

# Visualization

# Does inequality make people better off?

An argument that it does:

1. unequal rewards motivate work and innovation
2. work and innovation make everyone better off

An empirical prediction from that argument:

- ▶ Countries with higher income inequality will have higher economic output

## Defining concepts: Income inequality

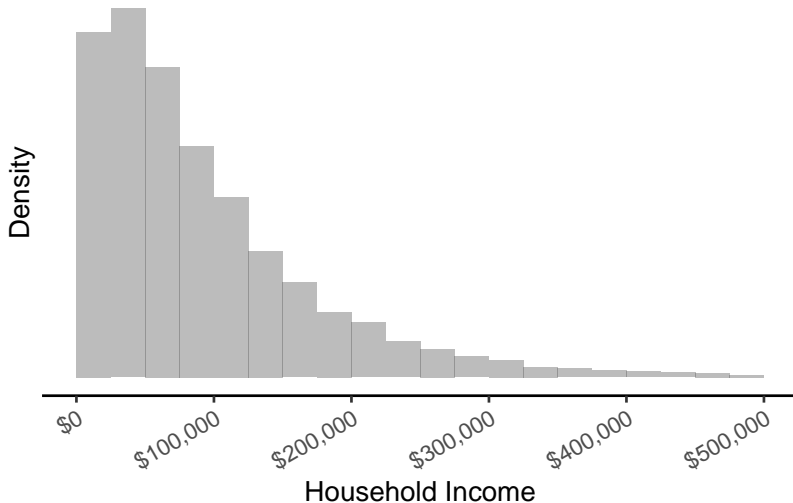
We will measure inequality using **quantiles**

- ▶ The 0.1 quantile (or 10th percentile)  
is the value such that 10% of households earn less
- ▶ The 0.5 quantile (or 50th percentile or median)  
is the value such that 50% of households earn less
- ▶ The 0.9 quantile (or 90th percentile)  
is the value such that 90% of households earn less

The 90/10 ratio is (90th percentile) / (10th percentile)

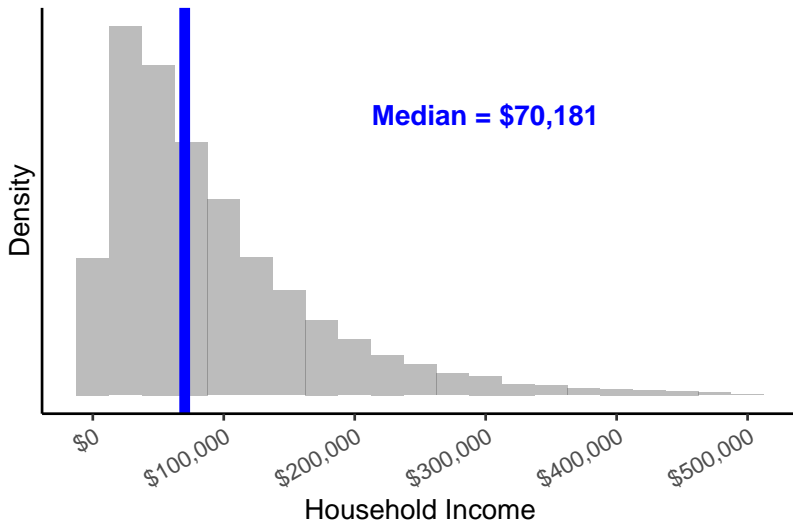
## Defining concepts: Income inequality

2022 Annual Social and Economic Supplement to the Current Population Survey (U.S. households)



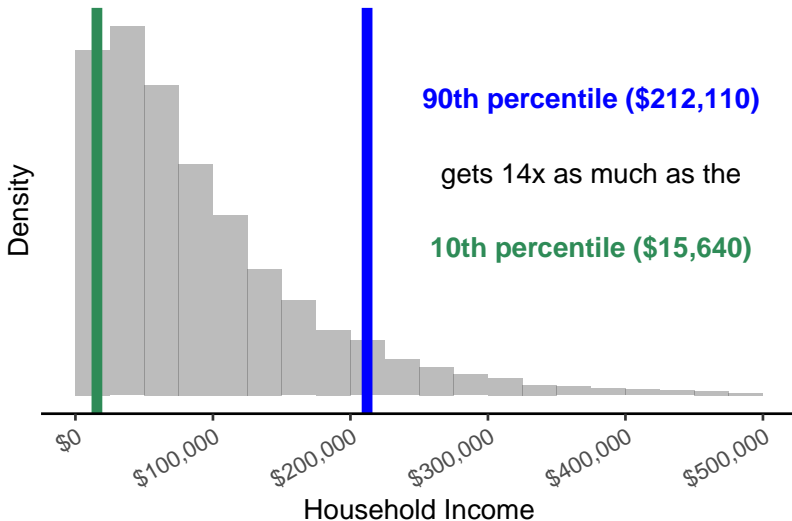
## Defining concepts: Income inequality

2022 Annual Social and Economic Supplement to the Current Population Survey (U.S. households)



# Defining concepts: Income inequality

2022 Annual Social and Economic Supplement to the Current Population Survey (U.S. households)



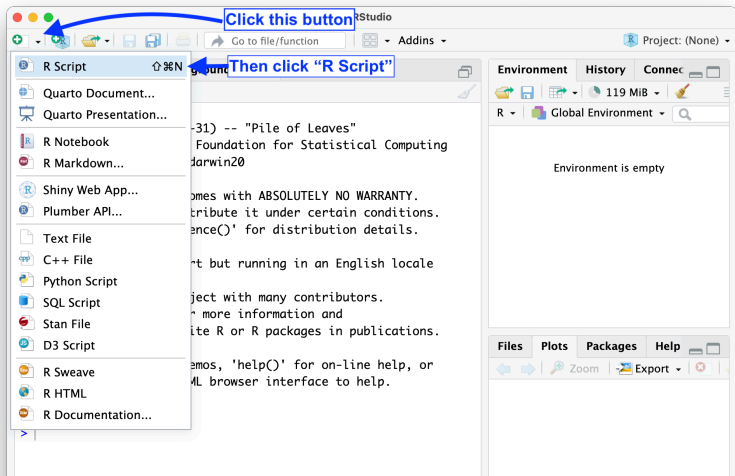
## Defining concepts: Economic productivity

- ▶ Gross Domestic Product Per Capita  
= (Total Economic Production) / (Number of People)  
*Prerequisites. You should first install R, RStudio, and the tidyverse package as described in the previous page.*

Visualizing data is an essential skill for data science. We will write our first code to visualize how countries' level of economic output is related to their level of inequality. We will use data reported in tabular form in Jencks 2002 Table 1, made available in digital form in `jencks_table1.csv`.

# Prepare the environment

Open a new R Script by clicking the button at the top left of RStudio. Save your R Script in a folder you will use for this exercise by clicking File -> Save from the menu at the very top of your screen.





# Explore the data

Type `table1` in your console. You can see the data!

```
> table1
# A tibble: 14 x 4
  country    ratio    gdp life_expectancy
  <chr>      <dbl> <dbl>      <dbl>
1 Sweden    2.6  0.68    78.9
2 Finland   2.7  0.68    76.6
3 Norway    2.8  0.85    77.8
4 Denmark   2.9  0.79    75.4
5 Netherlands 3.2  0.75    77.5
6 Germany   3.2  0.71    76.6
7 Belgium   3.2  0.74    76.4
8 France    3.5  0.66    78.4
9 Switzerland 3.6  0.84    78.5
10 Italy     4.8  0.67    77.6
11 Canada    4    0.78    78.2
12 Australia 4.3  0.75     78
13 UK        4.6  0.67     77
14 US        5.6  1       75.7
```

The data contain four variables (columns):

- ▶ `country` country name
- ▶ `ratio` ratio of 90th to 10th percentile of household income. You can think of this as how many dollars a high-income household receives for each dollar that a low-income household receives
- ▶ `gdp` Gross Domestic Product Per Capita, expressed as a proportion of U.S. GDP
- ▶ `life_expectancy` life expectancy at birth

## Produce a visualization

To visualize data, we will use the `ggplot()` function which you have already loaded into your R session as part of the `tidyverse` package.

### Begin with an empty graph

A function in R takes in **arguments** and returns an **object**. The arguments are the inputs that we give to the function. The function then returns something back to us.

The `ggplot()` function takes two arguments:

- ▶ `data = table1` says that data will come from the object `table1`
- ▶ `mapping = aes(x = ratio, y = gdp)` maps the data to the aesthetics of the graph. This line says that the `ratio` variable will be placed on the *x*-axis and the `gdp` variable will be on the *y*-axis.

When you run this code, the function returns an object which is the resulting plot. The plot will appear in the Plots pane in RStudio.

## Customizing your graph

Now it is your turn! Create additional layers with additional lines connected by +. Be creative! Here are some ideas:

- ▶ add a `geom_smooth()` layer to show the trend in the data
- ▶ add a `geom_text()` layer to label the points with country names. To do so, you will need to add the aesthetic mapping `label = country`.
  - ▶ Pro tip: If you want text to be far from points, install and load the `ggrepel` package and use `geom_text_repel`.
- ▶ add labels to your visualization for axes and titles
- ▶ label the axes with `scale_x_continuous(name = "your text here")` and `scale_y_continuous(name = "your text here")`

There are many possible graphs to make. An example is below.



## Interpret your graph

Once you are happy with your graph,

- ▶ write a few sentences explaining your graph
- ▶ discuss what questions you would like to ask next

<https://www.youtube.com/embed/YWTZ6xxRiP8>