Gov 51: Visualizing Distributions

Matthew Blackwell

Harvard University

Studying political efficacy

- 2002 WHO survey of people in China and Mexico.

Studying political efficacy

- 2002 WHO survey of people in China and Mexico.
- Goal: determine feelings of political efficacy.

Studying political efficacy

- 2002 WHO survey of people in China and Mexico.
- Goal: determine feelings of political efficacy.
- Question: “How much say do you have in getting the government to address issues that interest you?”

Studying political efficacy

- 2002 WHO survey of people in China and Mexico.
- Goal: determine feelings of political efficacy.
- Question: “How much say do you have in getting the government to address issues that interest you?”
    1. No say at all

Studying political efficacy

- 2002 WHO survey of people in China and Mexico.
- Goal: determine feelings of political efficacy.
- Question: “How much say do you have in getting the government to address issues that interest you?”
    1. No say at all
    2. little say

Studying political efficacy

- 2002 WHO survey of people in China and Mexico.
- Goal: determine feelings of political efficacy.
- Question: “How much say do you have in getting the government to address issues that interest you?”
    1. No say at all
    2. little say
    3. some say

Studying political efficacy

- 2002 WHO survey of people in China and Mexico.
- Goal: determine feelings of political efficacy.
- Question: “How much say do you have in getting the government to address issues that interest you?”
    1. No say at all
    2. little say
    3. some say
    4. a lot of say

Studying political efficacy

- 2002 WHO survey of people in China and Mexico.
- Goal: determine feelings of political efficacy.
- Question: “How much say do you have in getting the government to address issues that interest you?”
    1. No say at all
    2. little say
    3. some say
    4. a lot of say
    5. unlimited say

Data

- Load the data:

```
vignettes <- read.csv("data/vignettes.csv")
head(vignettes)
```

```
##   self alison jane moses china age
## 1    1      5    5     2     0  31
## 2    1      1    5     5     0  54
## 3    2      3    1     1     0  50
## 4    2      4    2     1     0  22
## 5    2      3    3     3     0  52
## 6    1      3    1     5     0  50
```

Contingency table

- `table()` shows how many units are in each category of a variable:

Contingency table

- `table()` shows how many units are in each category of a variable:

```
table(vignettes$self)
```

Contingency table

- `table()` shows how many units are in each category of a variable:

```
table(vignettes$self)
```

```
##
##   1   2   3   4   5
## 327 210 130  56  58
```

Contingency table

- `table()` shows how many units are in each category of a variable:

```
table(vignettes$self)
```

```
##
##   1   2   3   4   5
## 327 210 130  56  58
```

- `prop.table()` converts these counts into **proportions** of units:

Contingency table

- `table()` shows how many units are in each category of a variable:

```
table(vignettes$self)
```

```
##
##   1   2   3   4   5
## 327 210 130  56  58
```

- `prop.table()` converts these counts into **proportions** of units:

```
prop.table(table(vignettes$self))
```

Contingency table

- `table()` shows how many units are in each category of a variable:

```
table(vignettes$self)
```

```
##
##   1   2   3   4   5
## 327 210 130  56  58
```

- `prop.table()` converts these counts into **proportions** of units:

```
prop.table(table(vignettes$self))
```

```
##
##      1      2      3      4      5
## 0.4187 0.2689 0.1665 0.0717 0.0743
```

Contingency table

- `table()` shows how many units are in each category of a variable:

```
table(vignettes$self)
```

```
##
##   1   2   3   4   5
## 327 210 130  56  58
```

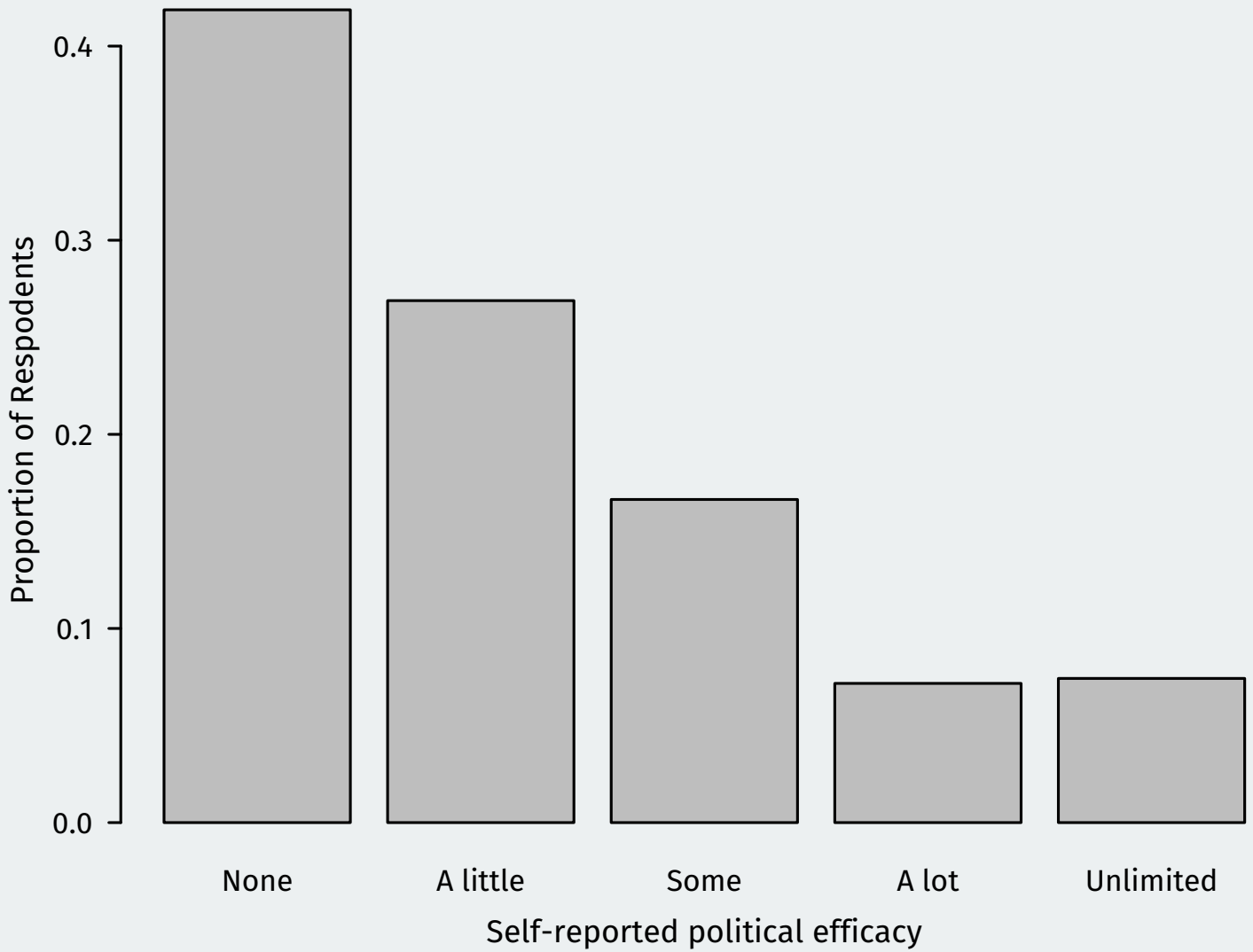
- `prop.table()` converts these counts into **proportions** of units:

```
prop.table(table(vignettes$self))
```

```
##
##      1      2      3      4      5
## 0.4187 0.2689 0.1665 0.0717 0.0743
```

- Useful way to visualize this information: **barplot**

Barplot example



Barplots in R

- The `barplot()` function can help us visualize a categorical variable:

Barplots in R

- The `barplot()` function can help us visualize a categorical variable:

```
barplot(height = prop.table(table(vignettes$self)),
        names = c("None", "A little",
                  "Some", "A lot", "Unlimited"),
        xlab = "Self-reported political efficacy",
        ylab = "Proportion of Respodents")
```

Barplots in R

- The `barplot()` function can help us visualize a categorical variable:

```
barplot(height = prop.table(table(vignettes$self)),
        names = c("None", "A little",
                  "Some", "A lot", "Unlimited"),
        xlab = "Self-reported political efficacy",
        ylab = "Proportion of Respodents")
```

- Arguments:

Barplots in R

- The `barplot()` function can help us visualize a categorical variable:

```
barplot(height = prop.table(table(vignettes$self)),
        names = c("None", "A little",
                  "Some", "A lot", "Unlimited"),
        xlab = "Self-reported political efficacy",
        ylab = "Proportion of Respodents")
```

- Arguments:
  - `height`: height each bar should take (proportions in this case)

Barplots in R

- The `barplot()` function can help us visualize a categorical variable:

```
barplot(height = prop.table(table(vignettes$self)),
        names = c("None", "A little",
                  "Some", "A lot", "Unlimited"),
        xlab = "Self-reported political efficacy",
        ylab = "Proportion of Respodents")
```

- Arguments:
  - `height`: height each bar should take (proportions in this case)
  - `names`: vector of labels for the each category/bar

Barplots in R

- The `barplot()` function can help us visualize a categorical variable:

```
barplot(height = prop.table(table(vignettes$self)),
        names = c("None", "A little",
                  "Some", "A lot", "Unlimited"),
        xlab = "Self-reported political efficacy",
        ylab = "Proportion of Respodents")
```

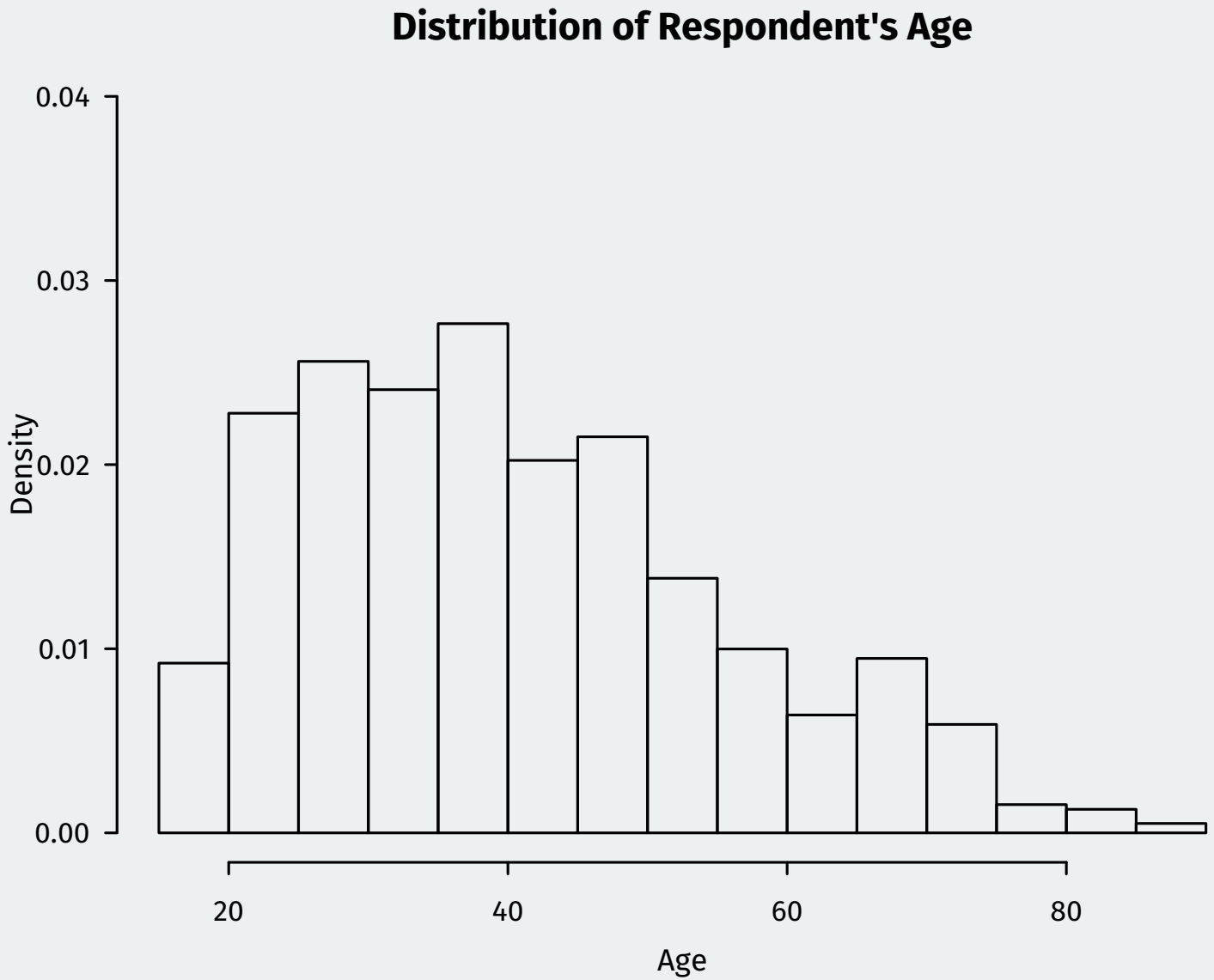
- Arguments:
  - `height`: height each bar should take (proportions in this case)
  - `names`: vector of labels for the each category/bar
  - `xlab`, `ylab` are axis labels

Histogram

- **Histograms** visualize density of continuous/numeric variable.

Histogram

- **Histograms** visualize density of continuous/numeric variable.



How to create histograms?

- How to create a histogram by hand:

How to create histograms?

- How to create a histogram by hand:
    1. create bins along the variable of interest

How to create histograms?

- How to create a histogram by hand:
    1. create bins along the variable of interest
    2. count number of observations in each bin

How to create histograms?

- How to create a histogram by hand:

    1. create bins along the variable of interest
    2. count number of observations in each bin
    3. **density** = bin height

$$\text{density} = \frac{\text{proportion of observations in bin}}{\text{bin width}}$$

How to create histograms?

- How to create a histogram by hand:

    1. create bins along the variable of interest
    2. count number of observations in each bin
    3. **density** = bin height

$$\text{density} = \frac{\text{proportion of observations in bin}}{\text{bin width}}$$

- The areas of the bins $=$ proportion of observations in those bins.

How to create histograms?

- How to create a histogram by hand:

  1. create bins along the variable of interest
  2. count number of observations in each bin
  3. **density** = bin height

$$\text{density} = \frac{\text{proportion of observations in bin}}{\text{bin width}}$$

- The areas of the bins $=$ proportion of observations in those bins.
  - $\rightsquigarrow$ area of the blocks sum to 1 (100%)

How to create histograms?

- How to create a histogram by hand:

  1. create bins along the variable of interest
  2. count number of observations in each bin
  3. **density** = bin height

$$\text{density} = \frac{\text{proportion of observations in bin}}{\text{bin width}}$$

- The areas of the bins $=$ proportion of observations in those bins.
  - $\rightsquigarrow$ area of the blocks sum to 1 (100%)
  - Can lead to confusion: height of block can go above 1!

How to create histograms?

- How to create a histogram by hand:

  1. create bins along the variable of interest
  2. count number of observations in each bin
  3. **density** = bin height

$$\text{density} = \frac{\text{proportion of observations in bin}}{\text{bin width}}$$

- The areas of the bins $=$ proportion of observations in those bins.
  - $\rightsquigarrow$ area of the blocks sum to 1 (100%)
  - Can lead to confusion: height of block can go above 1!
  - With equal-width bins, height is proportional to proportion in bin.

Histograms in R

- In R, we use `hist()` with `freq = FALSE`:

Histograms in R

- In R, we use `hist()` with `freq = FALSE`:

```
hist(x = vignettes$age, freq = FALSE, ylim = c(0, 0.04),
     xlab = "Age", main = "Distribution of Respondent's Age")
```

Histograms in R

- In R, we use `hist()` with `freq = FALSE`:

```
hist(x = vignettes$age, freq = FALSE, ylim = c(0, 0.04),
     xlab = "Age", main = "Distribution of Respondent's Age")
```

- Other arguments:

Histograms in R

- In R, we use `hist()` with `freq = FALSE`:

```
hist(x = vignettes$age, freq = FALSE, ylim = c(0, 0.04),
     xlab = "Age", main = "Distribution of Respondent's Age")
```

- Other arguments:
  - `ylim` sets the range of the y-axis to show.

Histograms in R

- In R, we use `hist()` with `freq = FALSE`:

```
hist(x = vignettes$age, freq = FALSE, ylim = c(0, 0.04),
     xlab = "Age", main = "Distribution of Respondent's Age")
```

- Other arguments:
  - `ylim` sets the range of the y-axis to show.
  - `main` sets the title for the figure.

Histograms in R

- In R, we use `hist()` with `freq = FALSE`:

```
hist(x = vignettes$age, freq = FALSE, ylim = c(0, 0.04),
     xlab = "Age", main = "Distribution of Respondent's Age")
```

- Other arguments:
  - `ylim` sets the range of the y-axis to show.
  - `main` sets the title for the figure.
- We can also choose the bin locations on our own via:

Histograms in R

- In R, we use `hist()` with `freq = FALSE`:

```
hist(x = vignettes$age, freq = FALSE, ylim = c(0, 0.04),
     xlab = "Age", main = "Distribution of Respondent's Age")
```

- Other arguments:
  - `ylim` sets the range of the y-axis to show.
  - `main` sets the title for the figure.
- We can also choose the bin locations on our own via:
  - `breaks`: location of the bin breaks, or

Histograms in R

- In R, we use `hist()` with `freq = FALSE`:

```
hist(x = vignettes$age, freq = FALSE, ylim = c(0, 0.04),
     xlab = "Age", main = "Distribution of Respondent's Age")
```

- Other arguments:
  - `ylim` sets the range of the y-axis to show.
  - `main` sets the title for the figure.
- We can also choose the bin locations on our own via:
  - `breaks`: location of the bin breaks, or
  - `nclass` (number of bins)

Histograms in R

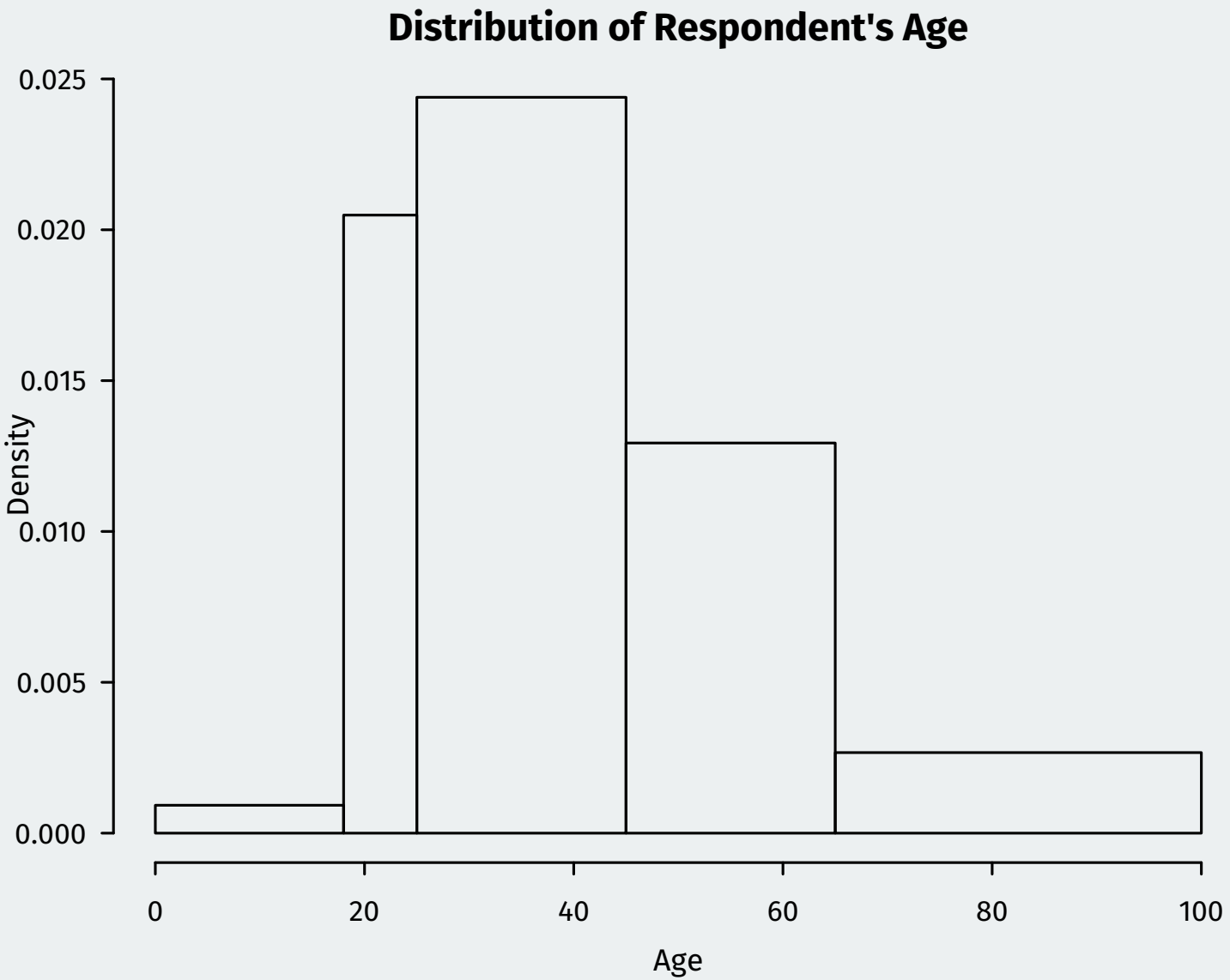
- In R, we use `hist()` with `freq = FALSE`:

```
hist(x = vignettes$age, freq = FALSE, ylim = c(0, 0.04),
     xlab = "Age", main = "Distribution of Respondent's Age")
```

- Other arguments:
  - `ylim` sets the range of the y-axis to show.
  - `main` sets the title for the figure.
- We can also choose the bin locations on our own via:
  - `breaks`: location of the bin breaks, or
  - `nclass` (number of bins)

```
hist(vignettes$age, freq = FALSE,
     breaks = c(0, 18, 25, 45, 65, 100),
     xlab = "Age",
     main = "Distribution of Respondent's Age")
```

Creating our own bins

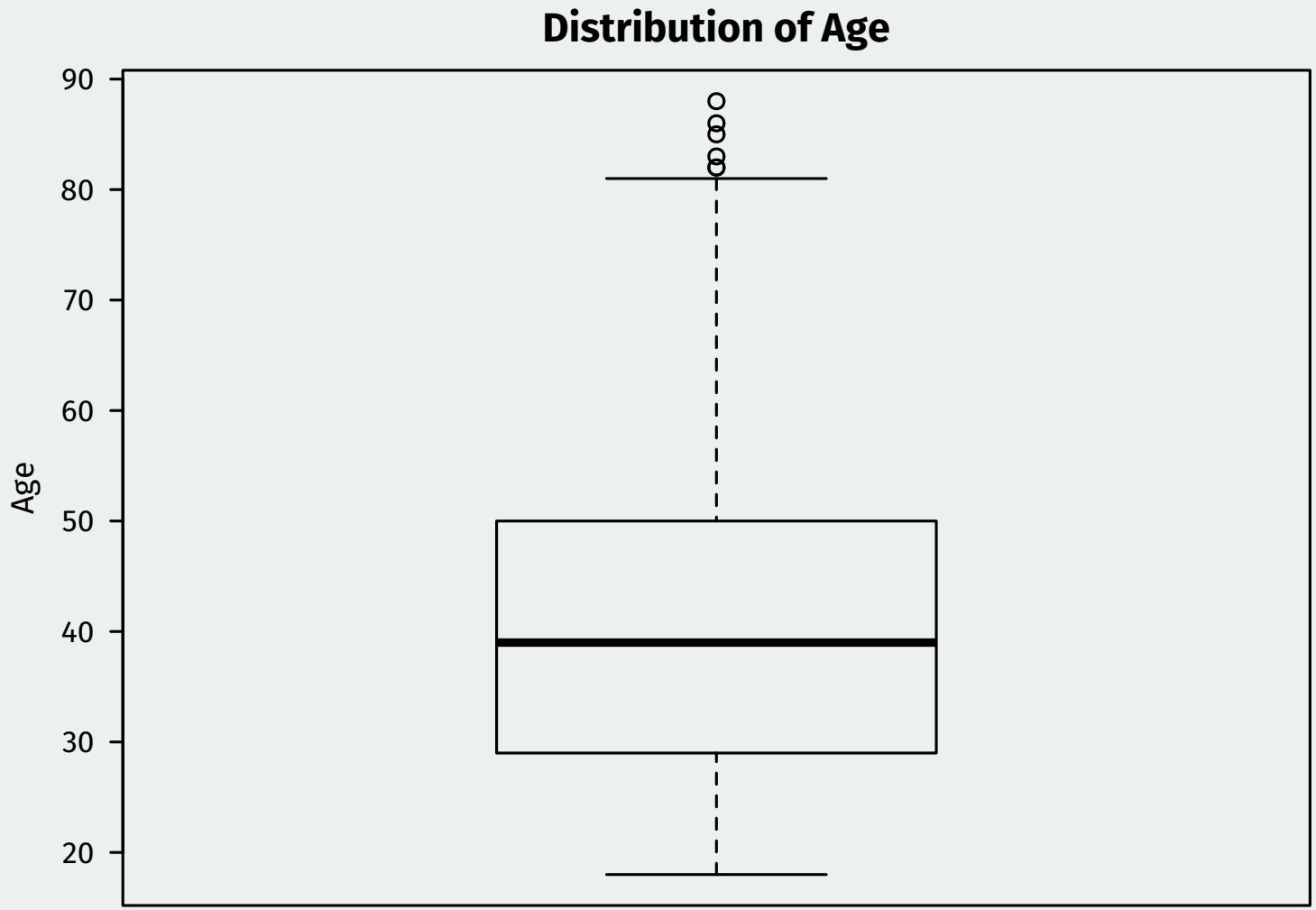


Boxplot

- A **boxplot** can characterize the distribution of continuous variables

Boxplot

- A **boxplot** can characterize the distribution of continuous variables



Boxplots in R

- “Box” represents range between lower and upper quartile.

Boxplots in R

- “Box” represents range between lower and upper quartile.
- “Whiskers” represents either:

Boxplots in R

- “Box” represents range between lower and upper quartile.
- “Whiskers” represents either:
    - 1.5 $\times$ IQR or max/min of the data, whichever is smaller.

Boxplots in R

- “Box” represents range between lower and upper quartile.
- “Whiskers” represents either:
    - 1.5 $\times$ IQR or max/min of the data, whichever is smaller.
    - Points beyond whiskers are outliers.

Boxplots in R

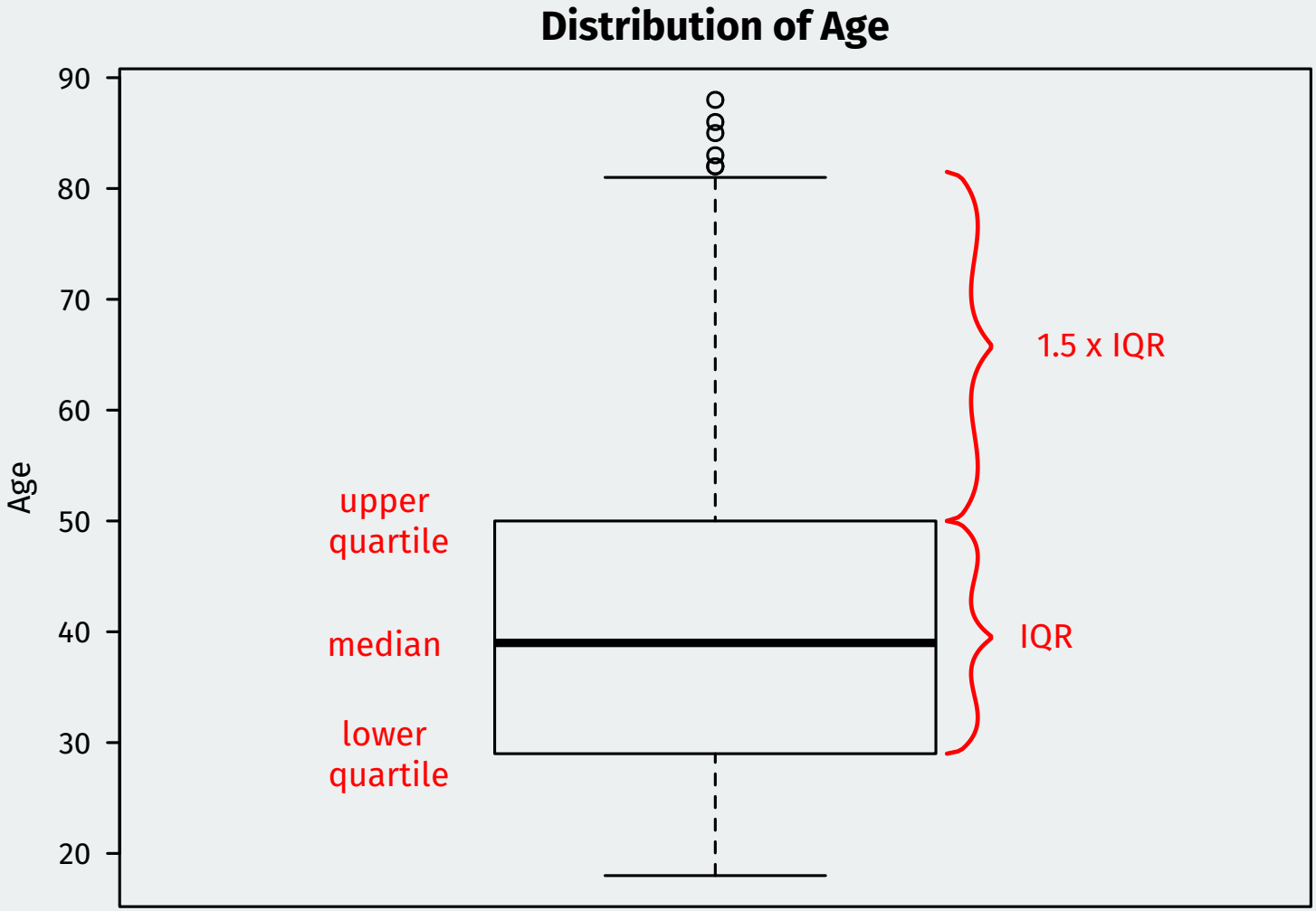
- "Box" represents range between lower and upper quartile.
- "Whiskers" represents either:
    - 1.5 $\times$ IQR or max/min of the data, whichever is smaller.
    - Points beyond whiskers are outliers.
- Use `boxplot()` in R:

Boxplots in R

- "Box" represents range between lower and upper quartile.
- "Whiskers" represents either:
  - 1.5 $\times$ IQR or max/min of the data, whichever is smaller.
  - Points beyond whiskers are outliers.
- Use `boxplot()` in R:

```
boxplot(vignettes$age, main = "Distribution of Age",
        ylab = "Age")
```

Boxplot


Distribution of Age
Age
90
80
70
60
50
40
30
20
upper
quartile
median
lower
quartile
1.5 x IQR
IQR

Review

- Visualizing single discrete/categorical variables: **barplots**

Review

- Visualizing single discrete/categorical variables: **barplots**
- Visualizing continuous variables: **histograms**, **boxplots**