point(aes(y = Murder_rate), color = ‘blue’) +
     geom_point(aes(y = Suicide_rate), color = ‘orange’)
```

↓

```
ggplot(data = TAB) +
     geom_point(aes(x = Year, y = rate, color = event))
```

```
ggplot(data = TAB, aes(x = Year, y = rate, color = event)) +
     geom_point()
```

Elements of a plot / graphic

aesthetics, melt

Fictitious example :

Year	Murder_rate	Suicide_rate
1992	1,10	7,6
1995	1,30	12,1
1996	1,20	13,8
2000	0,67	5,5
2003	1,10	11,5

↓ melt

Year	event	rate
1992	murder	1,10
1992	suicide	7,6
1995	murder	1,30
1995	suicide	12,1
1996	murder	1,20
1996	suicide	13,8
2000	murder	0,67
2000	suicide	5,5
2003	murder	1,10
2003	suicide	11,5

```
ggplot(data = TAB) +
     geom_point(aes(x = Year, y = Murder_rate))
```

```
ggplot(data = TAB, aes(x = Year, y = Murder_rate)) +
     geom_point()
```

```
ggplot(data = TAB, aes(x = Year)) +
     geom_point(aes(y = Murder_rate), color = ‘blue’) +
     geom_point(aes(y = Suicide_rate), color = ‘orange’)
```

↓

```
ggplot(data = TAB) +
     geom_point(aes(x = Year, y = rate, color = event))
```

```
ggplot(data = TAB, aes(x = Year, y = rate, color = event)) +
     geom_point()
```

Elements of a plot / graphic

aesthetics, melt

Fictitious example :

Year	Murder_rate	Suicide_rate
1992	1,10	7,6
1995	1,30	12,1
1996	1,20	13,8
2000	0,67	5,5
2003	1,10	11,5

↓ melt

Year	event	rate
1992	murder	1,10
1992	suicide	7,6
1995	murder	1,30
1995	suicide	12,1
1996	murder	1,20
1996	suicide	13,8
2000	murder	0,67
2000	suicide	5,5
2003	murder	1,10
2003	suicide	11,5

```
ggplot(data = TAB) +
     geom_point(aes(x = Year, y = Murder_rate))

ggplot(data = TAB, aes(x = Year, y = Murder_rate)) +
     geom_point()

ggplot(data = TAB, aes(x = Year)) +
     geom_point(aes(y = Murder_rate), color = ‘blue’) +
     geom_point(aes(y = Suicide_rate), color = ‘orange’)
```

↓

```
ggplot(data = TAB) +
     geom_point(aes(x = Year, y = rate, color = event))

ggplot(data = TAB, aes(x = Year, y = rate, color = event)) +
     geom_point()
```

Elements of a plot / graphic

aesthetics, melt

Fictitious example :

Year	Murder_rate	Suicide_rate
1992	1,10	7,6
1995	1,30	12,1
1996	1,20	13,8
2000	0,67	5,5
2003	1,10	11,5

↓ melt

Year	event	rate
1992	murder	1,10
1992	suicide	7,6
1995	murder	1,30
1995	suicide	12,1
1996	murder	1,20
1996	suicide	13,8
2000	murder	0,67
2000	suicide	5,5
2003	murder	1,10
2003	suicide	11,5

What about ??

```
ggplot(data = TAB, aes(x = Year, y = rate, color = ‘blue’)) +
     geom_point()
```

```
ggplot(data = TAB, aes(x = Year)) +
     geom_point(aes(y = Murder_rate), color = ‘blue’) +
     geom_point(aes(y = Suicide_rate), color = ‘orange’)
```

↓

```
ggplot(data = TAB) +
     geom_point(aes(x = Year, y = rate, color = event))

ggplot(data = TAB, aes(x = Year, y = rate, color = event)) +
     geom_point()
```

Elements of a plot / graphic

aesthetics, melt

Fictitious example :

Year	Murder_rate	Suicide_rate
1992	1,10	7,6
1995	1,30	12,1
1996	1,20	13,8
2000	0,67	5,5
2003	1,10	11,5

↓ melt

Year	event	rate
1992	murder	1,10
1992	suicide	7,6
1995	murder	1,30
1995	suicide	12,1
1996	murder	1,20
1996	suicide	13,8
2000	murder	0,67
2000	suicide	5,5
2003	murder	1,10
2003	suicide	11,5

```
ggplot(data = TAB) +
     geom_point(aes(x = Year, y = Murder_rate))

ggplot(data = TAB, aes(x = Year, y = Murder_rate)) +
     geom_point()

ggplot(data = TAB, aes(x = Year)) +
     geom_point(aes(y = Murder_rate), color = ‘blue’) +
     geom_point(aes(y = Suicide_rate), color = ‘orange’)
```

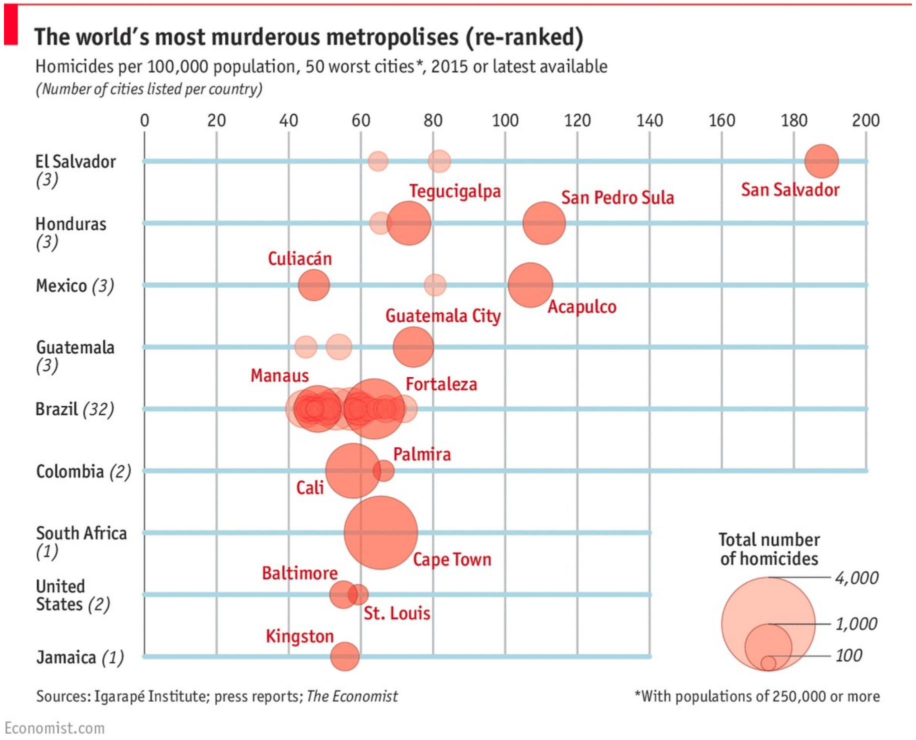
↓

```
ggplot(data = TAB) +
     geom_point(aes(x = Year, y = rate, color = event))

ggplot(data = TAB, aes(x = Year, y = rate, color = event)) +
     geom_point()
```

EXERCISE 1

FILE : EX1_TAB_homicide.csv

Create a simple ggplot graphic with one geometry.



The world's most murderous metropolises (re-ranked)

Homicides per 100,000 population, 50 worst cities*, 2015 or latest available

(Number of cities listed per country)

Sources: Igarapé Institute; press reports; *The Economist*

*With populations of 250,000 or more

Economist.com

Elements of a plot / graphic

geom_ ...

« Basic » geometries :

- point
- line
- boxplot
- histogram

Other geometries :

- bar / *col*
- point / *jitter*
- segment
- boxplot / violin
- abline / *hline* / *vline*
- raster / tile
- contour / density
- errorbar
- label
- ...

Elements of a plot / graphic

geom_ …

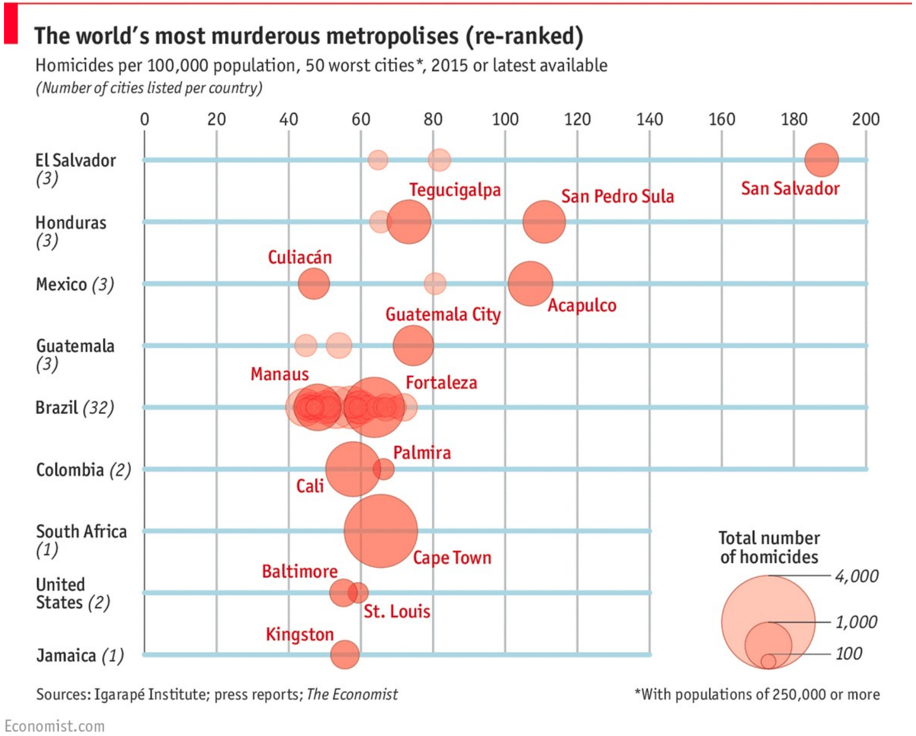
« Basic » geometries :

- point
- line
- boxplot
- histogram

```
ggplot(data = TAB) +
  geom_point(aes( x = Homicides_per100000
                , y = Country
                , size = Homicides_tot )
           , color = ‘red’
           , alpha = 0.5)
```

Other geometries :

- bar / *col*
- point / *jitter*
- segment
- boxplot / violin
- abline / *hline* / *vline*
- raster / tile
- contour / density
- errorbar
- label
- …



The world’s most murderous metropolises (re-ranked)

Homicides per 100,000 population, 50 worst cities*, 2015 or latest available

(Number of cities listed per country)

Sources: Igarapé Institute; press reports; *The Economist*

*With populations of 250,000 or more

Economist.com

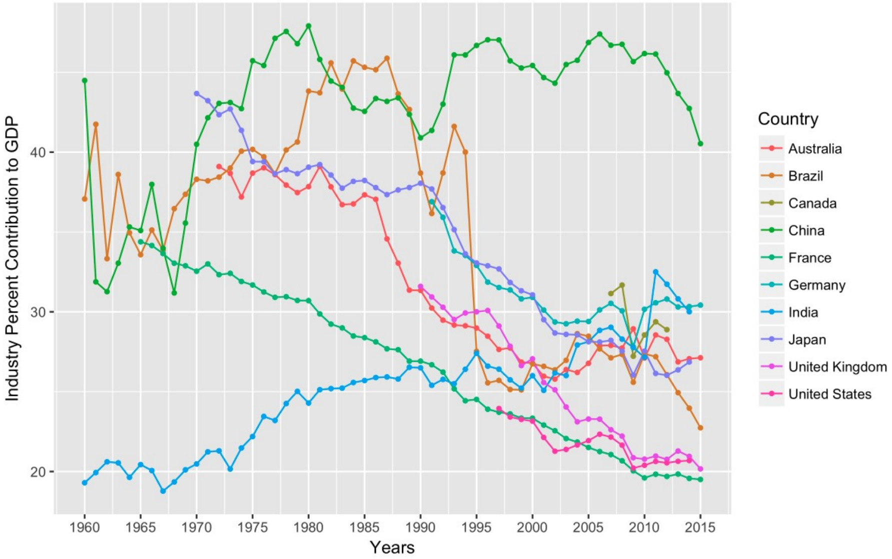
Elements of a plot / graphic

geom_ …

« Basic » geometries :

- point
- line
- boxplot
- histogram

```
ggplot(data = TAB
         , aes( x = Years
               , y = IPC_GDP)
               , color = Country )) +
  geom_line() +
  geom_point()
```



Other geometries :

- bar / *col*
- point / *jitter*
- segment
- boxplot / violin
- abline / *hline* / *vline*
- raster / tile
- contour / density
- errorbar
- label
- …

Elements of a plot / graphic

geom_ ...

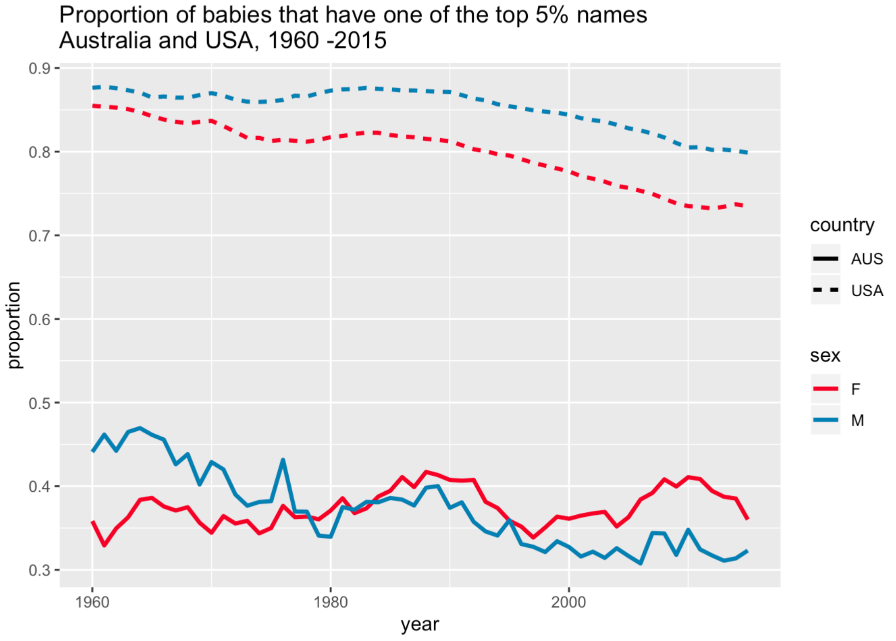
« Basic » geometries :

- point
- line
- boxplot
- histogram

```
ggplot(data = TAB
         , aes( x = year
                , y = proportion)
                , lty = country
                , color = sex )) +
  geom_line()
```

Other geometries :

- bar / *col*
- point / *jitter*
- segment
- boxplot / violin
- abline / *hline* / *vline*
- raster / tile
- contour / density
- errorbar
- label
- ...


Proportion of babies that have one of the top 5% names
Australia and USA, 1960 -2015

Elements of a plot / graphic

geom_ …

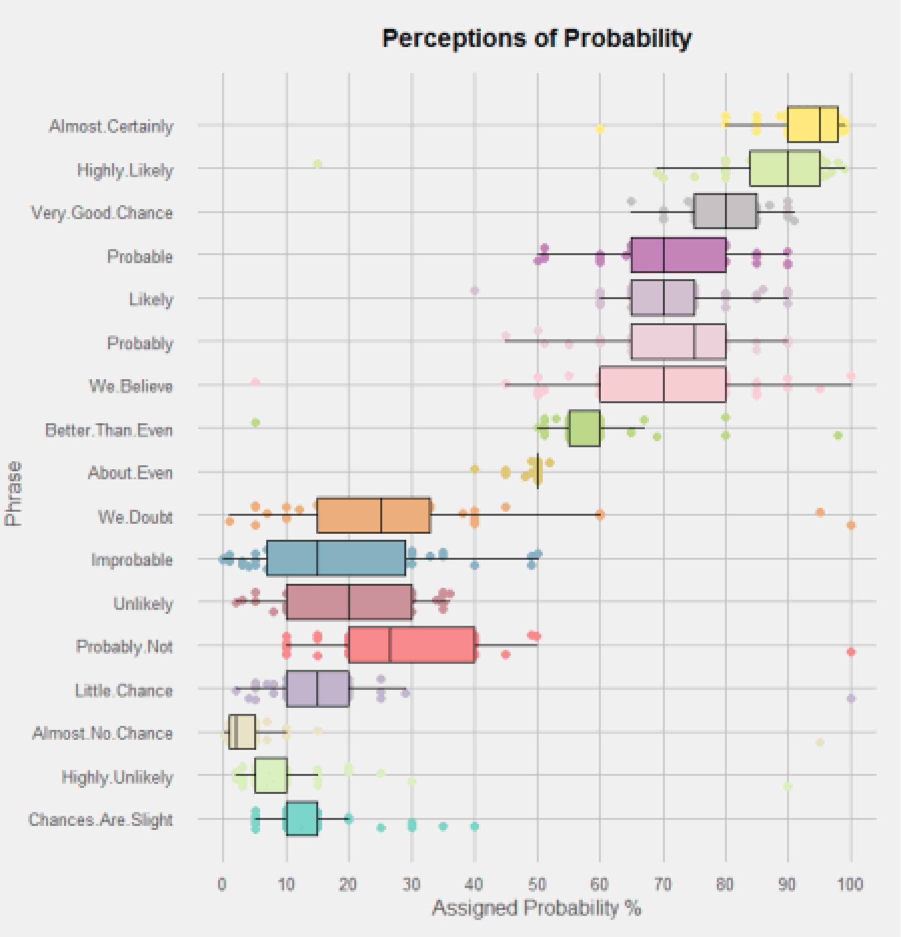
« Basic » geometries :

- point
- line
- boxplot
- histogram

```
ggplot(data = TAB
       , aes( x = Phrase
            , y = Probability )) +
  geom_point(aes( color = Phrase )) +
  geom_boxplot(aes( fill = Phrase )) +
  coord_flip()
```

Other geometries :

- bar / *col*
- point / *jitter*
- segment
- boxplot / violin
- abline / *hline* / *vline*
- raster / tile
- contour / density
- errorbar
- label
- …



Elements of a plot / graphic

geom_ …

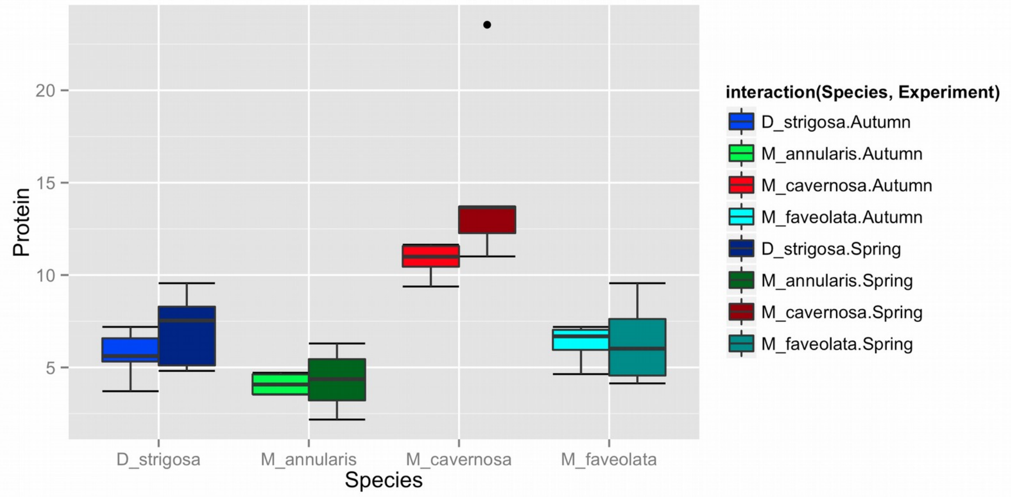
« Basic » geometries :

- point
- line
- boxplot
- histogram

```
ggplot(data = TAB
       , aes( x = Species
             , y = Protein
             , fill = interaction(Species, Experiment) )) +
  geom_boxplot()
```

Other geometries :

- bar / *col*
- point / *jitter*
- segment
- boxplot / violin
- abline / *hline / vline*
- raster / tile
- contour / density
- errorbar
- label
- …



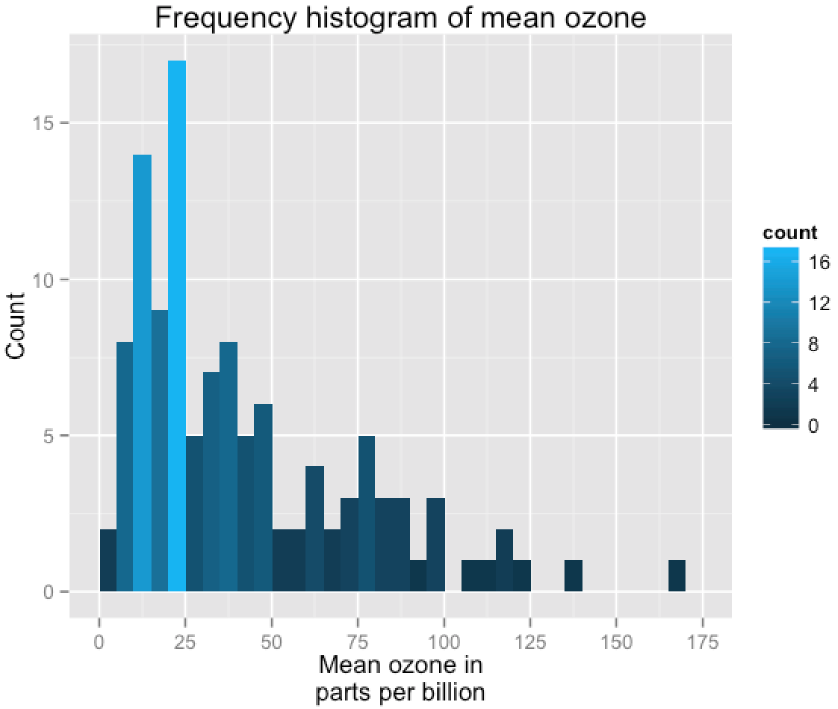
Elements of a plot / graphic

geom_ ...

« Basic » geometries :

- point
- line
- boxplot
- histogram

```
ggplot(data = TAB
       , aes( x = Ozone )) +
  geom_histogram(aes( fill = ..count.. ))
```



Other geometries :

- bar / *col*
- point / *jitter*
- segment
- boxplot / violin
- abline / *hline* / *vline*
- raster / tile
- contour / density
- errorbar
- label
- ...

Elements of a plot / graphic

geom_ …

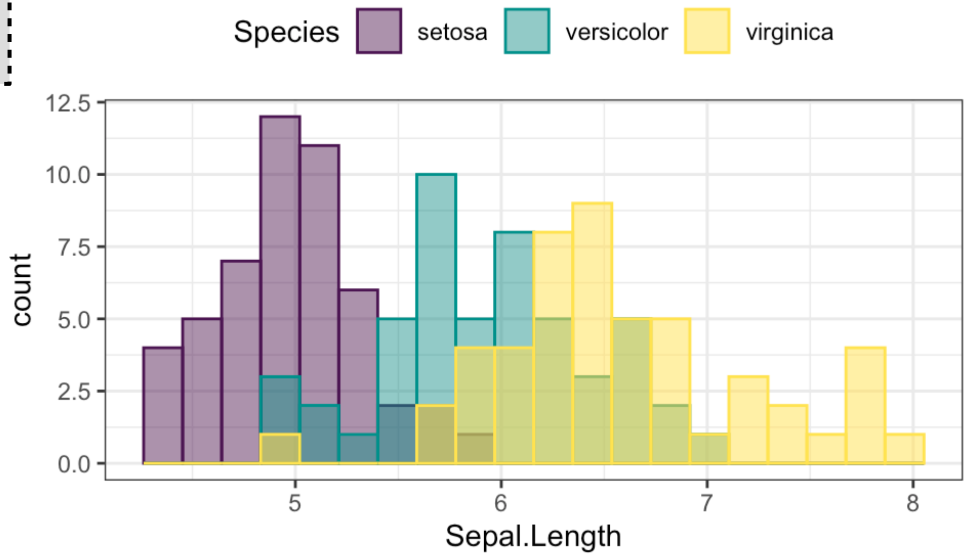
« Basic » geometries :

- point
- line
- boxplot
- histogram

Other geometries :

- bar / *col*
- point / *jitter*
- segment
- boxplot / violin
- abline / *hline* / *vline*
- raster / tile
- contour / density
- errorbar
- label
- …

```
ggplot(data = TAB
       , aes( x = Sepal.Length
             , fill = Species
             , color = Species )) +
  geom_histogram(aes( y = ..count.. )
                 , position = ‘identity’)
                 , alpha = 0.5)
```



Elements of a plot / graphic

geom_ ...

<u>« Basic » geometries :</u>

- point
- line
- boxplot
- histogram

<u>Other geometries :</u>

- bar / *col*
- point / *jitter*
- segment
- boxplot / violin
- abline / *hline* / *vline*
- raster / tile
- contour / density
- errorbar
- label
- ...

Elements of a plot / graphic

geom_ …

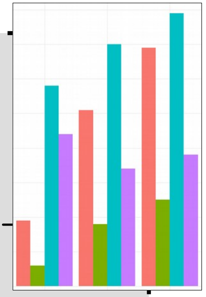
« Basic » geometries :

- point
- line
- boxplot
- histogram

Other geometries :

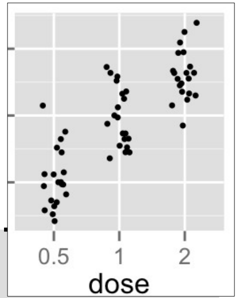
- bar / *col*
- point / *jitter*
- segment
- boxplot / violin
- abline / *hline* / *vline*
- raster / tile
- contour / density
- errorbar
- label
- …

```
ggplot(data = TAB) +
  geom_bar(aes( x = Country ))


    ------------------------------------------


ggplot(data = TAB) +
  geom_bar(aes( x = Country, y = Nb_cities )
           , stat = ‘identity’)

ggplot(data = TAB) +
  geom_col(aes( x = Country, y = Nb_cities ))
```



Elements of a plot / graphic

geom_ …



« Basic » geometries :

- point
- line
- boxplot
- histogram

Other geometries :

- bar / *col*
- point / *jitter*
- segment
- boxplot / violin
- abline / *hline* / *vline*
- raster / tile
- contour / density
- errorbar
- label
- …

```
ggplot(data = TAB) +
  geom_bar(aes( x = Country ))

    ------------------------------------

ggplot(data = TAB) +
  geom_bar(aes( x = Country, y = Nb_cities )
           , stat = ‘identity’)

ggplot(data = TAB) +
  geom_col(aes( x = Country, y = Nb_cities ))
```

```
ggplot(data = TAB) +
  geom_point(aes( x = Country, y = Size_cities )
           , position = jitter)

ggplot(data = TAB) +
  geom_point(aes( x = Country, y = Size_cities )
           , position = position_jitter(width = 0.1
                                       , height = 0.1))

ggplot(data = TAB) +
  geom_jitter(aes( x = Country, y = Size_cities ))
```

Elements of a plot / graphic

geom_ …

« Basic » geometries :

- point
- line
- boxplot
- histogram

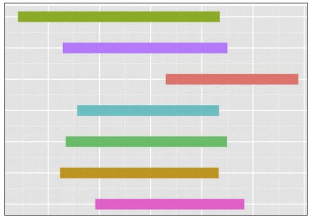
Other geometries :

- bar / *col*
- point / *jitter*
- segment
- boxplot / violin
- abline / *hline* / *vline*
- raster / tile
- contour / density
- errorbar
- label
- …

```
ggplot(data = TAB) +
  geom_bar(aes( x = Country ))


  ----------------------------------------


ggplot(data = TAB) +
  geom_bar(aes( x = Country, y = Nb_cities )
          , stat = ‘identity’)


ggplot(data = TAB) +
  geom_col(aes( x = Country, y = Nb_cities ))
```

```
ggplot(data = TAB) +
  geom_point(aes( x = Country, y = Size_cities )
            , position = jitter)


ggplot(data = TAB) +
  geom_point(aes( x = Country, y = Size_cities )
            , position = position_jitter(width = 0.1
                                        , height = 0.1))


ggplot(data = TAB) +
  geom_jitter(aes( x = Country, y = Size_cities ))
```

```
ggplot(data = TAB) +
  geom_segment(aes( x = Country
                  , xend = Country
                  , y = Production_min
                  , yend = Production_max ))
```



Elements of a plot / graphic

geom_ ...

« Basic » geometries :

- point
- line
- boxplot
- histogram

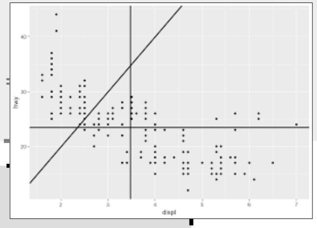
Other geometries :

- bar / *col*
- point / *jitter*
- segment
- boxplot / violin
- abline / *hline / vline*
- raster / tile
- contour / density
- errorbar
- label
- ...

```
ggplot(data = TAB) +
  geom_bar(aes( x = Country ))

  -------------------------------------

ggplot(data = TAB) +
  geom_bar(aes( x = Country, y = Nb_cities )
           , stat = ‘identity’)

ggplot(data = TAB) +
  geom_col(aes( x = Country, y = Nb_cities ))
```

```
ggplot(data = TAB) +
  geom_segment(aes( x = Country
                  , xend = Country
                  , y = Production_min
                  , yend = Production_max ))
```

```
ggplot(data = TAB) +
  geom_point(aes( x = Country, y = Size_cities )
             , position = jitter)

ggplot(data = TAB) +
  geom_point(aes( x = Country, y = Size_cities )
             , position = position_jitter(width = 0.1
                                         , height = 0.1))

ggplot(data = TAB) +
  geom_jitter(aes( x = Country, y =
```



```
ggplot(data = TAB) +
  geom_abline(slope = 1, intercept = 0) +
  geom_hline(yintercept = 10, lty = 2) +
  geom_vline(xintercept = seq(0,5,1))
```

Elements of a plot / graphic

geom_ …

<u>« Basic » geometries :</u>

- point
- line
- boxplot
- histogram

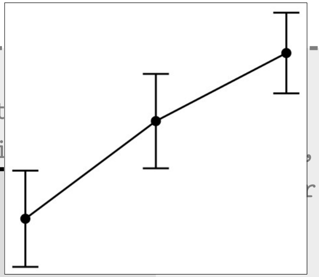
<u>Other geometries :</u>

- bar / *col*
- point / *jitter*
- segment
- boxplot / violin
- abline / *hline* / *vline*
- raster / tile
- contour / density
- errorbar
- label
- …

```
ggplot(data = TAB) +
  geom_bar(aes( x = Country ))

    --------------------

ggplot(data = TAB) +
  geom_bar(aes( x = C
           , stat = '

ggplot(data = TAB) +
  geom_col(aes( x = Country, y = Nb_cities ))
```

```
ggplot(dat
  geom_poi                 , y = Size_cities )
                          r)

                         = Country, y = Size_cities )
                         on = position_jitter(width = 0.1
                                              , height = 0.1))

ggplot(data = TAB) +
  geom_jitter(aes( x = Country, y = Size_cities ))
```

```
ggplot(data = TAB) +
  geom_errorbar(aes( x = Country
                   , y = Size_mean
                   , ymin = Size_min
                   , ymax = Size_max ))
```



```
ggplot(data = TAB) +
  geom_segment(aes( x = Country
                  , xend = Country
                  , y = Production_min
                  , yend = Production_max ))
```

```
ggplot(data = TAB) +
  geom_abline(slope = 1, intercept = 0) +
  geom_hline(yintercept = 10, lty = 2) +
  geom_vline(xintercept = seq(0,5,1))
```

Elements of a plot / graphic

geom_ …

« Basic » geometries :

- point
- line
- boxplot
- histogram

Other geometries :

- bar / *col*
- point / *jitter*
- segment
- boxplot / violin
- abline / *hline / vline*
- raster / tile
- contour / density
- errorbar
- label
- …

```
ggplot(data = TAB) +
  geom_bar(aes( x = Country ))

   --------------------

ggplot(data = TAB) +
  geom_bar(aes( x = Co
           , stat = ‘i

ggplot(data = TAB) +
  geom_co
```

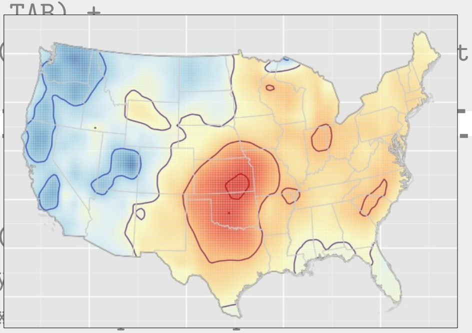
```
ggplot(data = TAB) +
  geom_point(aes( x = Country, y = Size_cities )
                                                  n = jitter)

                  = Country, y = Size_cities )
                  n = position_jitter(width = 0.1
                                           , height = 0.1))

ggplot(data = TAB) +
  r(                                              ties ))
```

```
ggplot(data = TAB) +
  geom_errorbar(aes( x = Country
                   , y = Size_mean
                   , ymin = Size_min
                   , ymax = Size_max ))
```

```
ggplot(da
  geom_se

                     , y = Production_min
                     , yend = Production_max ))
```

```
=
e(
(y
geom_vline(x
```

```
ggplot(data = TAB, aes( x = x
                      , y = y
                      , fill = Correlation )) +
  geom_raster() +
  geom_contour()
```



Elements of a plot / graphic

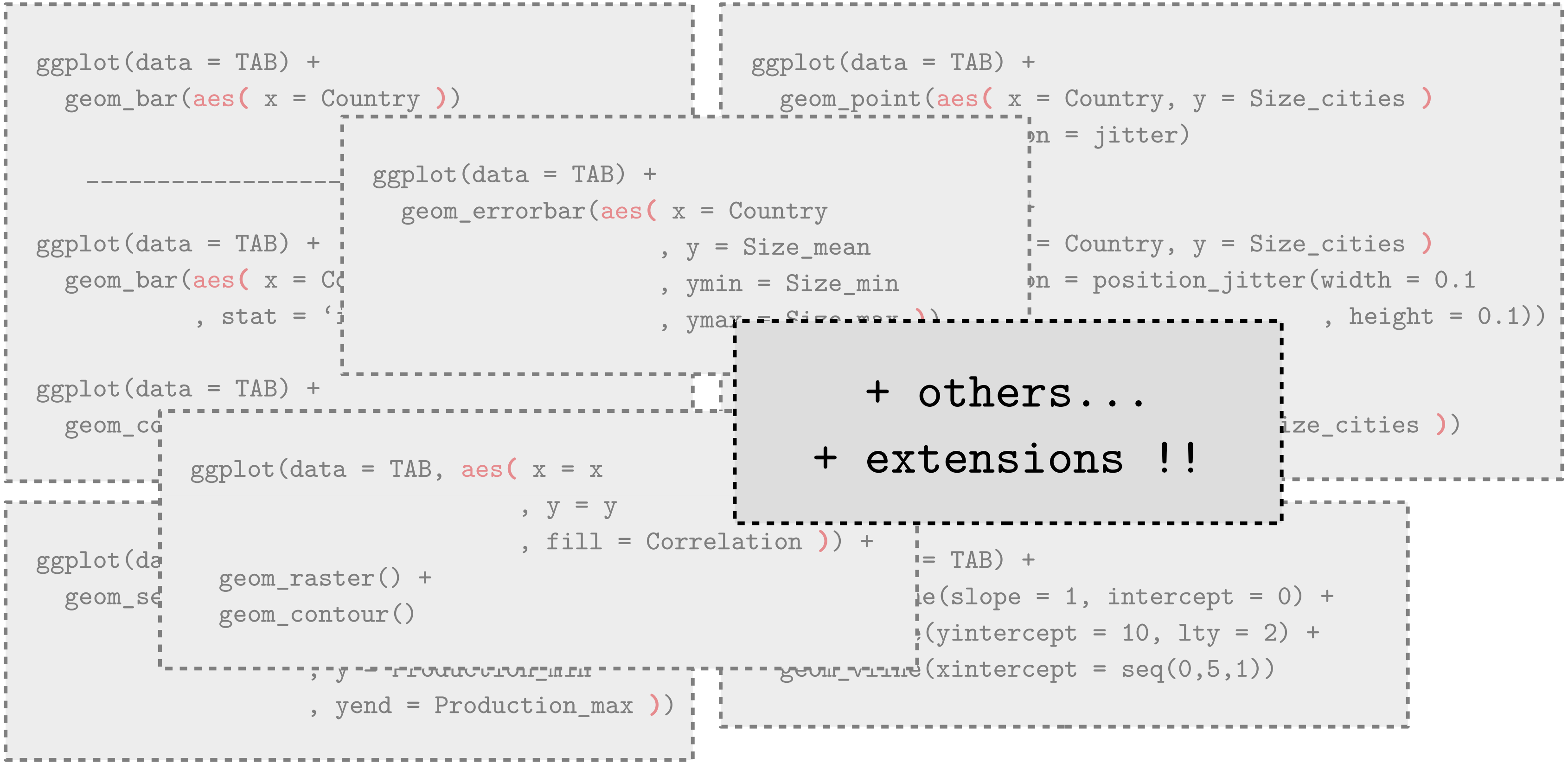
geom_ …

« Basic » geometries :

- point
- line
- boxplot
- histogram

Other geometries :

- bar / *col*
- point / *jitter*
- segment
- boxplot / violin
- abline / *hline* / *vline*
- raster / tile
- contour / density
- errorbar
- label
- …



```
ggplot(data = TAB) +
  geom_bar(aes( x = Country ))

   -----------------

ggplot(data = TAB) +
  geom_bar(aes( x = C
              , stat = ‘

ggplot(data = TAB) +
  geom_co
```

```
ggplot(data = TAB) +
  geom_point(aes( x = Country, y = Size_cities )
                                n = jitter)

                 = Country, y = Size_cities )
                 n = position_jitter(width = 0.1
                                   , height = 0.1))

                              ize_cities ))
```

```
ggplot(data = TAB) +
  geom_errorbar(aes( x = Country
                   , y = Size_mean
                   , ymin = Size_min
                   , ymax = Size_max ))
```

```
ggplot(data = TAB, aes( x = x
                      , y = y
                      , fill = Correlation )) +
  geom_raster() +
  geom_contour()
```

```
ggplot(da
  geom_se
                , y = Production_min
                , yend = Production_max ))
```

```
             = TAB) +
            e(slope = 1, intercept = 0) +
            (yintercept = 10, lty = 2) +
  geom_vline(xintercept = seq(0,5,1))
```

```
+ others...
+ extensions !!
```

EXERCISE 2

<u>FILE :</u> EX1_TAB_homicide.csv

Try a new geometry (boxplot).
Use several geometries.

<u>FILE :</u> EX2_TAB_countries.csv

Understand geom_bar / geom_col and that ggplot can manipulate and extract information from your data.

Elements of a plot / graphic

facet, melt

Fictitious example :

Year	Murder_rate	Suicide_rate
1992	1,10	7,6
1995	1,30	12,1
1996	1,20	13,8
2000	0,67	5,5
2003	1,10	11,5

↓ melt

Year	event	rate
1992	murder	1,10
1992	suicide	7,6
1995	murder	1,30
1995	suicide	12,1
1996	murder	1,20
1996	suicide	13,8
2000	murder	0,67
2000	suicide	5,5
2003	murder	1,10
2003	suicide	11,5

Elements of a plot / graphic

facet, melt

Fictitious example :

```
ggplot(data = TAB, aes(x = Year, y = rate, color = event)) +
     geom_point()
```

Year	Murder_rate	Suicide_rate
1992	1,10	7,6
1995	1,30	12,1
1996	1,20	13,8
2000	0,67	5,5
2003	1,10	11,5

↓ melt

Year	event	rate
1992	murder	1,10
1992	suicide	7,6
1995	murder	1,30
1995	suicide	12,1
1996	murder	1,20
1996	suicide	13,8
2000	murder	0,67
2000	suicide	5,5
2003	murder	1,10
2003	suicide	11,5

Elements of a plot / graphic

facet, melt

Fictitious example :

Year	Murder_rate	Suicide_rate
1992	1,10	7,6
1995	1,30	12,1
1996	1,20	13,8
2000	0,67	5,5
2003	1,10	11,5

↓ melt

Year	event	rate
1992	murder	1,10
1992	suicide	7,6
1995	murder	1,30
1995	suicide	12,1
1996	murder	1,20
1996	suicide	13,8
2000	murder	0,67
2000	suicide	5,5
2003	murder	1,10
2003	suicide	11,5

```
ggplot(data = TAB, aes(x = Year, y = rate, color = event)) +
      geom_point()
```

```
ggplot(data = TAB, aes(x = Year, y = rate)) +
      geom_point() +
      facet_wrap(~ event)
```

Elements of a plot / graphic

facet, melt

Fictitious example :

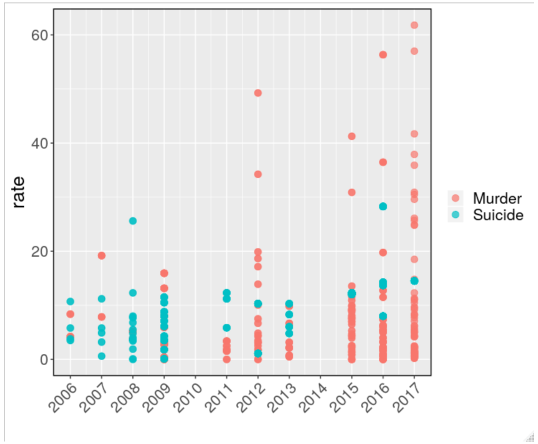
Year	Murder_rate	Suicide_rate
1992	1,10	7,6
1995	1,30	12,1
1996	1,20	13,8
2000	0,67	5,5
2003	1,10	11,5

↓ melt

Year	event	rate
1992	murder	1,10
1992	suicide	7,6
1995	murder	1,30
1995	suicide	12,1
1996	murder	1,20
1996	suicide	13,8
2000	murder	0,67
2000	suicide	5,5
2003	murder	1,10
2003	suicide	11,5

```
ggplot(data = TAB, aes(x = Year, y = rate, color = event)) +
      geom_point()
```

```
ggplot(data = TAB, aes(x = Year, y = rate)) +
      geom_point() +
      facet_wrap(~ event)
```



Elements of a plot / graphic

facet, melt

Fictitious example :

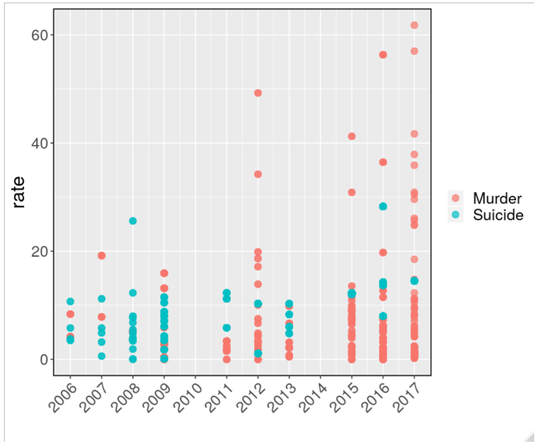
Year	Murder_rate	Suicide_rate
1992	1,10	7,6
1995	1,30	12,1
1996	1,20	13,8
2000	0,67	5,5
2003	1,10	11,5

↓ melt

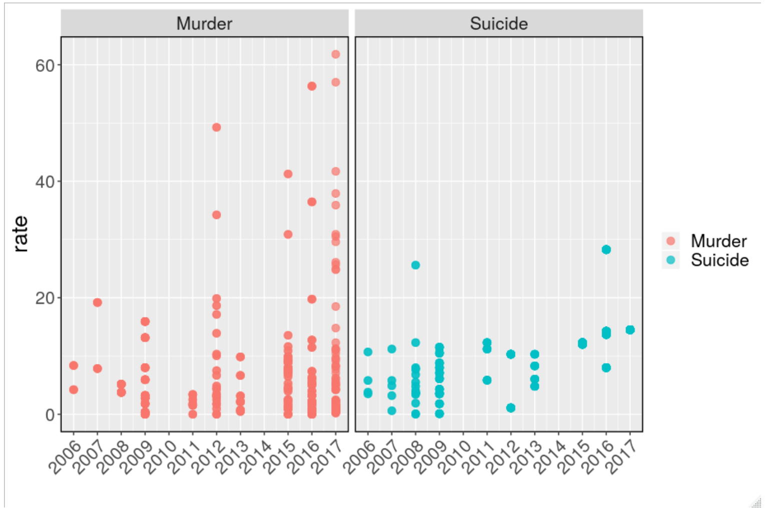
Year	event	rate
1992	murder	1,10
1992	suicide	7,6
1995	murder	1,30
1995	suicide	12,1
1996	murder	1,20
1996	suicide	13,8
2000	murder	0,67
2000	suicide	5,5
2003	murder	1,10
2003	suicide	11,5

```
ggplot(data = TAB, aes(x = Year, y = rate, color = event)) +
     geom_point()
```

```
ggplot(data = TAB, aes(x = Year, y = rate)) +
     geom_point() +
     facet_wrap(~ event)
```



→



Elements of a plot / graphic

facet, melt

`facet_wrap( … )`

- ~ 1 or 2 discrete values (factors)
- one after another, minimizing the number of rows and columns
- scales can be changed on both axes

`facet_grid( … )`

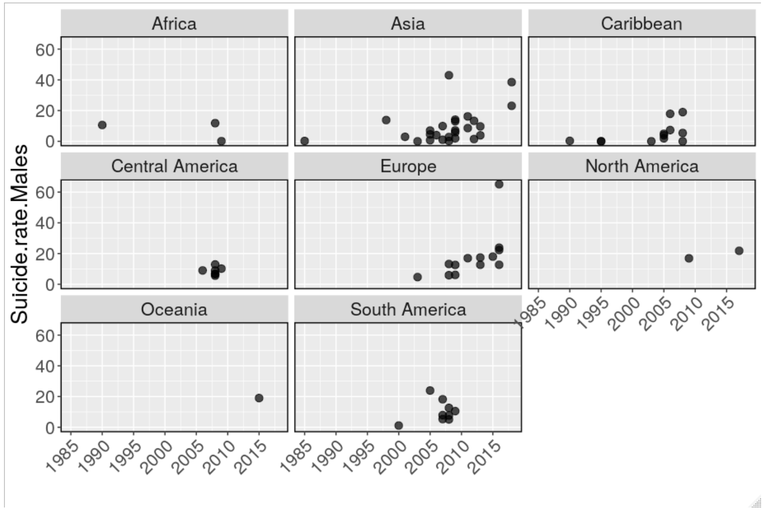
- ~ 1 or 2 discrete values (factors)
- each factor is either displayed on rows or on columns
- scales can be changed on same axes than facets

Elements of a plot / graphic

facet, melt

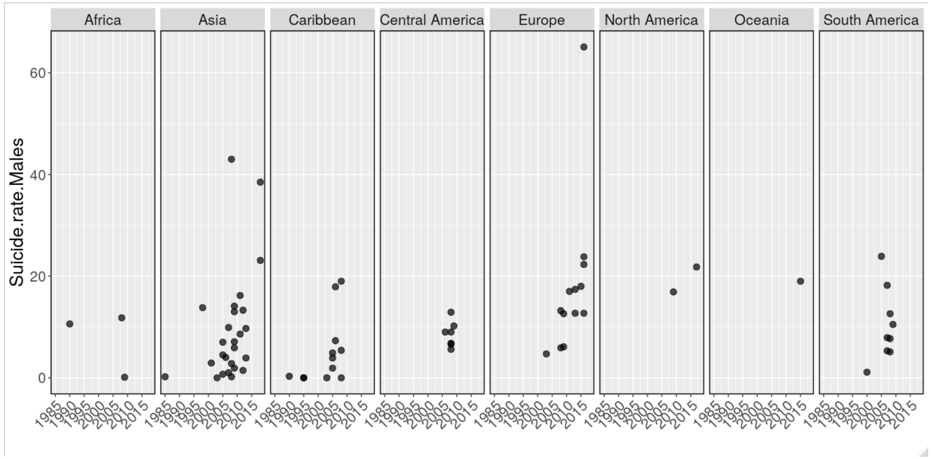
`facet_wrap( … )`

- ~ 1 or 2 discrete values (factors)
- one after another, minimizing the number of rows and columns
- scales can be changed on both axes



`facet_grid( … )`

- ~ 1 or 2 discrete values (factors)
- each factor is either displayed on rows or on columns
- scales can be changed on same axes than facets

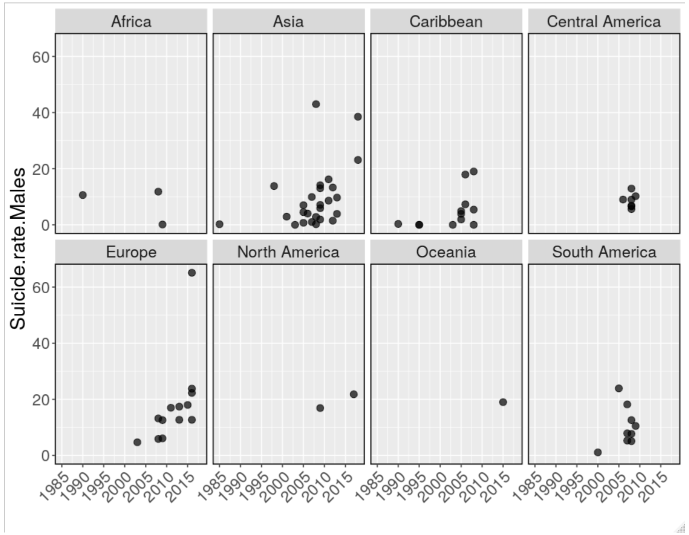


Elements of a plot / graphic

facet, melt

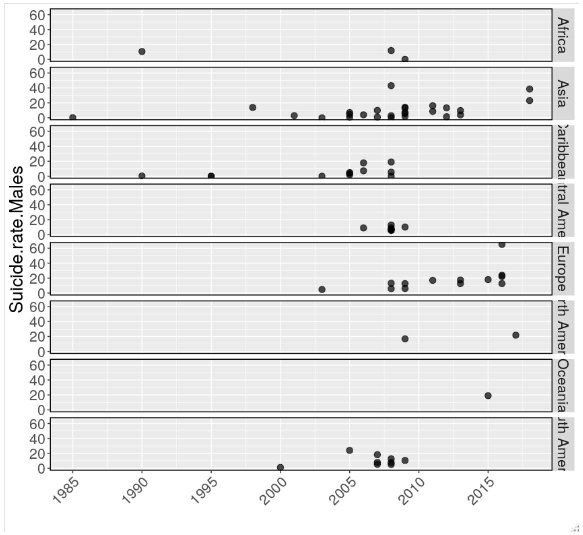
facet_wrap(…)

- ~ 1 or 2 discrete values (factors)
- one after another, minimizing the number of rows and columns
- scales can be changed on both axes



facet_grid(…)

- ~ 1 or 2 discrete values (factors)
- each factor is either displayed on rows or on columns
- scales can be changed on same axes than facets

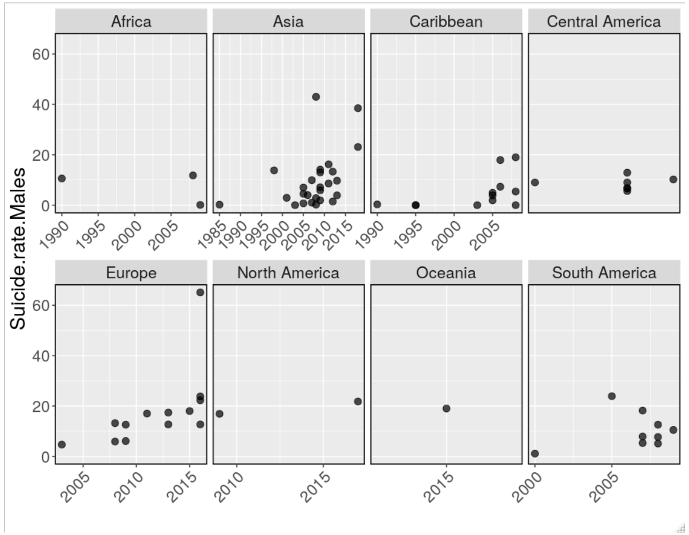


Elements of a plot / graphic

facet, melt

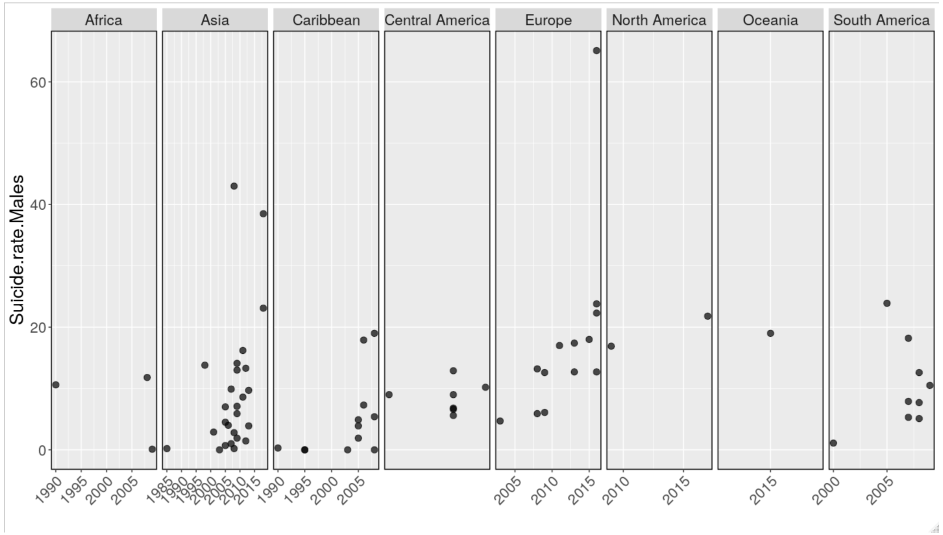
`facet_wrap( … )`

- ~ 1 or 2 discrete values (factors)
- one after another, minimizing the number of rows and columns
- scales can be changed on both axes



`facet_grid( … )`

- ~ 1 or 2 discrete values (factors)
- each factor is either displayed on rows or on columns
- scales can be changed on same axes than facets

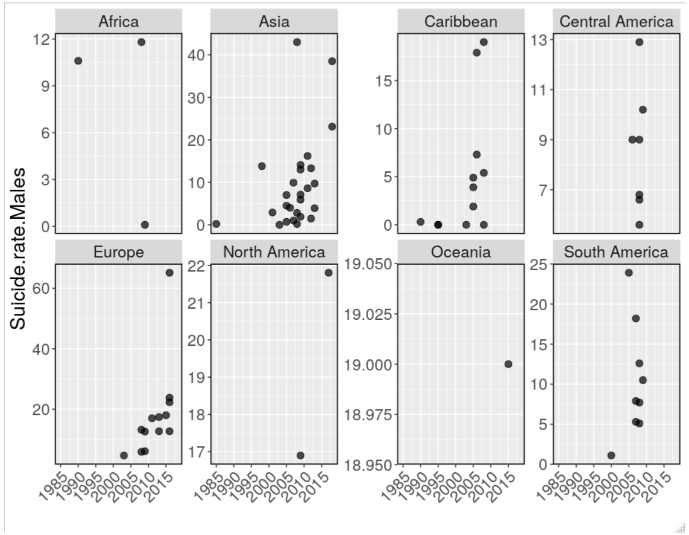


Elements of a plot / graphic

facet, melt

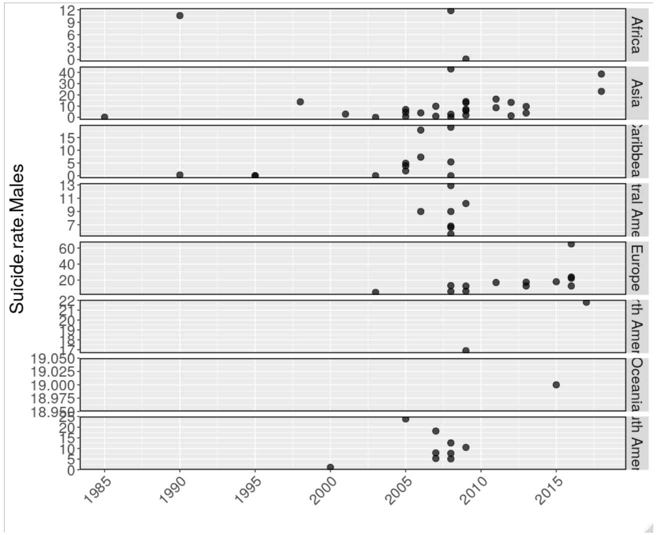
`facet_wrap( … )`

- ~ 1 or 2 discrete values (factors)
- one after another, minimizing the number of rows and columns
- scales can be changed on both axes



`facet_grid( … )`

- ~ 1 or 2 discrete values (factors)
- each factor is either displayed on rows or on columns
- scales can be changed on same axes than facets

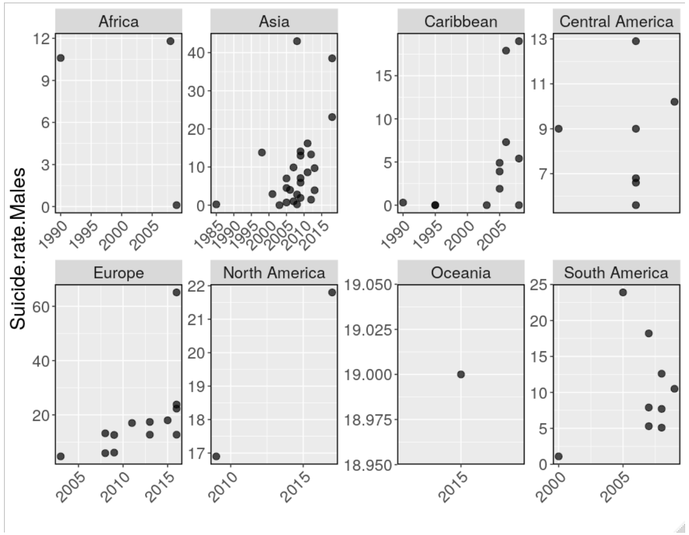


Elements of a plot / graphic

facet, melt

facet_wrap(…)

- ~ 1 or 2 discrete values (factors)
- one after another, minimizing the number of rows and columns
- scales can be changed on both axes



facet_grid(…)

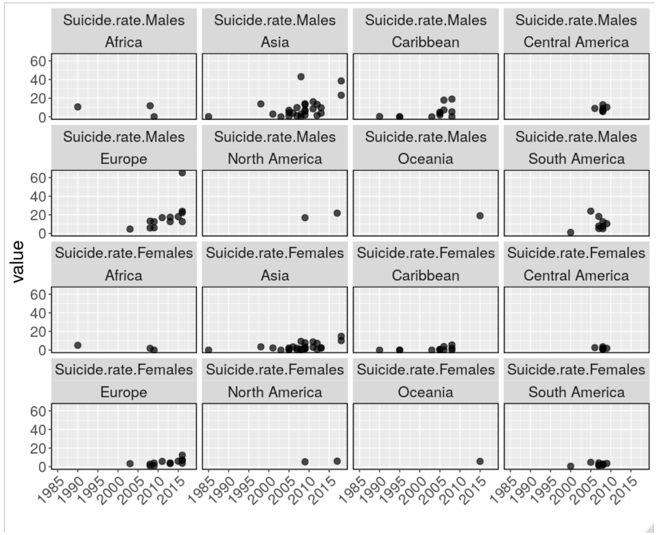
- ~ 1 or 2 discrete values (factors)
- each factor is either displayed on rows or on columns
- scales can be changed on same axes than facets

Elements of a plot / graphic

facet, melt

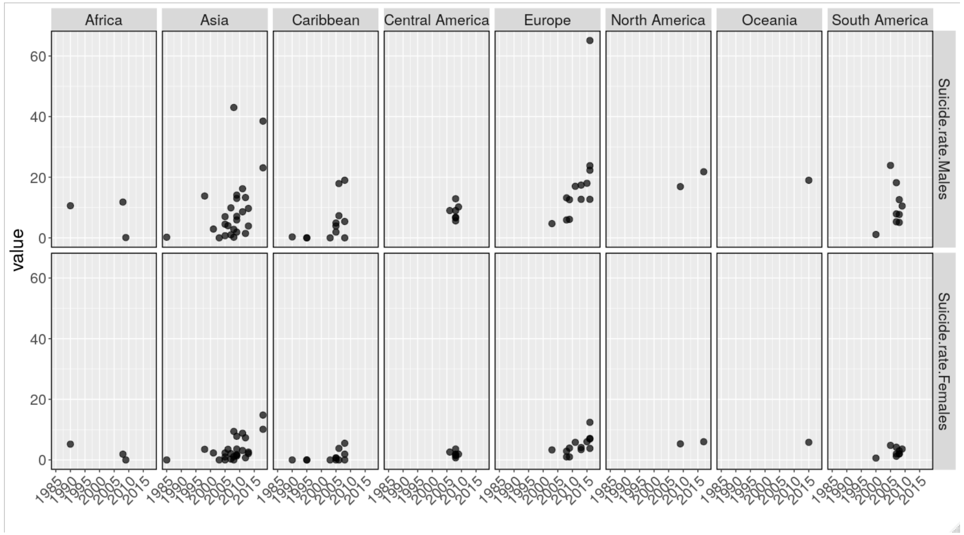
`facet_wrap( … )`

- ~ 1 or 2 discrete values (factors)
- one after another, minimizing the number of rows and columns
- scales can be changed on both axes



`facet_grid( … )`

- ~ 1 or 2 discrete values (factors)
- each factor is either displayed on rows or on columns
- scales can be changed on same axes than facets



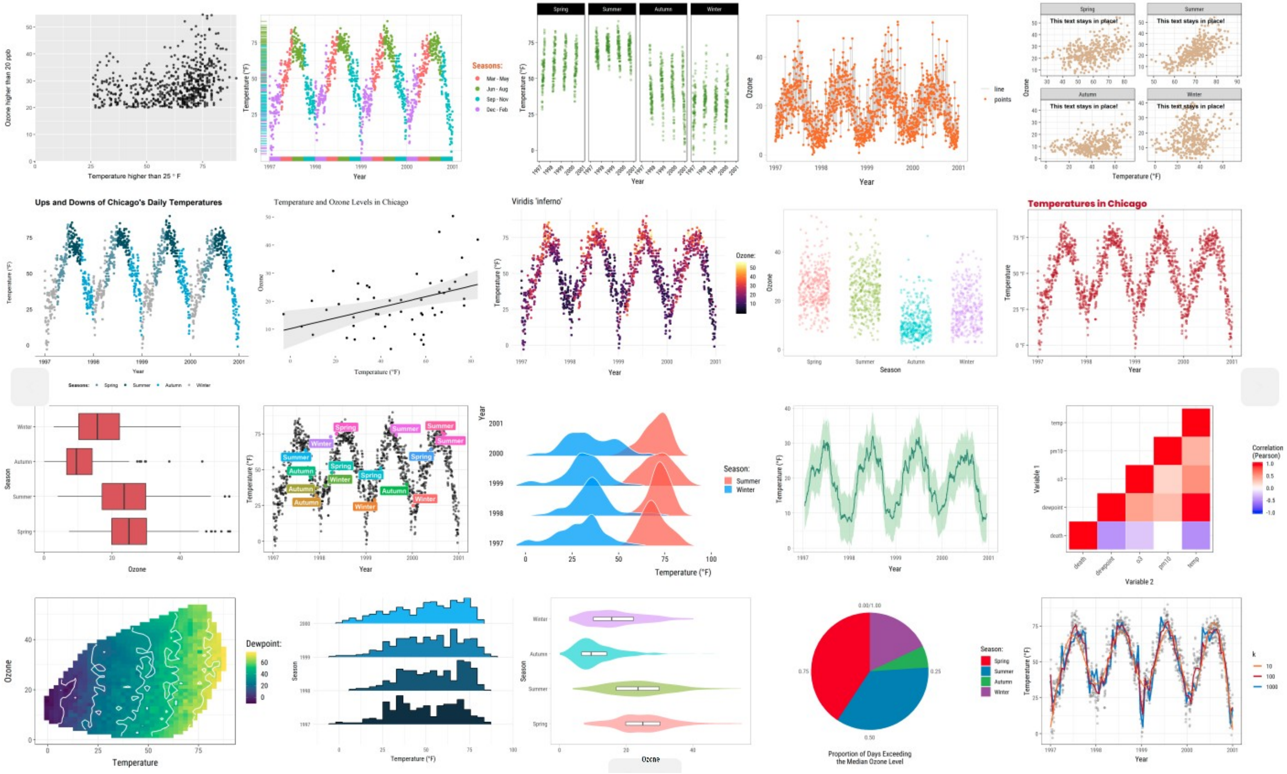
EXERCISE 3

FILE : EX3_TAB_lifeExpectancy.csv

Manipulate your data with melt to optimize your graphic potentialities.
Play with columns, facet, geometries to find what information you can extract from your data.

Elements of a plot / graphic

cosmetics



Elements of a plot / graphic

cosmetics : scales, legends

Each type of variation related to the data (i.e. *aesthetic*) is represented according to a specific scale which is detailed through a legend.

Elements of a plot / graphic

cosmetics : scales, legends

Each type of variation related to the data (i.e. *aesthetic*) is represented according to a specific scale which is detailed through a legend.

One scale for each aesthetic element ...

- x
- y

- size
- color
- fill
- shape
- linetype
- alpha

Elements of a plot / graphic

cosmetics : scales, legends

Each type of variation related to the data (i.e. *aesthetic*) is represented according to a specific scale which is detailed through a legend.

One scale for each aesthetic element ...

- x
- y

- size
- color
- fill
- shape
- linetype
- alpha

... directly adapted to the data type ...

- discrete
- continuous

... or defined by the user...

- manual

Elements of a plot / graphic

cosmetics : scales, legends

Each type of variation related to the data (i.e. *aesthetic*) is represented according to a specific scale which is detailed through a legend.

One scale for each aesthetic element ...

- x
- y

- size
- color
- fill
- shape
- linetype
- alpha

... directly adapted to the data type ...

- discrete
- continuous

... or defined by the user...

- manual

→ scale_*aesthetic*_*datatype*()

Elements of a plot / graphic

cosmetics : scales, legends

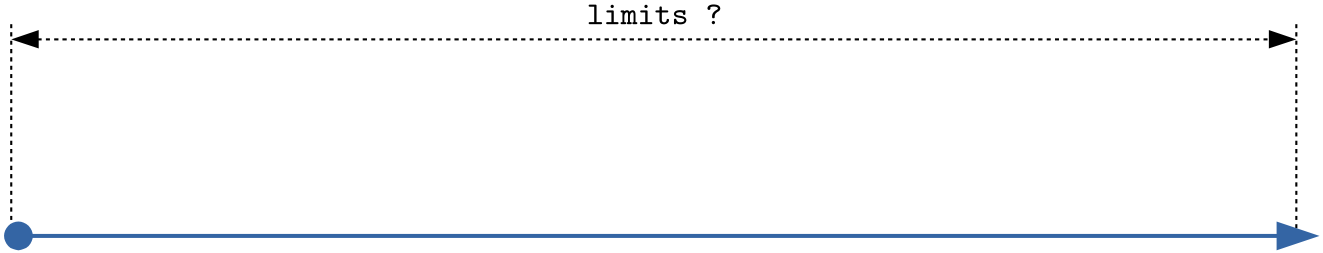
Each type of variation related to the data (i.e. *aesthetic*) is represented according to a specific scale which is detailed through a legend.

```
scale_aesthetic_discrete()
scale_aesthetic_continuous()
```

Elements of a plot / graphic

cosmetics : scales, legends

Each type of variation related to the data (i.e. *aesthetic*) is represented according to a specific scale which is detailed through a legend.

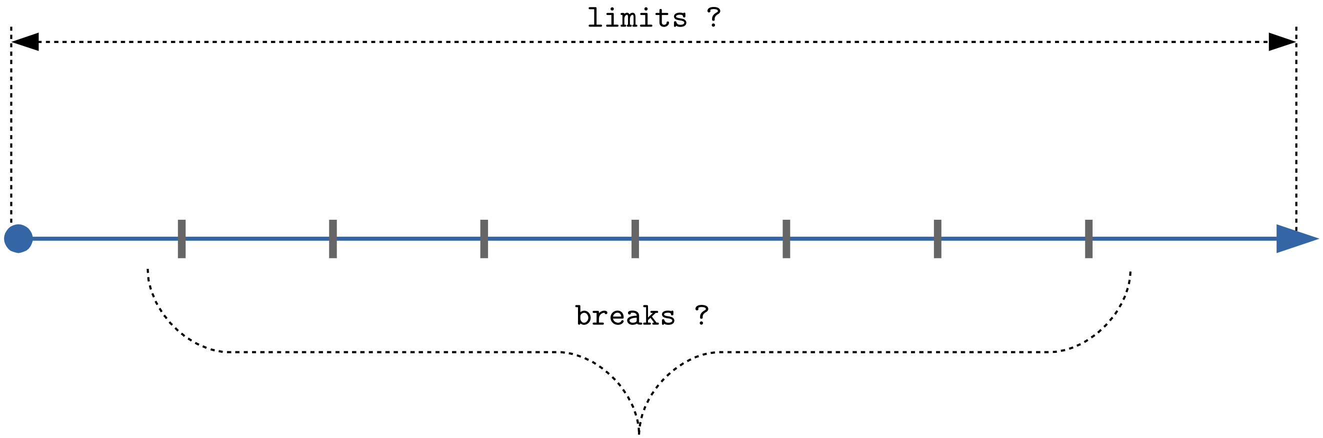


```
scale_aesthetic_discrete()
scale_aesthetic_continuous()
```

Elements of a plot / graphic

cosmetics : scales, legends

Each type of variation related to the data (i.e. *aesthetic*) is represented according to a specific scale which is detailed through a legend.

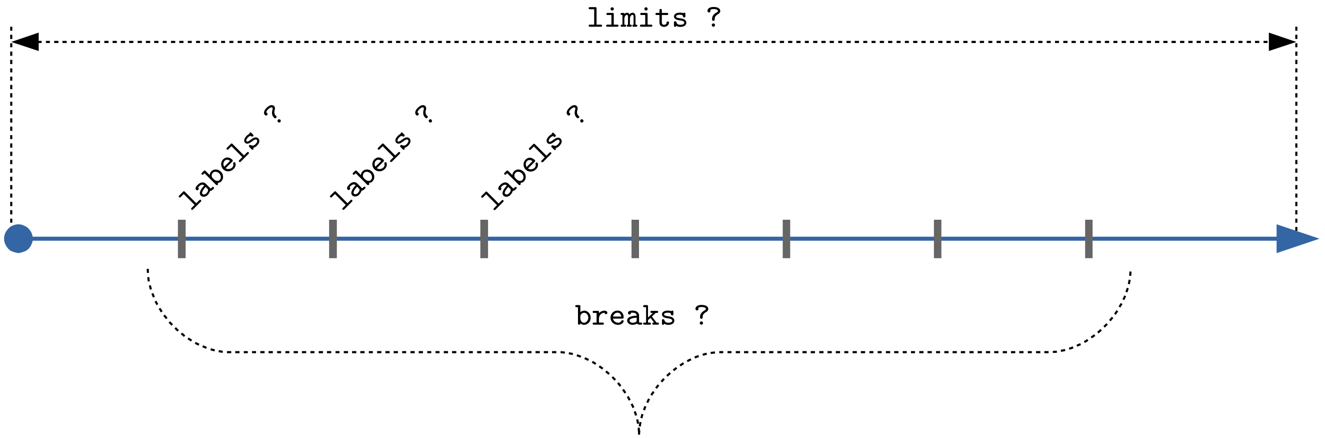


```
scale_aesthetic_discrete()
scale_aesthetic_continuous()
```

Elements of a plot / graphic

cosmetics : scales, legends

Each type of variation related to the data (i.e. *aesthetic*) is represented according to a specific scale which is detailed through a legend.

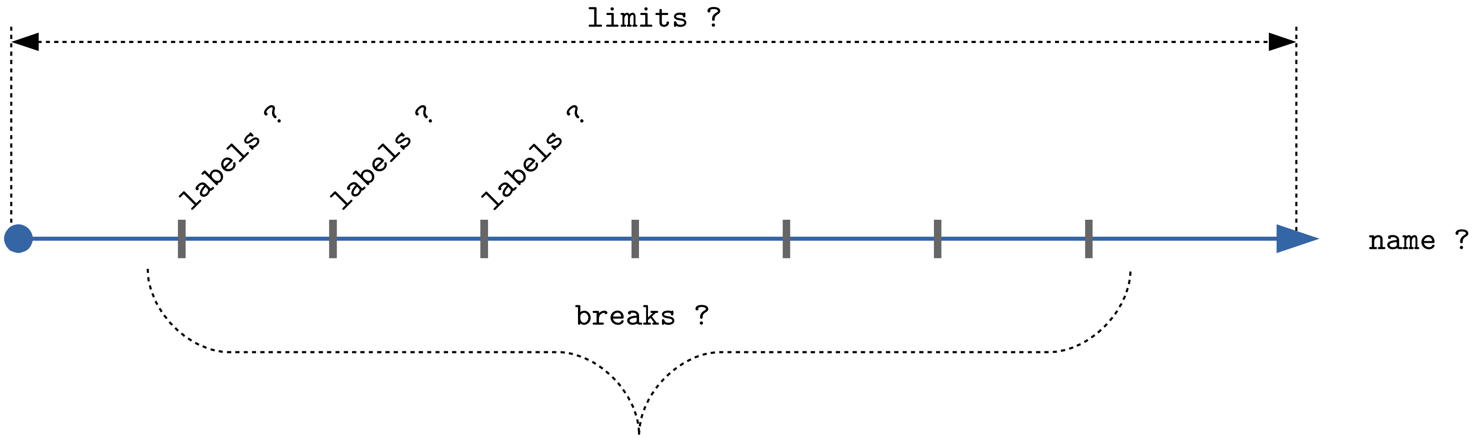


```
scale_aesthetic_discrete()
scale_aesthetic_continuous()
```

Elements of a plot / graphic

cosmetics : scales, legends

Each type of variation related to the data (i.e. *aesthetic*) is represented according to a specific scale which is detailed through a legend.

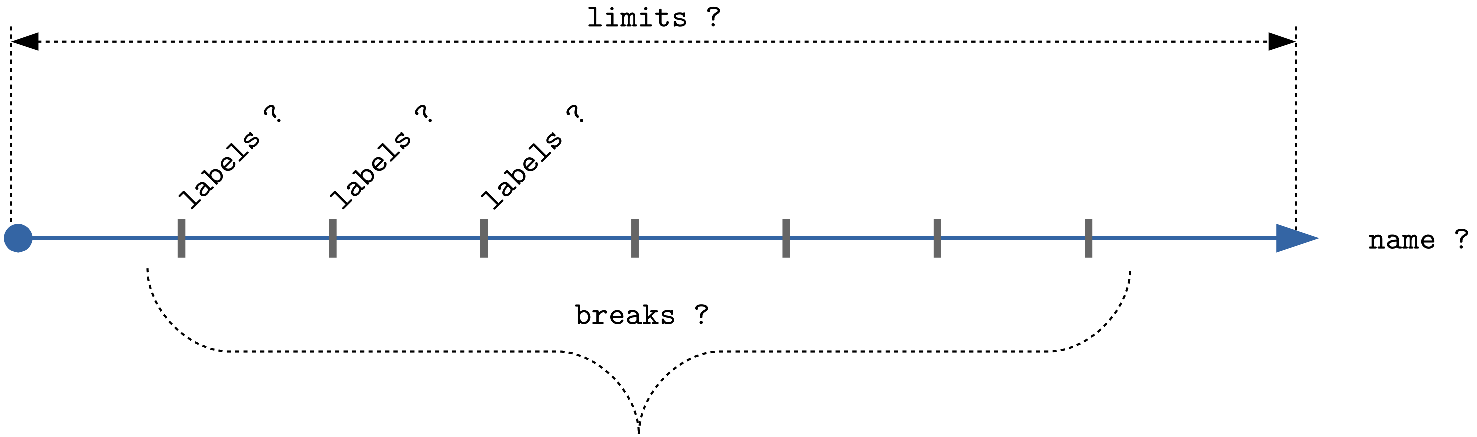


```
scale_aesthetic_discrete()
scale_aesthetic_continuous()
```

Elements of a plot / graphic

cosmetics : scales, legends

Each type of variation related to the data (i.e. *aesthetic*) is represented according to a specific scale which is detailed through a legend.



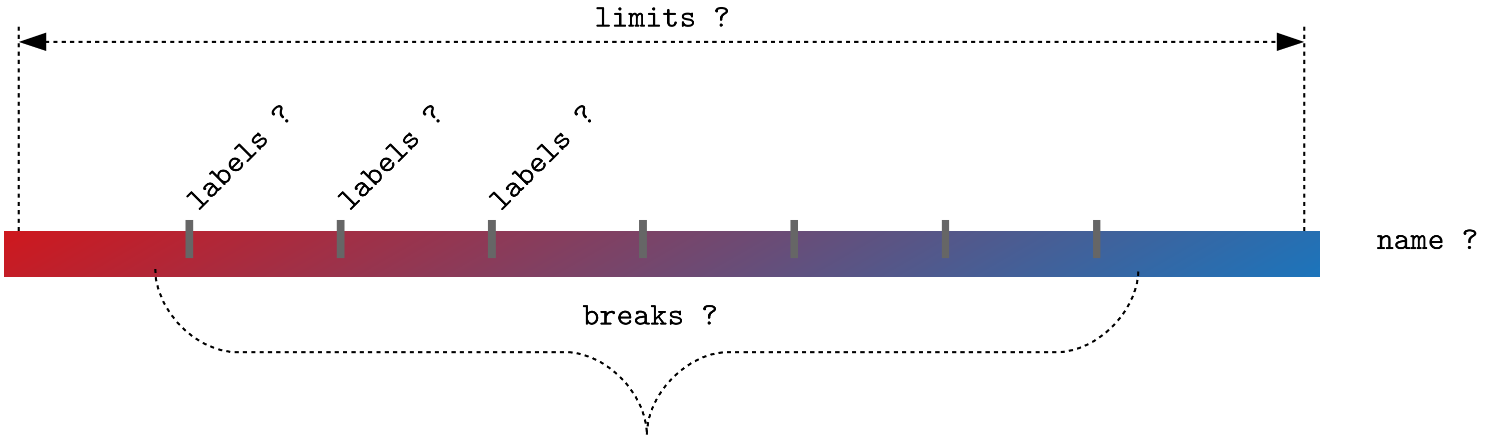
```
scale_aesthetic_discrete()
scale_aesthetic_continuous()
```

x, y

Elements of a plot / graphic

cosmetics : scales, legends

Each type of variation related to the data (i.e. *aesthetic*) is represented according to a specific scale which is detailed through a legend.



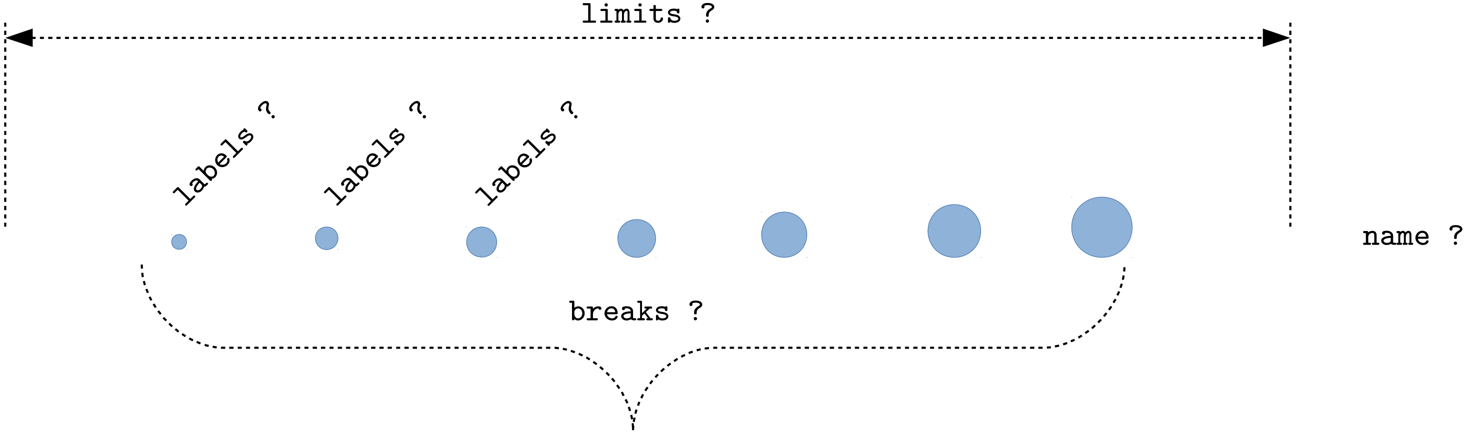
```
scale_aesthetic_discrete()
scale_aesthetic_continuous()
```

```
     x, y
color, fill
```

Elements of a plot / graphic

cosmetics : scales, legends

Each type of variation related to the data (i.e. *aesthetic*) is represented according to a specific scale which is detailed through a legend.



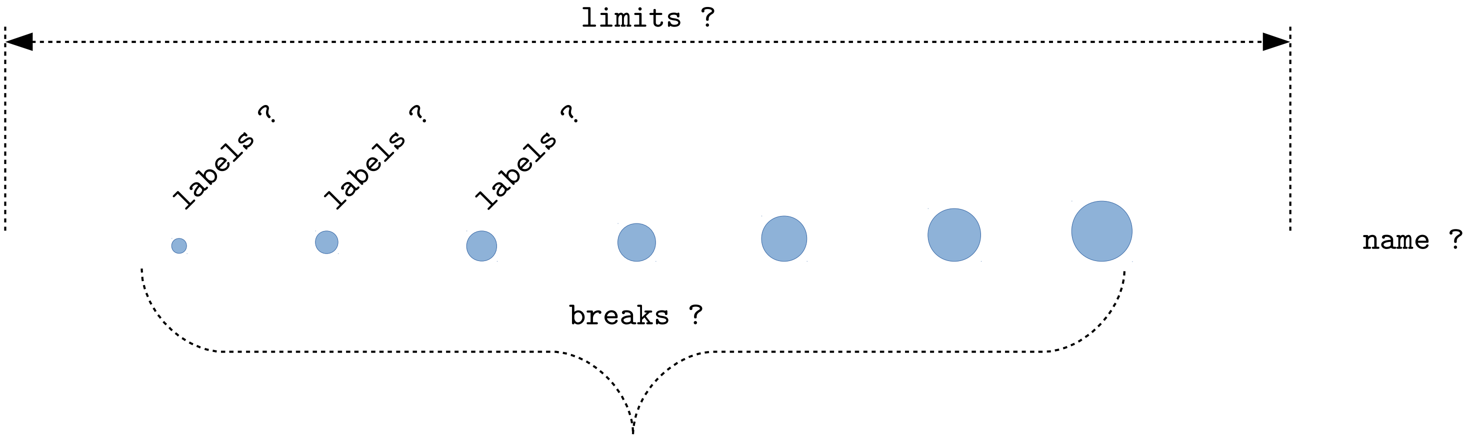
```
scale_aesthetic_discrete()
scale_aesthetic_continuous()
```

x, y   size
color, fill

Elements of a plot / graphic

cosmetics : scales, legends

Each type of variation related to the data (i.e. *aesthetic*) is represented according to a specific scale which is detailed through a legend.



```
scale_aesthetic_discrete()
scale_aesthetic_continuous()
```

```
      x, y                    size
color, fill    (alpha, shape, linetype)
```

Elements of a plot / graphic

cosmetics : scales, legends

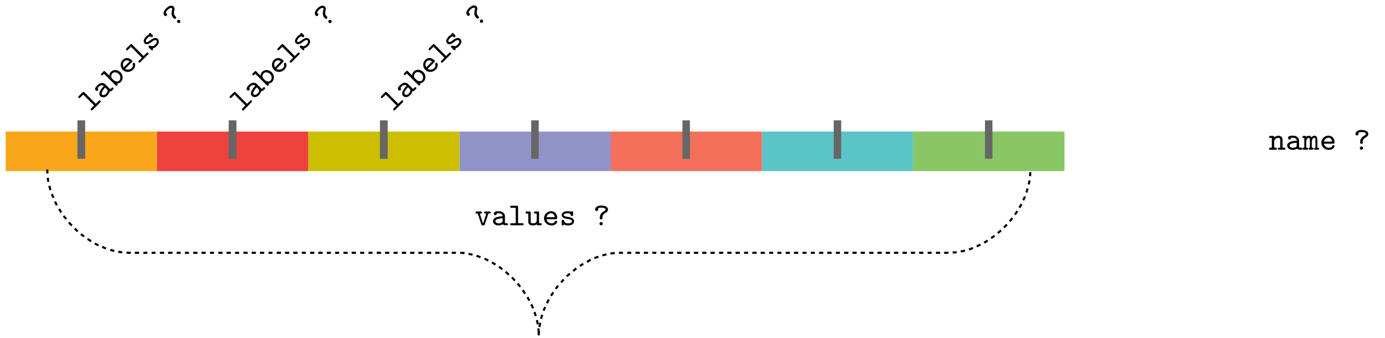
Each type of variation related to the data (i.e. *aesthetic*) is represented according to a specific scale which is detailed through a legend.

```
scale_aesthetic_manual() ?
```

Elements of a plot / graphic

cosmetics : scales, legends

Each type of variation related to the data (i.e. *aesthetic*) is represented according to a specific scale which is detailed through a legend.



`scale_`*`aesthetic`*`_manual() ?`

`color, fill` *to fix specific colors to discrete/categorical values*

Elements of a plot / graphic

cosmetics : theme

Each element of the graph that is not directly linked to the data can be set / modified within the theme layer.

Elements of a plot / graphic

cosmetics : theme

Each element of the graph that is not directly linked to the data can be set / modified within the theme layer.

All the elements related to ...

- axis
- legend
- panel
- plot
- strip

Elements of a plot / graphic

cosmetics : theme

Each element of the graph that is not directly linked to the data can be set / modified within the theme layer.

All the elements related to …

- axis
- legend
- panel
- plot
- strip

… and that correspond to a specific type of object …

- unit
- margin

- element_text
- element_line
- element_rect
- element_blank

Elements of a plot / graphic

cosmetics : theme

Each element of the graph that is not directly linked to the data can be set / modified within the theme layer.

All the elements related to ...

- axis
- legend
- panel
- plot
- strip

... and that correspond to a specific type of object ...

- unit
- margin

- element_text
- element_line
- element_rect
- element_blank

→

```
theme(axis.text.x = element_text(angle = 45, hjust = 1)
        , axis.ticks.length = unit(0,1, ‘cm’)
        , legend.position = ‘bottom’
        , plot.background = element_rect(fill = NA)
        , title = element_blank())
```

Elements of a plot / graphic

cosmetics : ggthemes

Package containing pre-defined themes !

(but you can still modify them with the theme function)

(and there is also basic themes defined in the ggplot2 package)

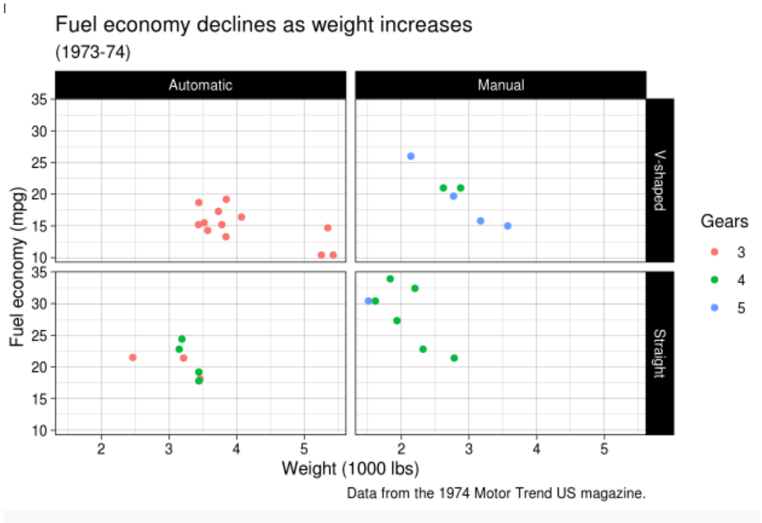
Elements of a plot / graphic

cosmetics : ggthemes

Package containing pre-defined themes !

(but you can still modify them with the theme function)

(and there is also basic themes defined in the ggplot2 package)



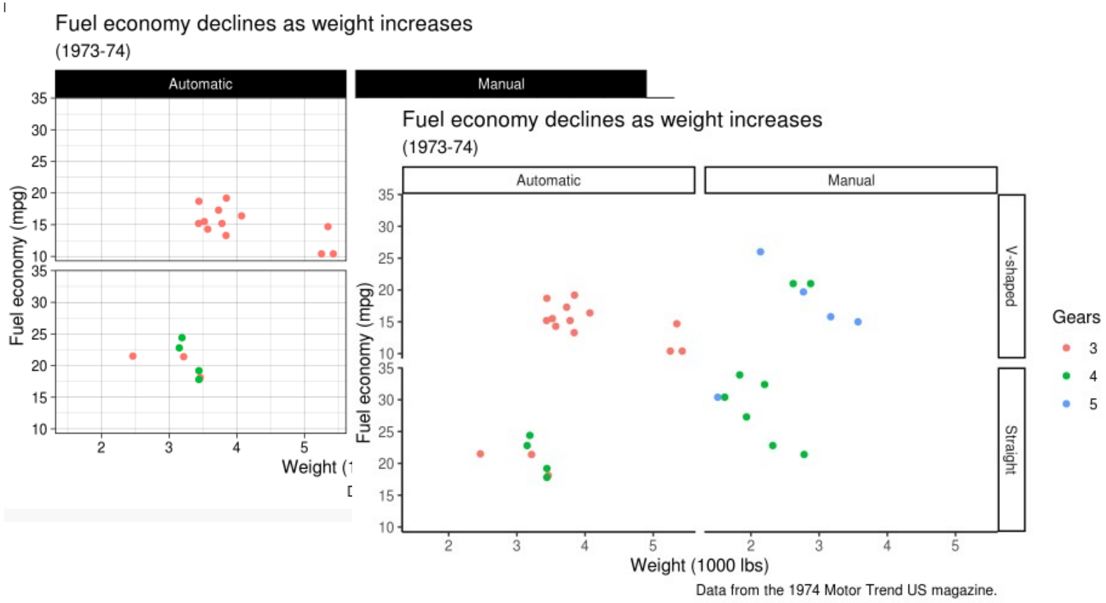
Elements of a plot / graphic

cosmetics : ggthemes

Package containing pre-defined themes !

(but you can still modify them with the theme function)

(and there is also basic themes defined in the ggplot2 package)



Fuel economy declines as weight increases

(1973-74)

Fuel economy declines as weight increases

(1973-74)

Data from the 1974 Motor Trend US magazine.

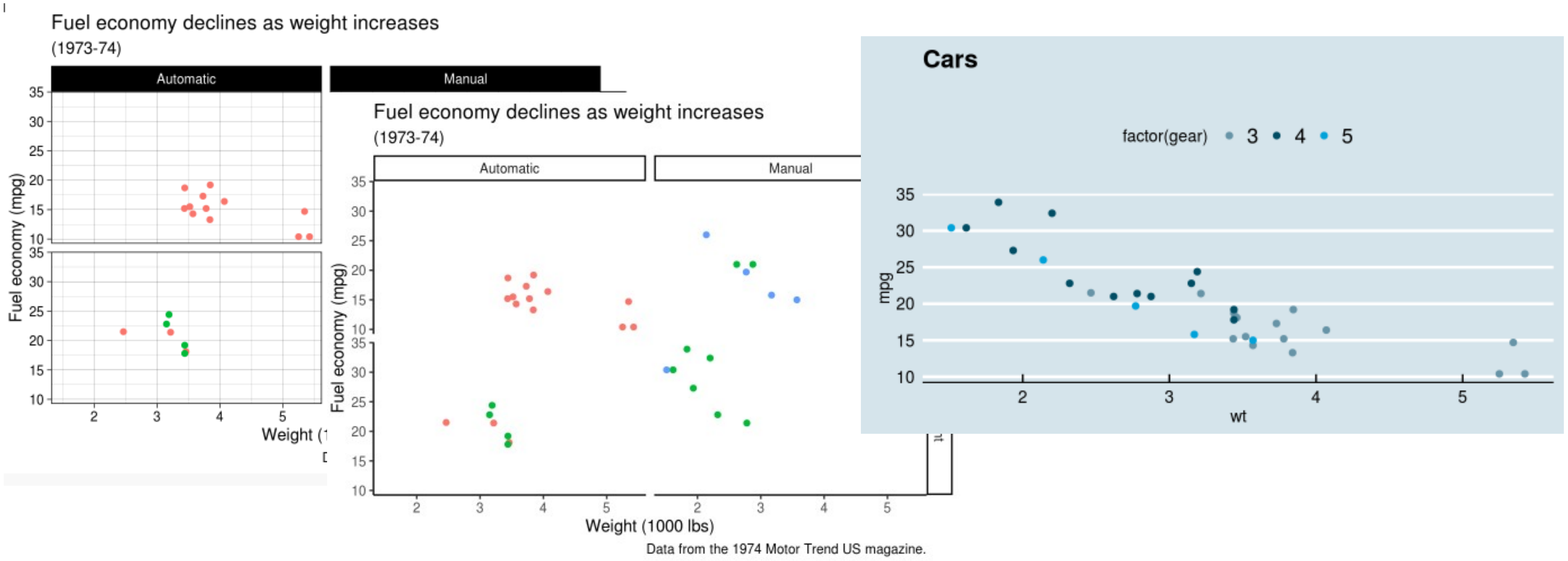
Elements of a plot / graphic

cosmetics : ggthemes

Package containing pre-defined themes !

(but you can still modify them with the theme function)

(and there is also basic themes defined in the ggplot2 package)



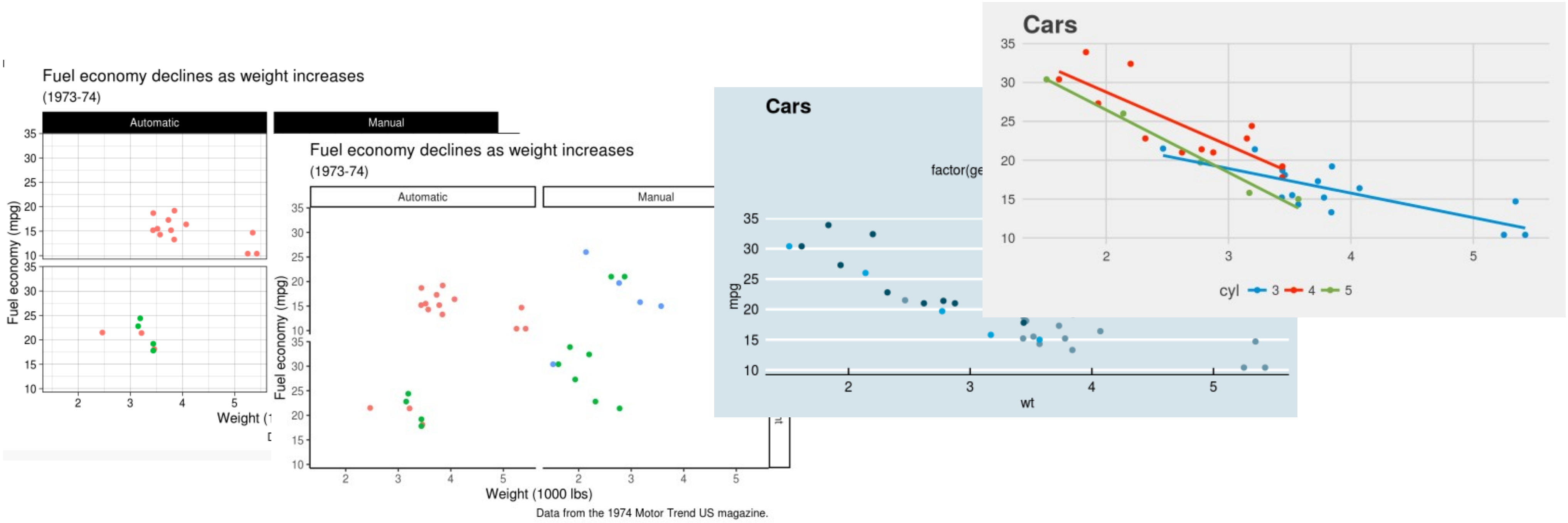
Elements of a plot / graphic

cosmetics : ggthemes

Package containing pre-defined themes !

(but you can still modify them with the theme function)

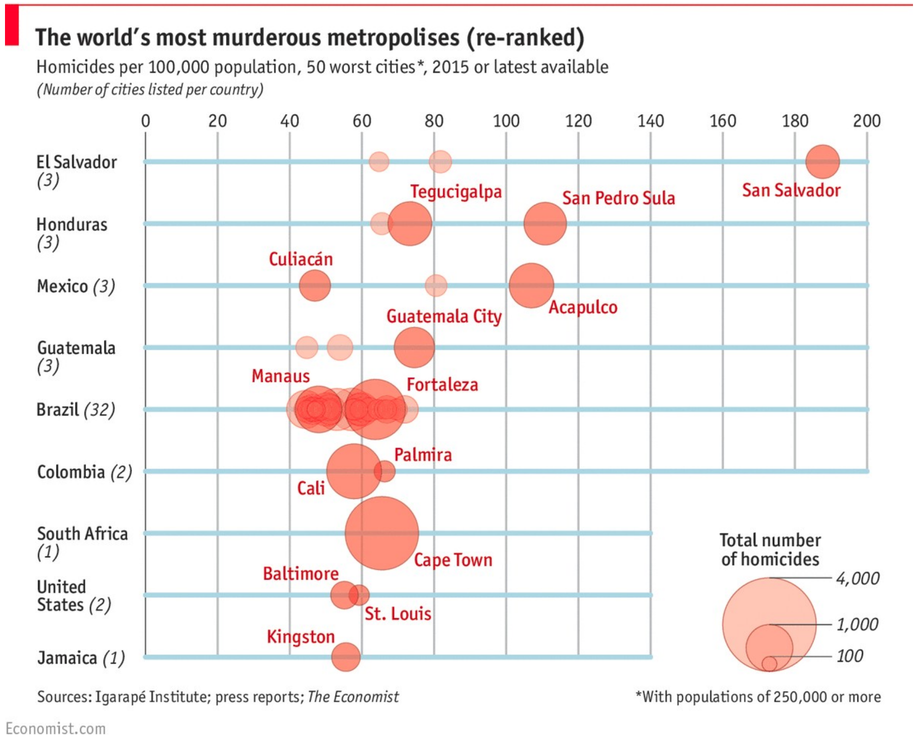
(and there is also basic themes defined in the ggplot2 package)



EXERCISE 4

<u>FILE :</u> EX1_TAB_homicide.csv

Perfect your graph to match with the example !



The world's most murderous metropolises (re-ranked)

Homicides per 100,000 population, 50 worst cities*, 2015 or latest available

(Number of cities listed per country)

Sources: Igarapé Institute; press reports; *The Economist*

*With populations of 250,000 or more

Economist.com

Elements of a plot / graphic

extensions

https://github.com/rstudio/cheatsheets/blob/master/data-visualization-2.1.pdf

https://www.data-to-viz.com/

http://colorbrewer2.org/
https://personal.sron.nl/~pault/

https://plot.ly/ggplot2/

http://www.ggplot2-exts.org/gallery/
https://www.ggplot2-exts.org/ggiraph.html