STATISTICAL HORIZONS

Data Visualization

Kieran Healy, Ph.D.

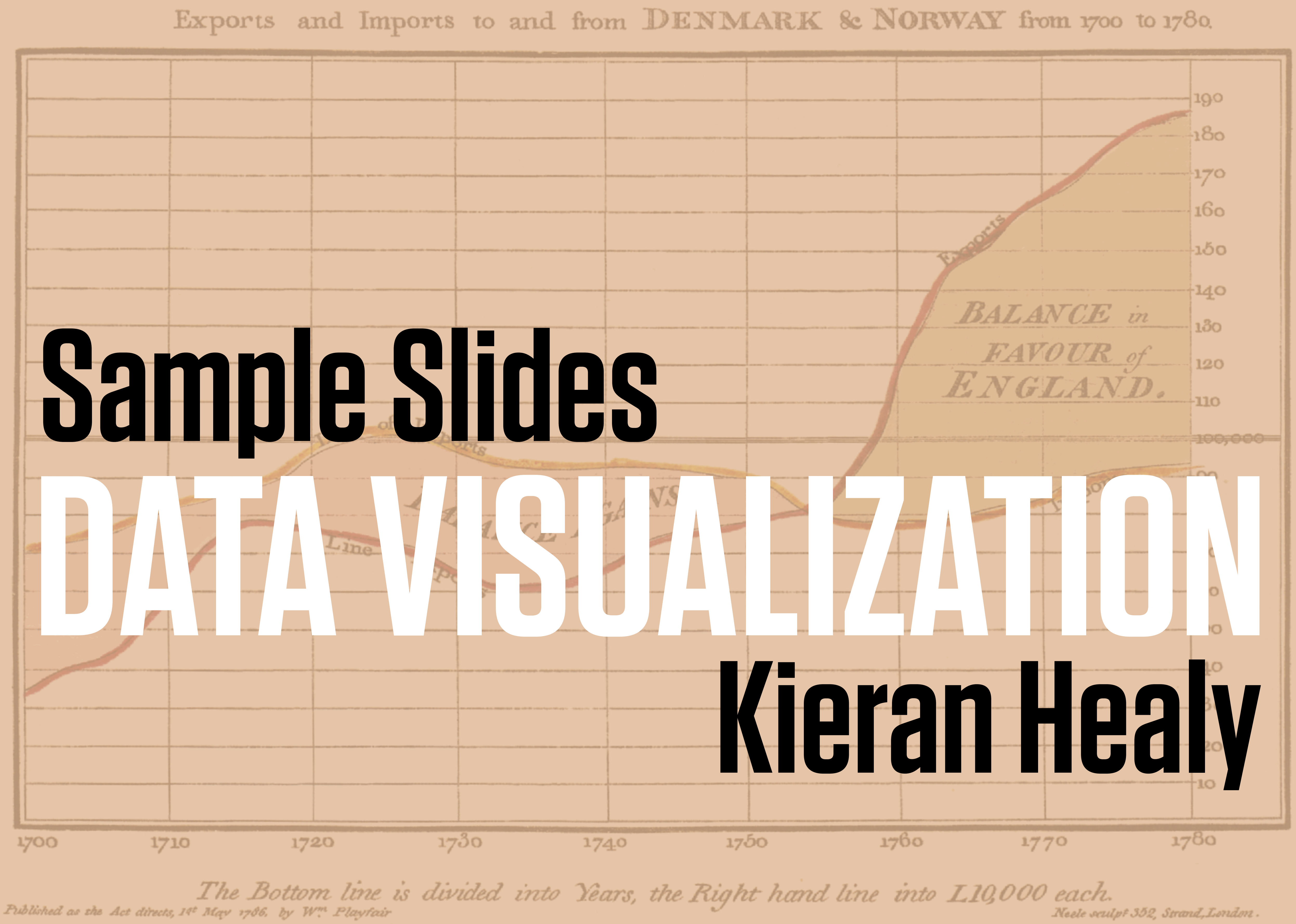
Upcoming Seminar:

December 1-2, 2017, Philadelphia, Pennsylvania

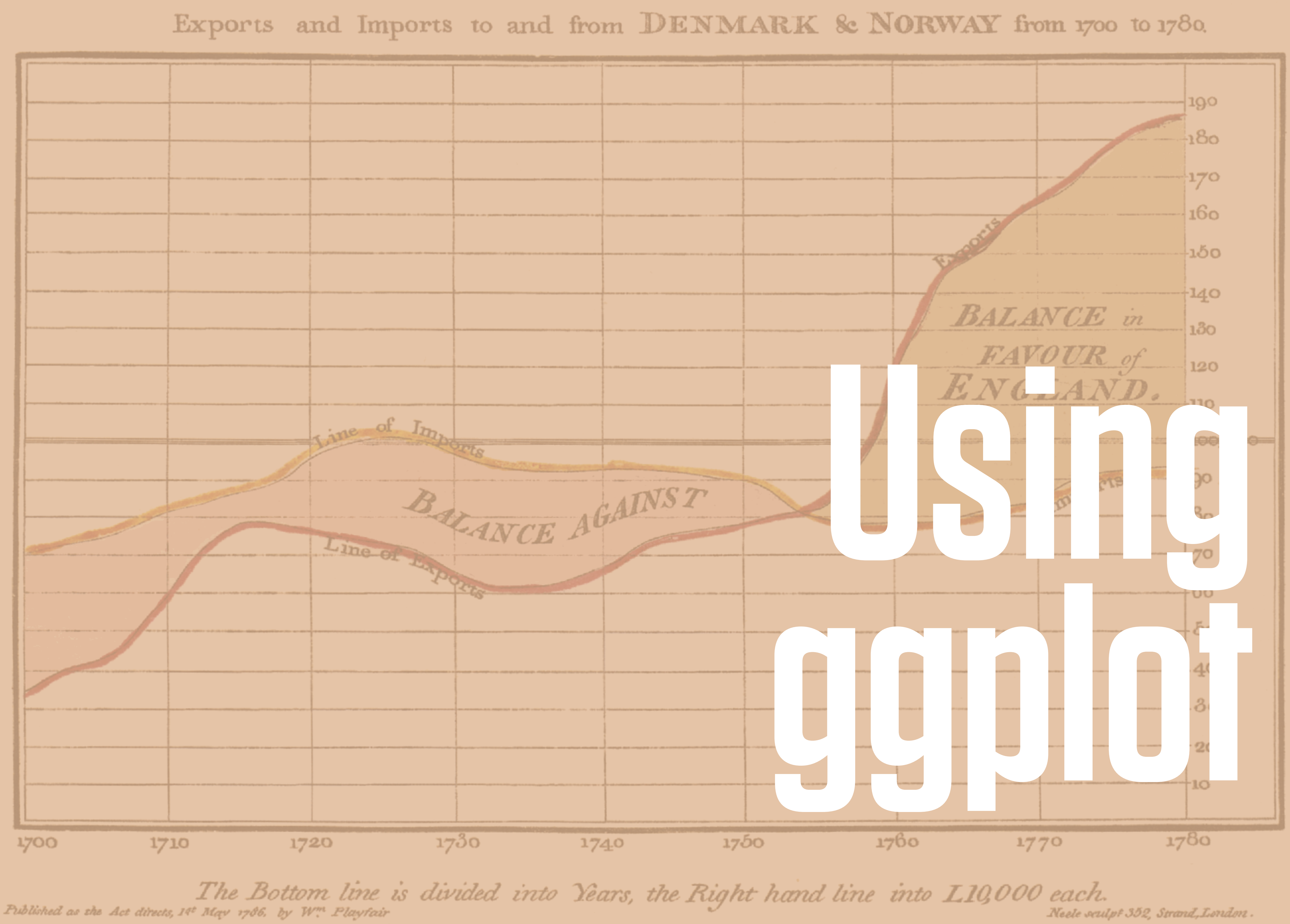
Sample Slides

DATA VISUALIZATION

Kieran Healy



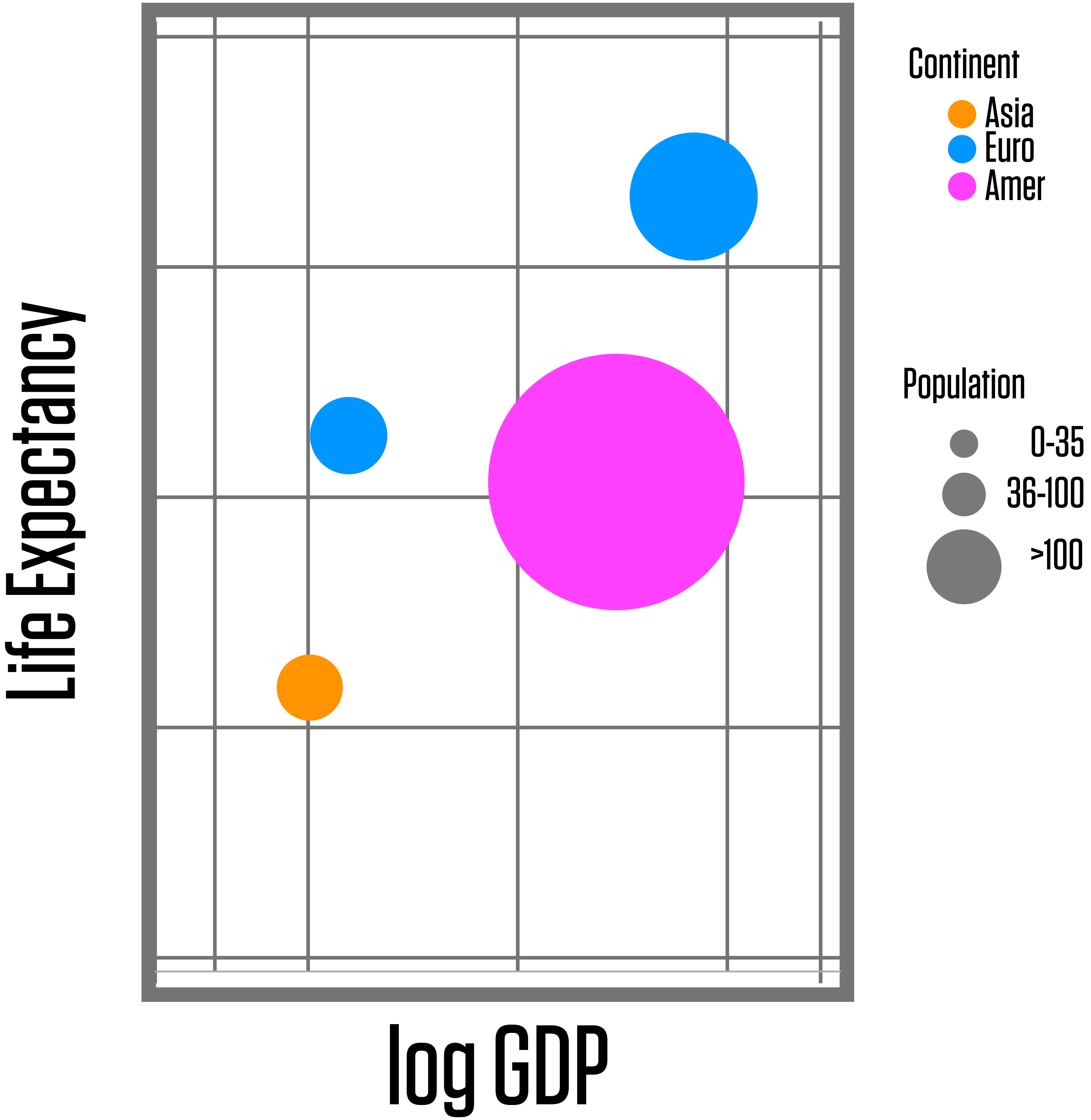
Using ggplot



HOW IT WORKS

gdp	lifexp	pop	continent
340	65	31	Euro
227	51	200	Amer
909	81	80	Euro
126	40	20	Asia

A Gapminder Plot



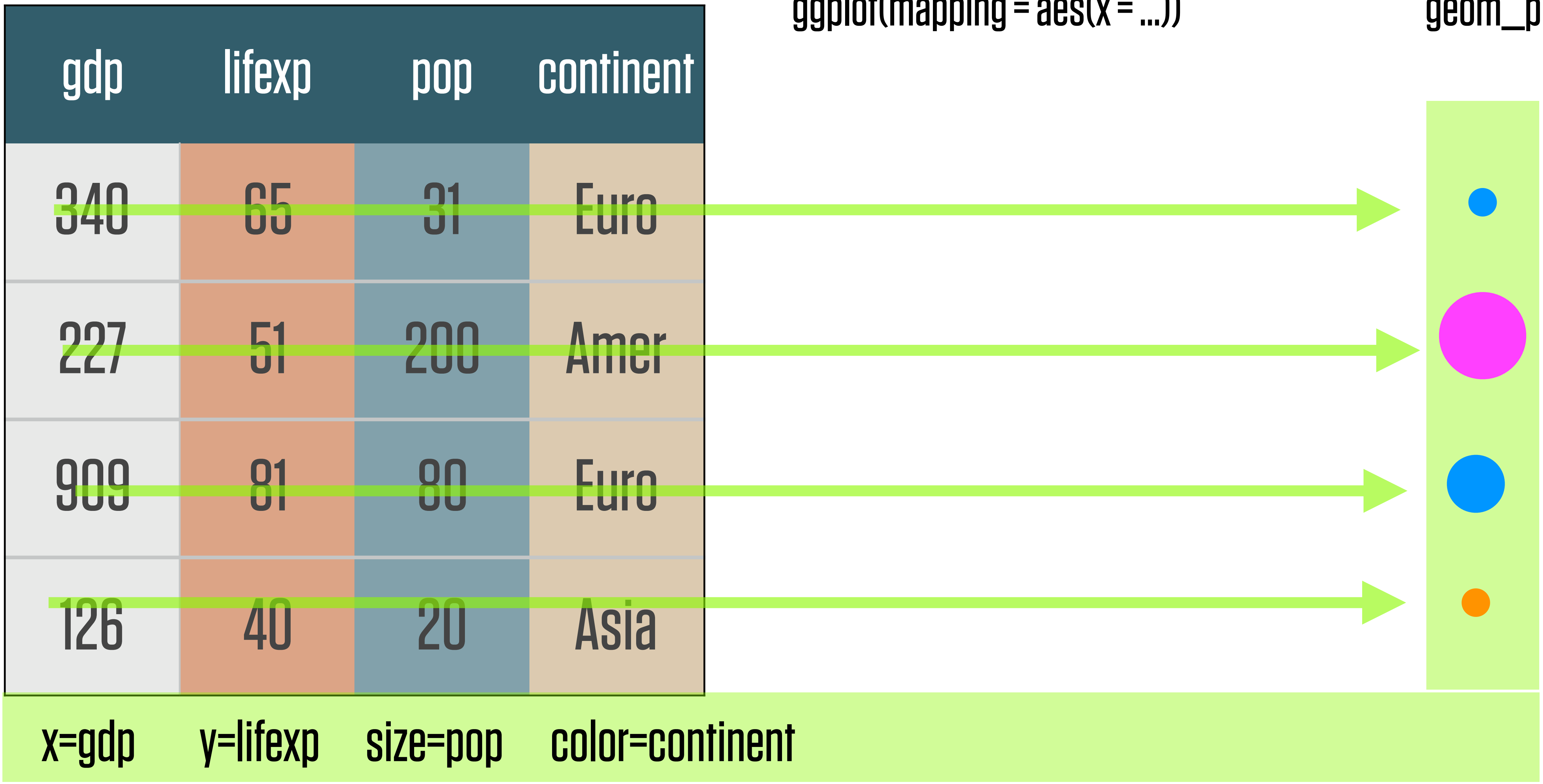
1. Tidy Data

2. Mapping

ggplot(mapping = aes(x = ...))

3. Geom

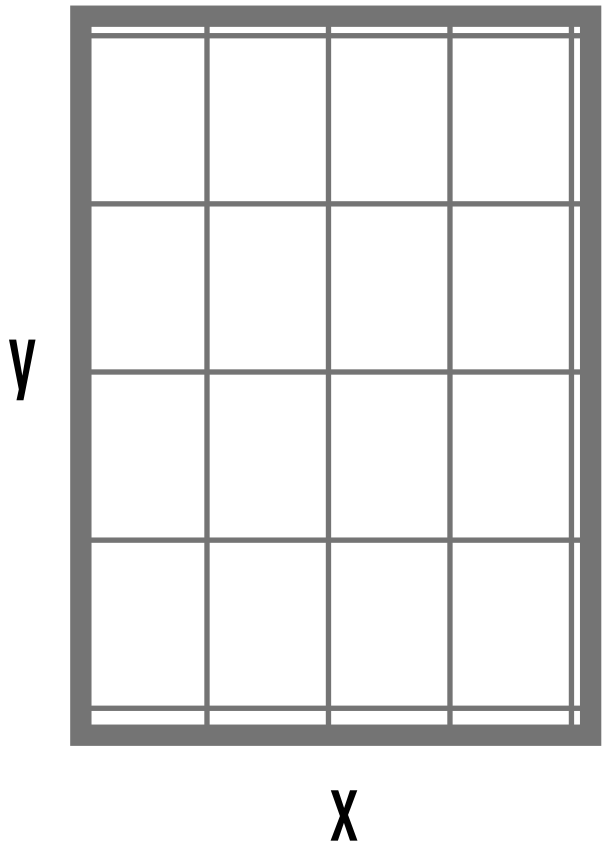
geom_point()



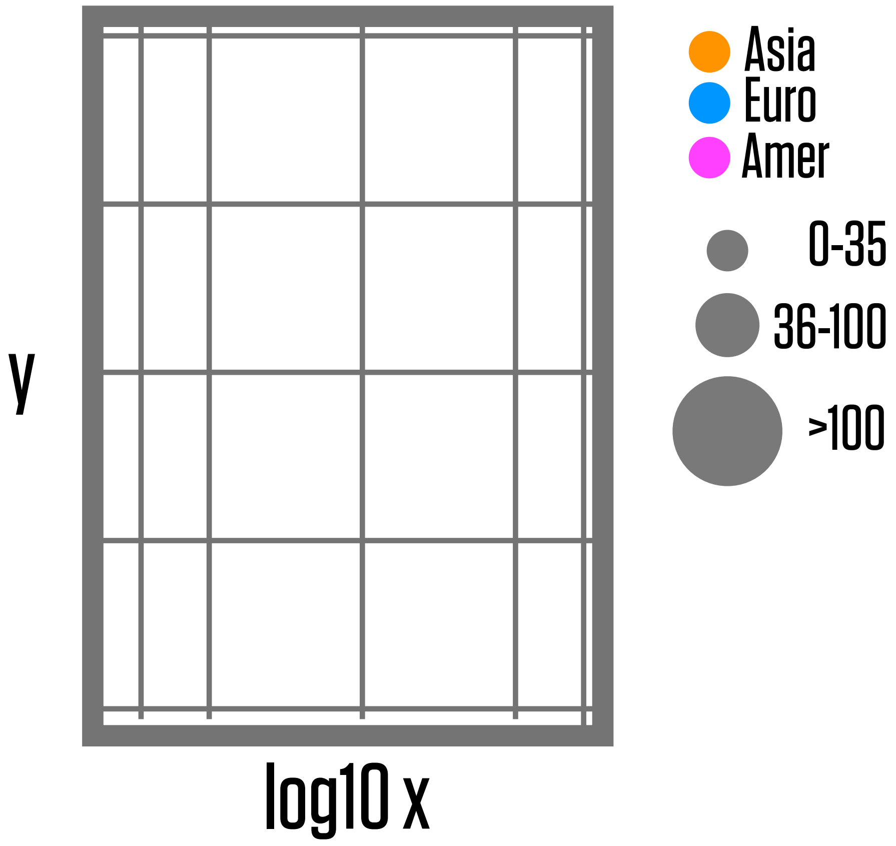
gdp	lifexp	pop	continent
340	65	31	Euro
227	51	200	Amer
909	81	80	Euro
126	40	20	Asia

x=gdp y=lifexp size=pop color=continent

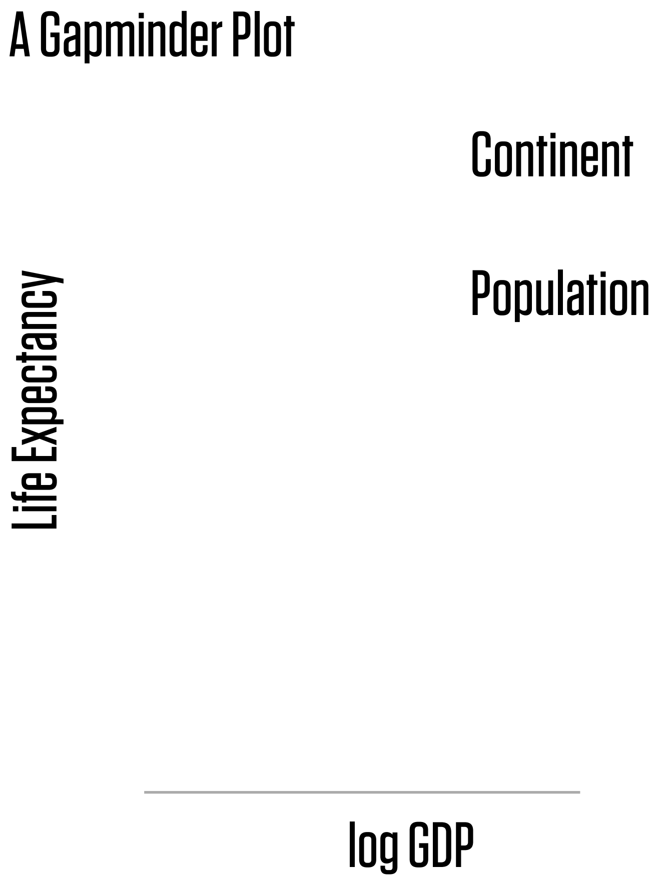
4. Coordinate System



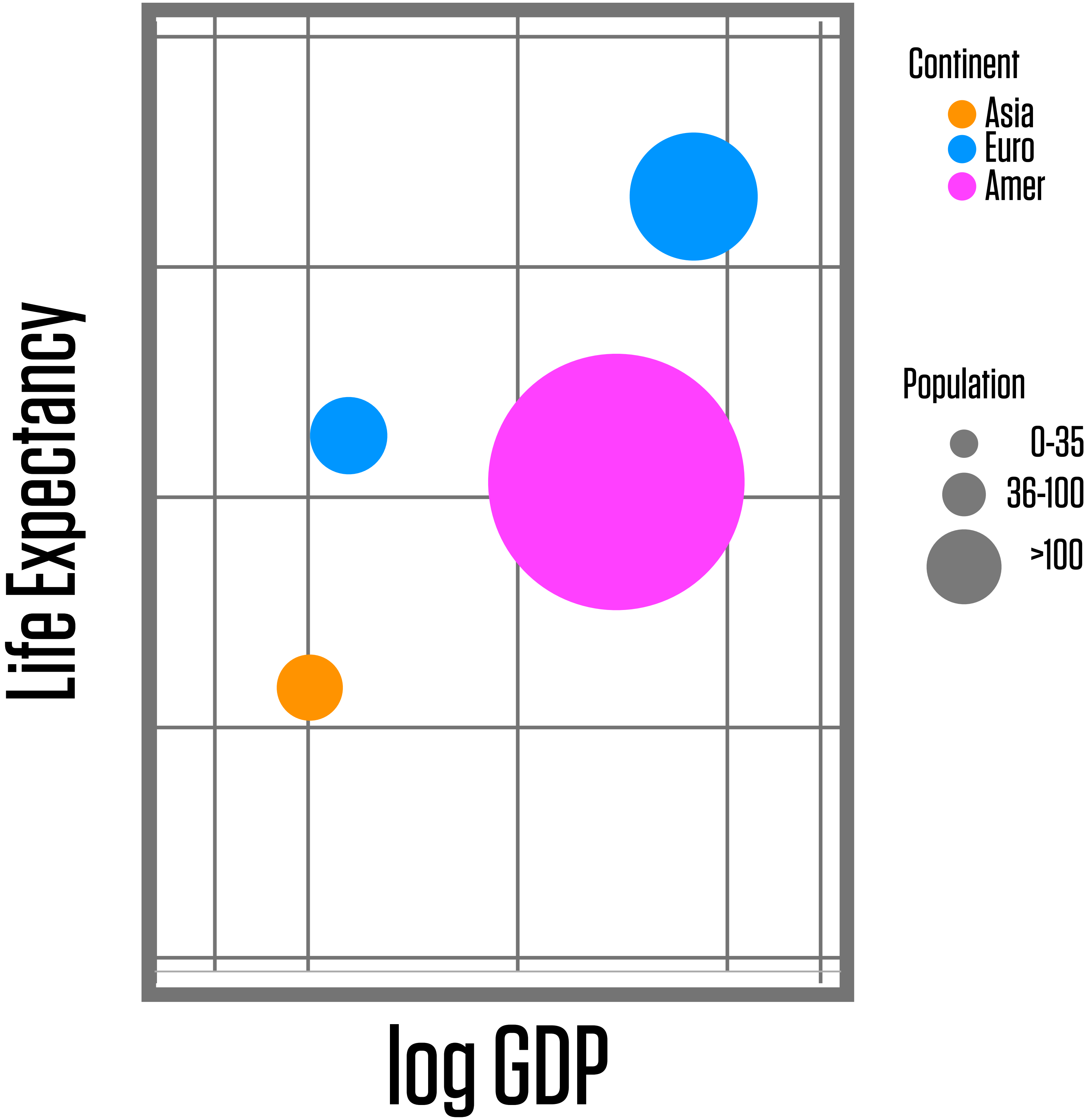
5. Scales



6. Labels & Guides



A Gapminder Plot



LAYER BY LAYER

```
head(gapminder)
```

```
## # A tibble: 6 × 6
##       country continent  year lifeExp      pop gdpPercap
##        <fctr>    <fctr> <int>   <dbl>    <int>     <dbl>
## 1 Afghanistan      Asia  1952  28.801  8425333  779.4453
## 2 Afghanistan      Asia  1957  30.332  9240934  820.8530
## 3 Afghanistan      Asia  1962  31.997 10267083  853.1007
## 4 Afghanistan      Asia  1967  34.020 11537966  836.1971
## 5 Afghanistan      Asia  1972  36.088 13079460  739.9811
## 6 Afghanistan      Asia  1977  38.438 14880372  786.1134
```

```
dim(gapminder)
## [1] 1704    6
```

```
p <- ggplot(data = gapminder)
```

Create a ggplot object
Data is gapminder table

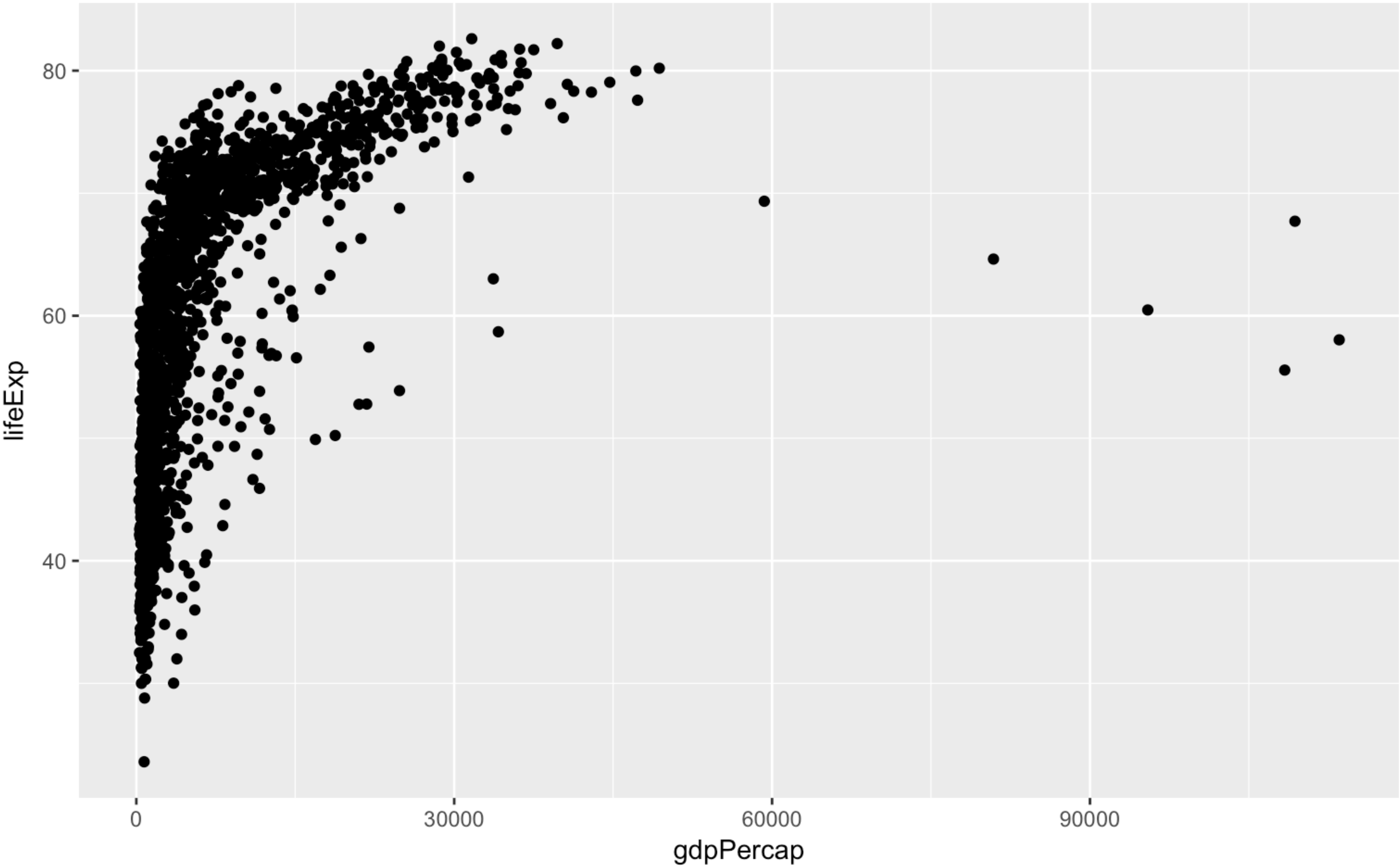
```r
p <- ggplot(data = gapminder,
            mapping = aes(x = gdpPercap,
                          y = lifeExp))
```

Mapping: tell ggplot the relationships you want to see

- The `mapping = aes(...)` instruction *links variables* to *things you will see* on the plot.
- The `x` and `y` values are the most obvious ones.
- Other aesthetic mappings can include, e.g., `color`, `shape`, and `size`.
- These mappings are not *directly* specifying what specific, e.g., colors or shapes will be on the plot. Rather they say which *variables* in the data will be *represented* by, e.g., colors and shapes on the plot.

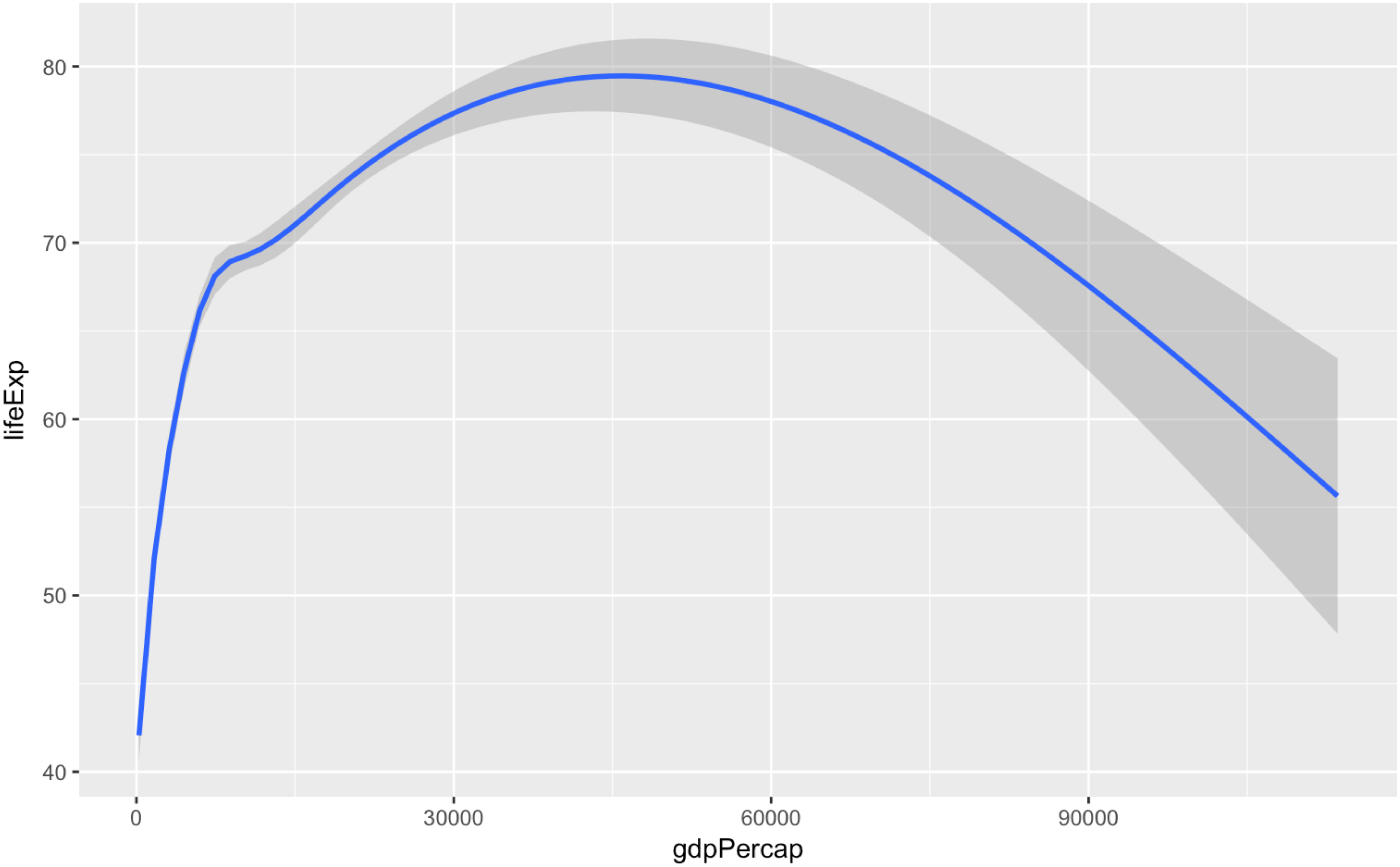
```
p + geom_point()
```

Add a geom layer to the plot


80
60
40
lifeExp
0
30000
60000
90000
gdpPercap

```r
p + geom_smooth()
```

Try a different geom

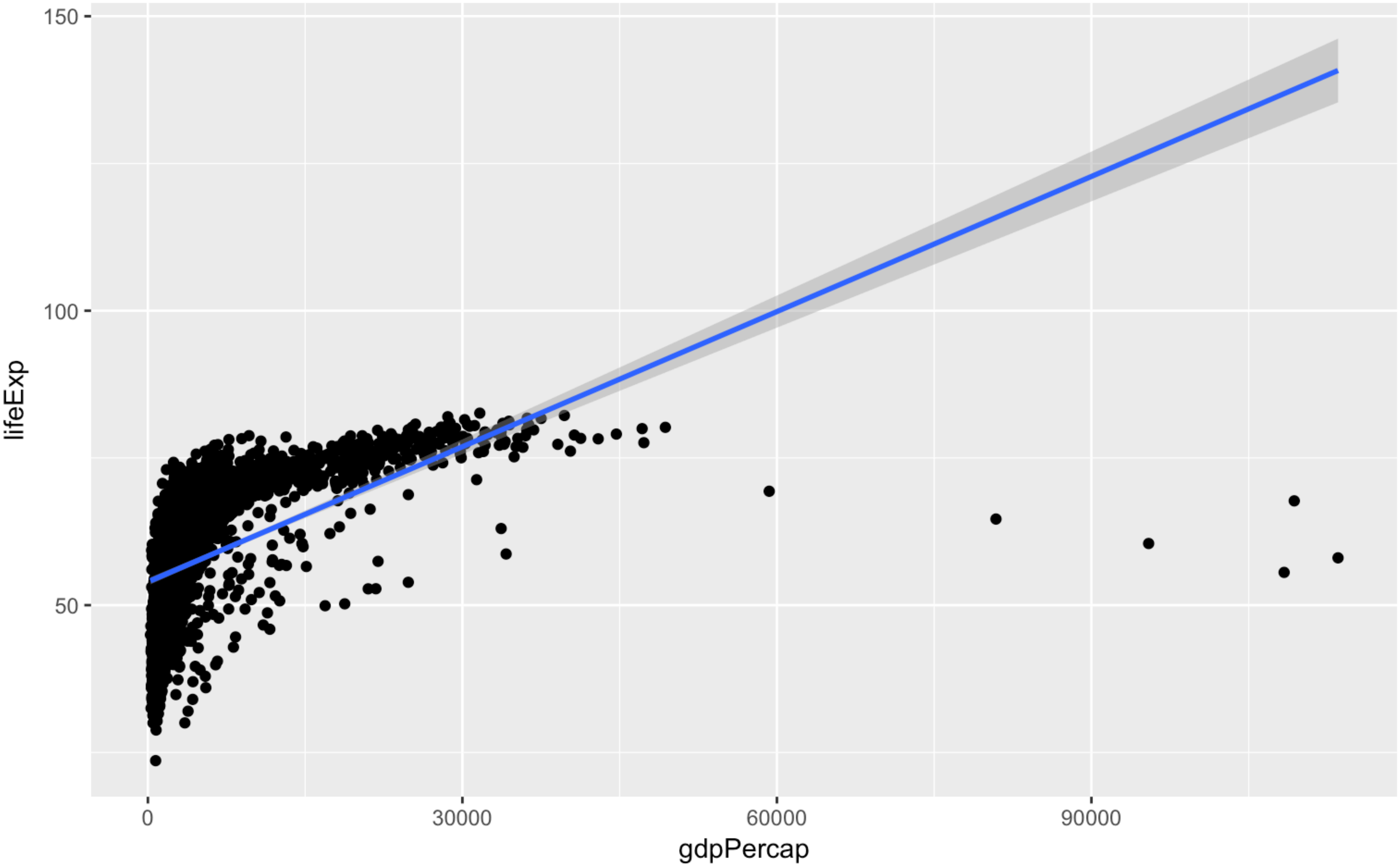

80
70
60
50
40
lifeExp
0
30000
60000
90000
gdpPercap

```r
p + geom_point() +
    geom_smooth(method='gam') +
    scale_x_log10(labels = scales::dollar)
```

This process is literally additive

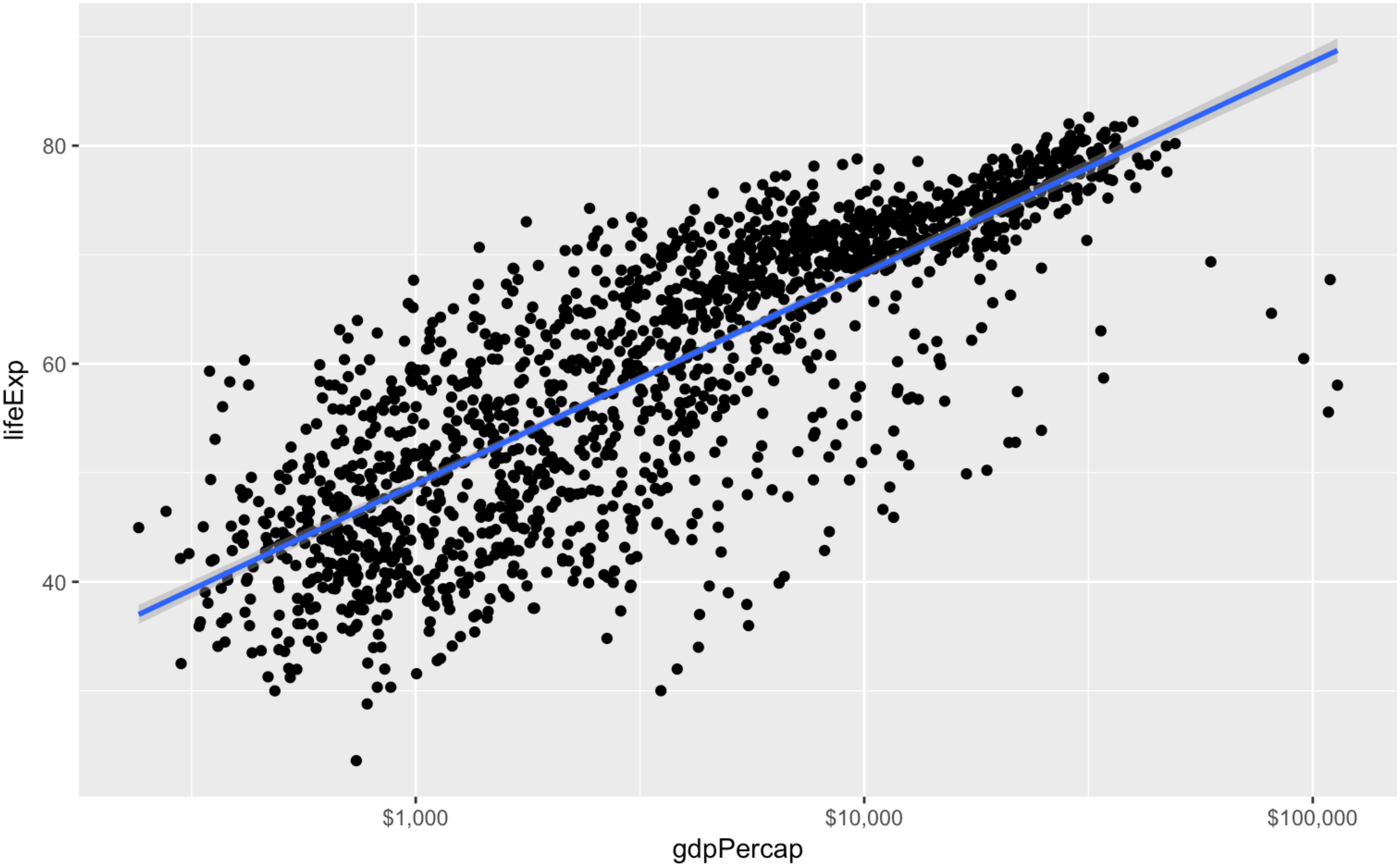
```r
p + geom_point() +
geom_smooth(method = “lm”)
```

Every geom is a function. It can take arguments.


150
100
50
lifeExp
0
30000
60000
90000
gdpPercap

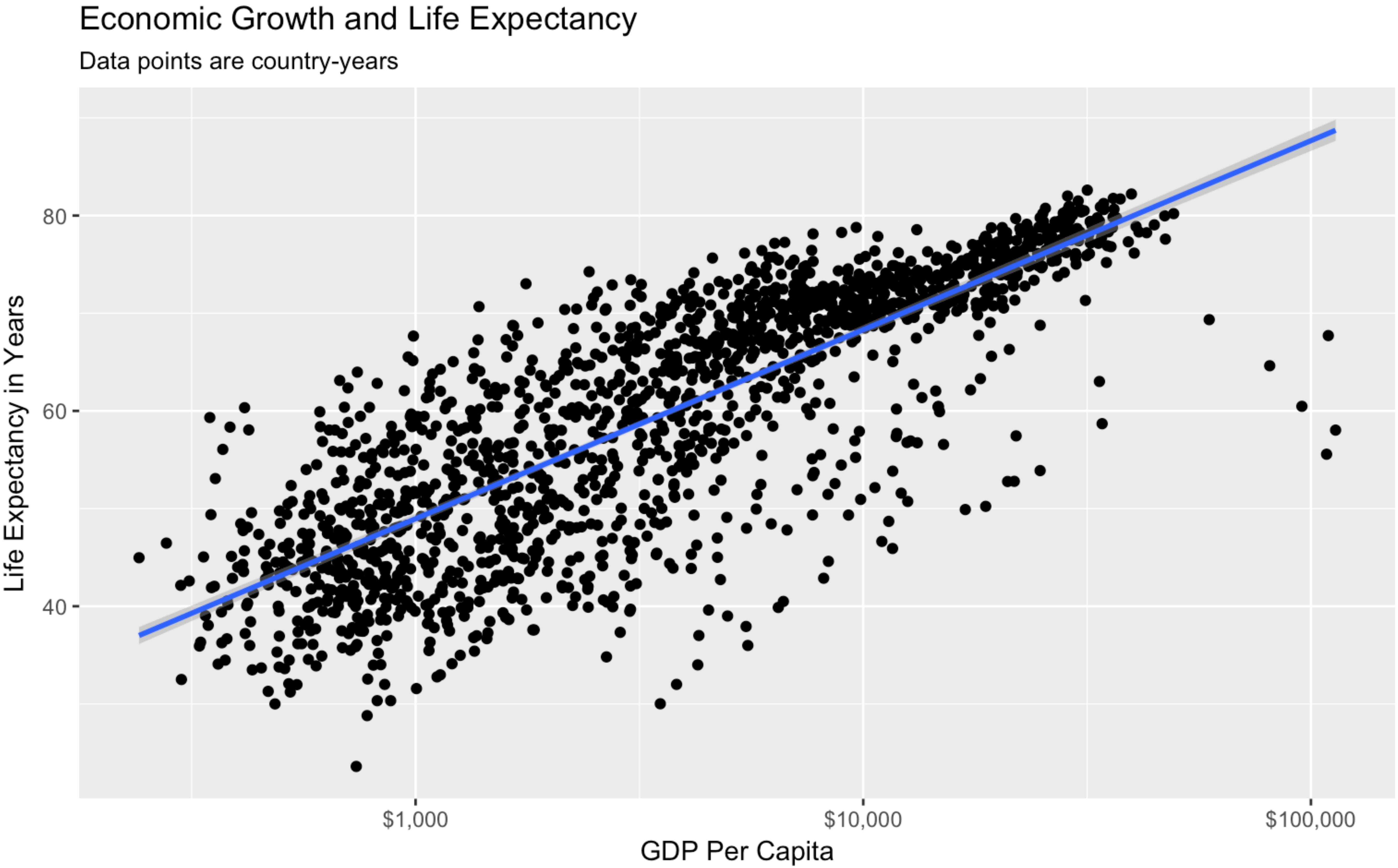
```r
p <- ggplot(data = gapminder,
            mapping = aes(x = gdpPercap,
                          y=lifeExp))
p + geom_point() +
    geom_smooth(method='gam') +
    scale_x_log10()
```

Keep Layering



80
60
40
lifeExp
$1,000
$10,000
$100,000
gdpPercap

```r
p + geom_point() +
    geom_smooth(method='gam') +
    scale_x_log10(labels = scales::dollar) +
    labs(x = "GDP Per Capita",
         y = "Life Expectancy in Years",
         title = "Economic Growth and Life Expectancy",
         subtitle = "Data points are country-years")
```


Economic Growth and Life Expectancy
Data points are country-years
Life Expectancy in Years
80
60
40
$1,000
$10,000
$100,000
GDP Per Capita