

# Visualization and Summary Statistics

UCLA Soc 114

## Review of last class

Say this code in English:

```
numbers <- c(1,2,3)
x_length <- length(numbers)
```

## Review of last class

Say this code in English:

```
numbers <- c(1,2,3)
x_length <- length(numbers)
```

- ▶ Store the vector `c(1,2,3)` in the object `numbers`
- ▶ Use the `length()` function to get the length of `numbers`

# Learning goals for today

- ▶ Reason about distributions
  - ▶ and visualize with `ggplot()`
- ▶ Understand summary statistics
  - ▶ and construct with `summarize()`
- ▶ Write clean code with the pipe `|>`

## How to visualize?

How might we visualize the U.S. income distribution?

Here are some data:

```
library(tidyverse)
incomeSimulated <- read_csv(
  file = "https://soc114.github.io/data/incomeSimulated.csv"
)
```

```
# A tibble: 1,000 x 2
```

```
      id hhincome
  <dbl>   <dbl>
```

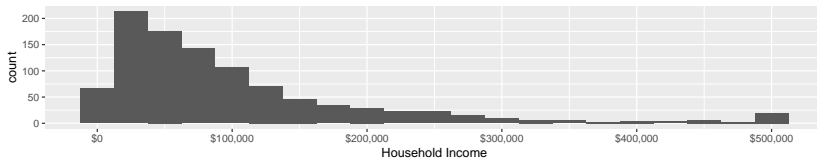
```
1     1    19170.
```

```
2     2   124474.
```

```
3     3    25114.
```

```
# i 997 more rows
```

# Visualize with a histogram



- ▶ Bins of \$25,000
- ▶ In each bin, count number of people

# Learning a new function: ggplot

```
ggplot(  
  data = incomeSimulated,  
  mapping = aes(x = hhincome)  
)
```



# Learning a new function: ggplot

```
ggplot(  
  data = incomeSimulated,  
  mapping = aes(x = hhincome)  
)
```



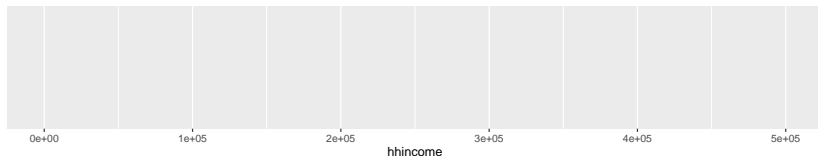
Two arguments get us started:

- ▶ data argument contains data
- ▶ mapping argument maps data to plot elements



# Learning a new function: ggplot

```
ggplot(  
  data = incomeSimulated,  
  mapping = aes(x = hhincome)  
)
```



Two arguments get us started:

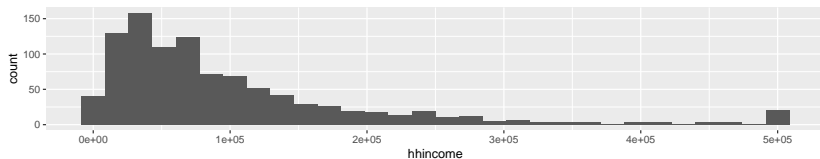
- ▶ data argument contains data
- ▶ mapping argument maps data to plot elements

Within mapping,

- ▶ aes() defines the aesthetics of the plot
- ▶ i.e. which variable goes along x-axis

## Adding a layer: geom\_histogram()

```
ggplot(  
  data = incomeSimulated,  
  mapping = aes(x = hhincome)  
) +  
  geom_histogram()
```



## Adding a layer: `geom_histogram()`

```
ggplot(  
  data = incomeSimulated,  
  mapping = aes(x = hhincome)  
) +  
  geom_histogram()
```



- ▶ The + indicates that a new layer is coming
- ▶ `geom_histogram()` is the new layer
- ▶ Inherits the data and mapping of the plot

# Update axis titles

```
ggplot(  
  data = incomeSimulated,  
  mapping = aes(x = hhincome)  
) +  
  geom_histogram() +  
  labs(  
    x = "Household Income",  
    y = "Count of Households in Bin"  
  )
```



# Learning goals for today

- ▶ Reason about distributions
  - ▶ and visualize with `ggplot()`
- ▶ Understand summary statistics
  - ▶ and construct with `summarize()`
- ▶ Write clean code with the pipe `|>`

## Imagine 3 income distributions

Household	Distribution 1	Distribution 2	Distribution 3
1	\$10k	\$40k	\$50k
2	\$60k	\$65k	\$60k
3	\$150k	\$70k	\$65k

Normative question: Which one is better?

# Summary statistic

A **summary statistic** aggregates a distribution to one number

# Summary statistic

A **summary statistic** aggregates a distribution to one number

For example, the mean

$$\text{mean}(\vec{x}) = \frac{x_1 + x_2 + \dots}{n}$$



## Summary statistic: Mean

Household	Distribution 1	Distribution 2	Distribution 3
1	\$10k	\$40k	\$50k
2	\$60k	\$65k	\$60k
3	\$150k	\$70k	\$65k
<b>Mean</b>	<b>\$73k</b>	<b>\$58k</b>	<b>\$58k</b>

## Summary statistic: Mean

Household	Distribution 1	Distribution 2	Distribution 3
1	\$10k	\$40k	\$50k
2	\$60k	\$65k	\$60k
3	\$150k	\$70k	\$65k
<b>Mean</b>	<b>\$73k</b>	<b>\$58k</b>	<b>\$58k</b>

By the mean, **Distribution 1** seems the best.

## Summary statistic: Median

1. Sort households by income.
2. Find where 50% of households have higher incomes

Household	Distribution 1	Distribution 2	Distribution 3
1	\$10k	\$40k	\$50k
2	\$60k	\$65k	\$60k
3	\$150k	\$70k	\$65k

## Summary statistic: Median

1. Sort households by income.
2. Find where 50% of households have higher incomes

Household	Distribution 1	Distribution 2	Distribution 3
1	\$10k	\$40k	\$50k
2	\$60k	\$65k	\$60k
3	\$150k	\$70k	\$65k
<b>Median</b>	<b>\$60k</b>	<b>\$65k</b>	<b>\$60k</b>

## Summary statistic: Median

1. Sort households by income.
2. Find where 50% of households have higher incomes

Household	Distribution 1	Distribution 2	Distribution 3
1	\$10k	\$40k	\$50k
2	\$60k	\$65k	\$60k
3	\$150k	\$70k	\$65k
<b>Median</b>	<b>\$60k</b>	<b>\$65k</b>	<b>\$60k</b>

By the median, **Distribution 2** seems the best.

## Aside: Percentiles generalize the median

The median is the value in the middle

- ▶ 50% of people have lower values
- ▶ Also called the 50th percentile

## Aside: Percentiles generalize the median

The median is the value in the middle

- ▶ 50% of people have lower values
- ▶ Also called the 50th percentile

Generalizes to other percentiles

- ▶ 10th percentile: Value such that 10% are lower
- ▶ 90th percentile: Value such that 90% are lower

These summarize the bottom and top of a distribution.

## Summary statistic: Minimum

Find the lowest value.

Household	Distribution 1	Distribution 2	Distribution 3
1	\$10k	\$40k	\$50k
2	\$60k	\$65k	\$60k
3	\$150k	\$70k	\$65k



## Summary statistic: Minimum

Find the lowest value.

Household	Distribution 1	Distribution 2	Distribution 3
1	\$10k	\$40k	\$50k
2	\$60k	\$65k	\$60k
3	\$150k	\$70k	\$65k
<b>Minimum</b>	<b>\$10k</b>	<b>\$40k</b>	<b>\$50k</b>

## Summary statistic: Minimum

Find the lowest value.

Household	Distribution 1	Distribution 2	Distribution 3
1	\$10k	\$40k	\$50k
2	\$60k	\$65k	\$60k
3	\$150k	\$70k	\$65k
<b>Minimum</b>	<b>\$10k</b>	<b>\$40k</b>	<b>\$50k</b>

By the minimum, **Distribution 3** seems the best.

## Which summary statistic to choose?

Minimum? Median? Mean?

Household	Distribution 1	Distribution 2	Distribution 3
1	\$10k	\$40k	\$50k
2	\$60k	\$65k	\$60k
3	\$150k	\$70k	\$65k

## Choosing a summary statistic

Which summary to choose is not an empirical question.

## Choosing a summary statistic

Which summary to choose is not an empirical question.

- ▶ Depends on what aspect of the distribution matters to you

# Choosing a summary statistic

Which summary to choose is not an empirical question.

- ▶ Depends on what aspect of the distribution matters to you

The value of a chosen summary statistic is empirical.

- ▶ Data can tell us a value for the mean, the median, the minimum, etc.

## The `summarize()` function

The `summarize()` function aggregates data to summaries.

- ▶ Input is a dataset with  $n$  rows
- ▶ Output is a summary with 1 row

## The summarize() function

```
incomeSimulated
```

```
# A tibble: 1,000 x 2
      id hhincome
  <dbl>   <dbl>
1     1    19170.
2     2   124474.
3     3    25114.
# i 997 more rows
```



## The summarize() function

```
summarize(  
  .data = incomeSimulated,  
  estimated_mean = mean(hhincome)  
)
```

```
# A tibble: 1 x 1  
  estimated_mean  
          <dbl>  
1         100899.
```

- ▶ .data is input data
- ▶ estimated\_mean is a variable in output data
- ▶ mean(hhincome) is the mean household income

## The summarize() function: Several summaries

```
summarize(  
  .data = incomeSimulated,  
  estimated_mean = mean(hhincome),  
  estimated_median = median(hhincome),  
  esitimated_min = min(hhincome)  
)
```

```
# A tibble: 1 x 3
```

	estimated_mean	estimated_median	esitimated_min
	<dbl>	<dbl>	<dbl>
1	100899.	69035.	0

# Learning goals for today

- ▶ Reason about distributions
  - ▶ and visualize with `ggplot()`
- ▶ Understand summary statistics
  - ▶ and construct with `summarize()`
- ▶ Write clean code with the pipe `|>`

## Piping code with |>

```
x <- c(1,2,3)  
length(x)
```

```
[1] 3
```

```
x |> length()
```

```
[1] 3
```

The pipe |> passes x as the first argument to the length() function.

## Piping code with |>

Stylistically helpful

- ▶ Data is a different kind of argument
- ▶ Pipes will help us in the future

```
incomeSimulated |>
  summarize(
    estimated_mean = mean(hhincome),
    estimated_median = median(hhincome),
    esitimated_min = min(hhincome)
  )
```

# A tibble: 1 x 3

	estimated_mean	estimated_median	esitimated_min
	<dbl>	<dbl>	<dbl>
1	100899.	69035.	0

# Learning goals for today

- ▶ Reason about distributions
  - ▶ and visualize with `ggplot()`
- ▶ Understand summary statistics
  - ▶ and construct with `summarize()`
- ▶ Write clean code with the pipe `|>`

You can now learn more: R4DS Ch 1 and Ch 3