PSC4375: Visualizing Distributions

Week 3: Lecture 5

Prof. Weldzius

Villanova University

Slides Updated: 2025-01-27

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
  2. little say
  3. some say
  4. a lot of say
  5. unlimited say

Data

- Load the data:

```
library(tidyverse)
data(vignettes, package = "qss")
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
```

```
## Also works if you downloaded the data:
# vignettes <- read.csv("data/vignettes.csv")
```

Contingency table

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

```
vignettes %>%
  count(self)
```

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

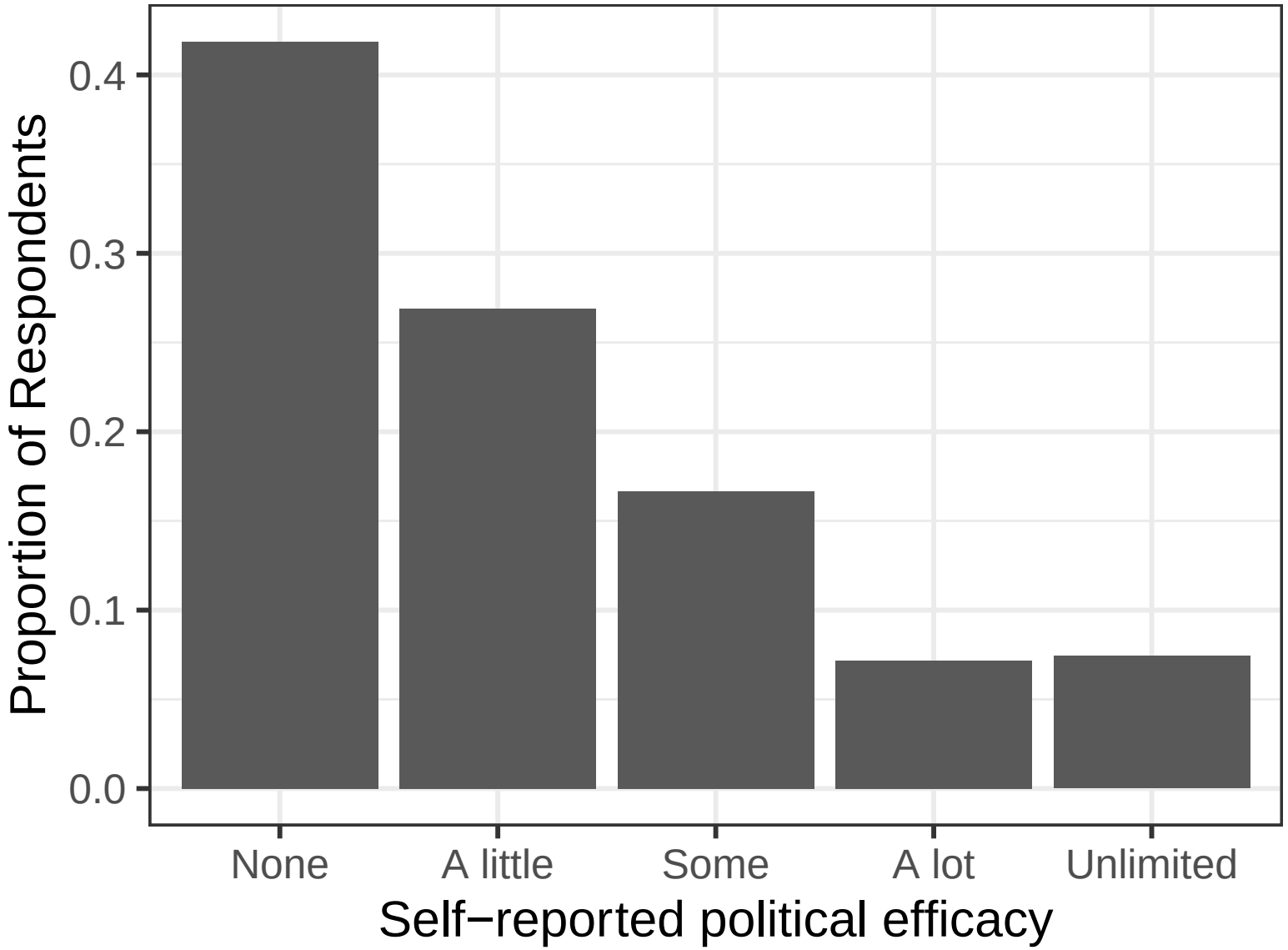
Contingency table (continued)

- using group_by we can convert these counts into proportions of units:

```
vignettes %>%
  group_by(self) %>%
  count() %>%
  ungroup() %>%
  mutate(prop = n / sum(n))
```

```
## # A tibble: 5 x 3
##    self     n   prop
##   <int> <int>  <dbl>
## 1     1   327 0.419
## 2     2   210 0.269
## 3     3   130 0.166
## 4     4    56 0.0717
## 5     5    58 0.0743
```

Useful way to visualize this information: barplot



Barplots in R

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

```
vignettes %>%
  ggplot(aes(x = as.factor(self),y=..prop..,group = 1)) +
  geom_bar() +
  scale_x_discrete(labels = c("None", "A little",
                   "Some", "A lot", "Unlimited")) +
  xlab("Self-reported political efficacy") +
  ylab("Proportion of Respondents") +
  theme_bw()
```

- Arguments:
  - `aes()`: the aestetic mapping of the plot (what you see)

Barplots in R

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

```
vignettes %>%
  ggplot(aes(x = as.factor(self),y=..prop..,group = 1)) +
  geom_bar() +
  scale_x_discrete(labels = c("None", "A little",
                   "Some", "A lot", "Unlimited")) +
  xlab("Self-reported political efficacy") +
  ylab("Proportion of Respondents") +
  theme_bw()
```

- Arguments:
  - `aes()`: the aestetic mapping of the plot (what you see)
  - `scale_x_discrete()`: changes the scale of the axis

Barplots in R

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

```
vignettes %>%
  ggplot(aes(x = as.factor(self),y=..prop..,group = 1)) +
  geom_bar() +
  scale_x_discrete(labels = c("None", "A little",
                   "Some", "A lot", "Unlimited")) +
  xlab("Self-reported political efficacy") +
  ylab("Proportion of Respondents") +
  theme_bw()
```

- Arguments:
  - `aes()`: the aestetic mapping of the plot (what you see)
  - `scale_x_discrete()`: changes the scale of the axis
  - `xlab()`, `ylab()` are axis labels

Barplots in R

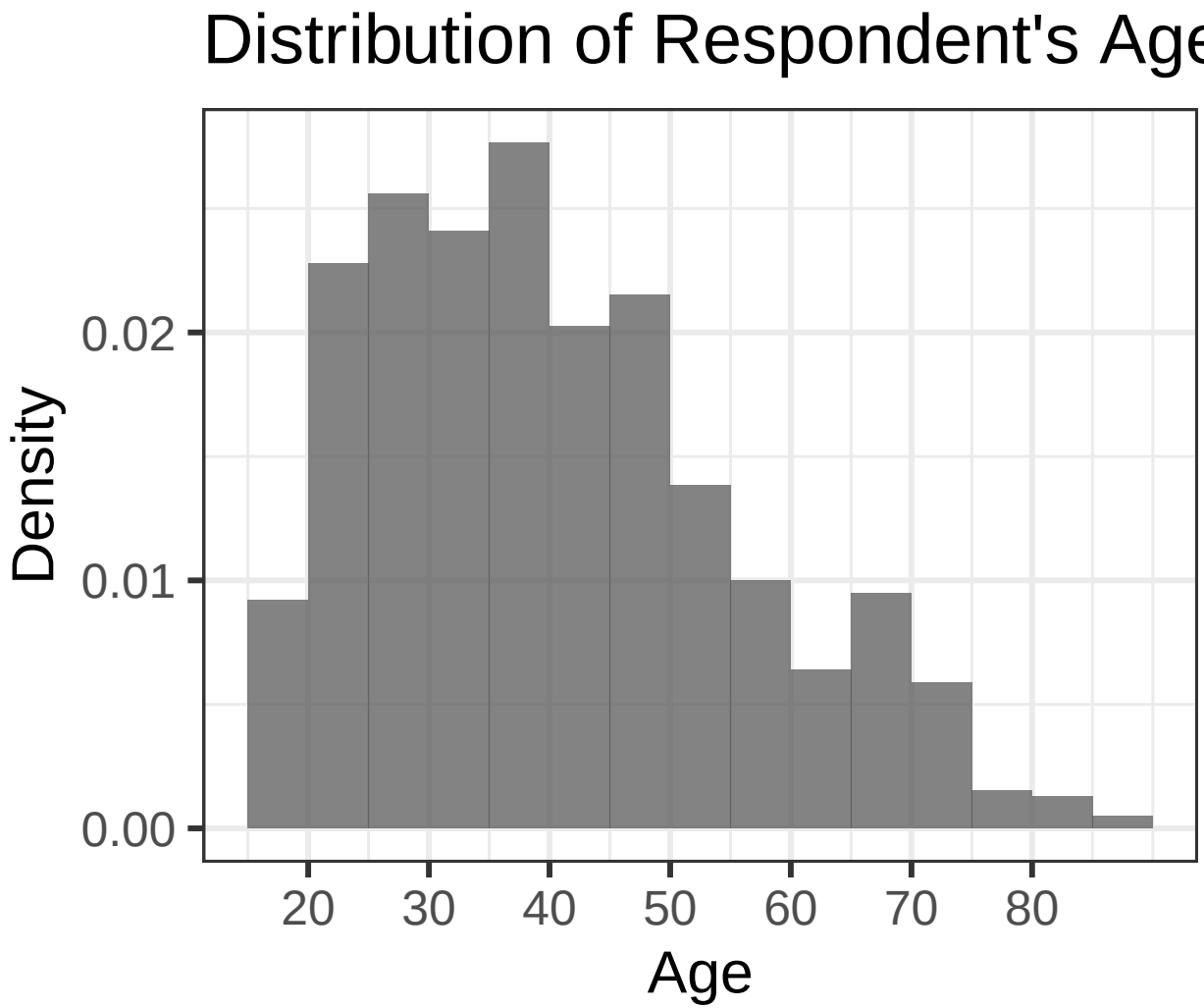
- The `barplot()` function can help us visualize a categorical variable:

```
vignettes %>%
  ggplot(aes(x = as.factor(self),y=..prop..,group = 1)) +
  geom_bar() +
  scale_x_discrete(labels = c("None", "A little",
                   "Some", "A lot", "Unlimited")) +
  xlab("Self-reported political efficacy") +
  ylab("Proportion of Respondents") +
  theme_bw()
```

- Arguments:
  - `aes()`: the aestetic mapping of the plot (what you see)
  - `scale_x_discrete()`: changes the scale of the axis
  - `xlab()`, `ylab()` are axis labels
  - `theme_bw()` removes grey background

Histogram

- **Histograms** visualize density of a 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.

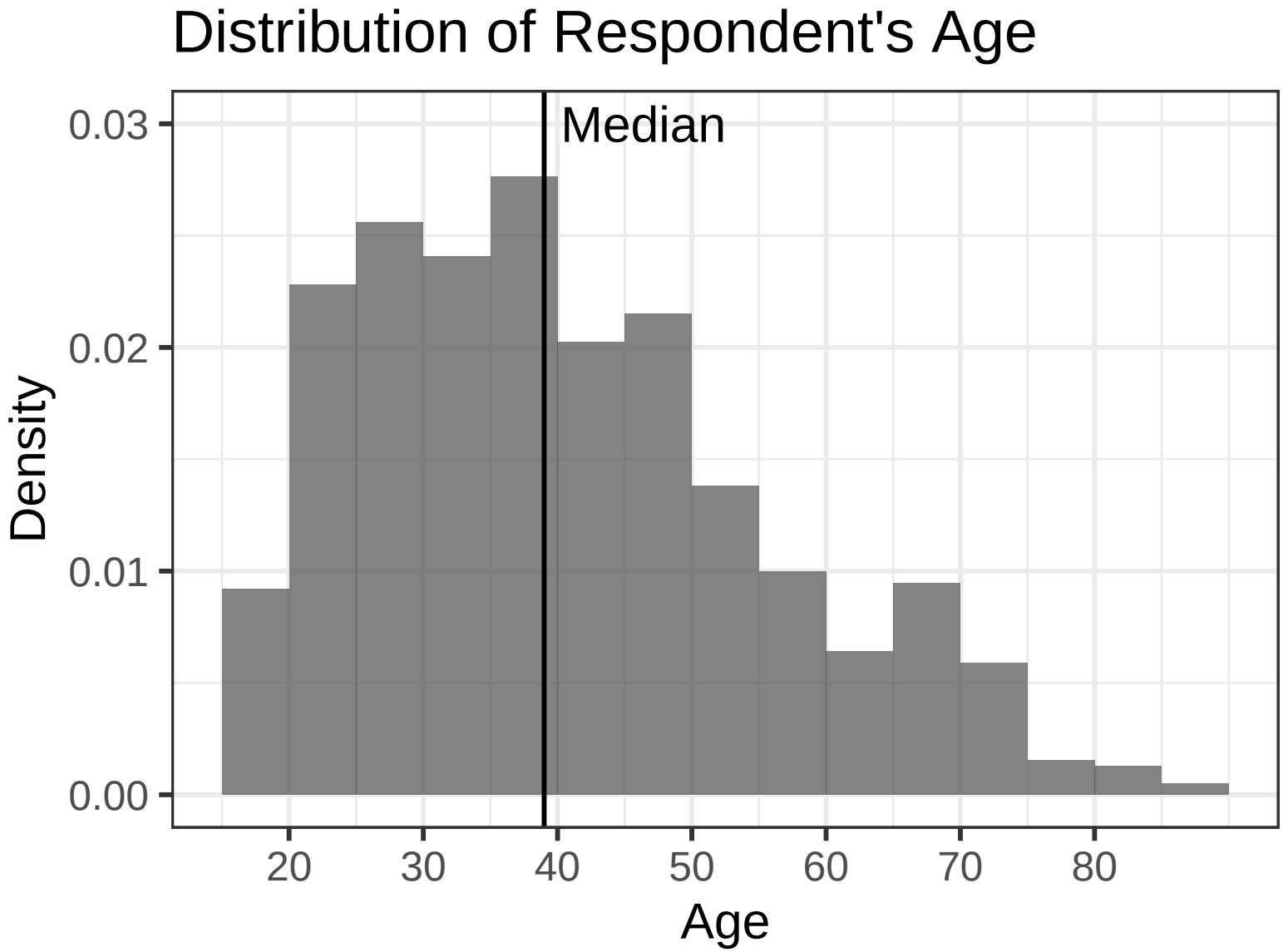
Histograms in R

- In ggplot, we use `geom_histogram()`:

```
vignettes %>%
  ggplot(aes(x = age,
             y = ..density..)) +
  geom_histogram(binwidth = 5, # how wide for each bin
                 boundary = 0, # bin position
                 alpha=0.75) + # reduces opaqueness
  scale_x_continuous(breaks = seq(20, 80,by = 10)) +
  labs(title = "Distribution of Respondent's Age",
       y = "Density",
       x = "Age") +
  theme_bw()
```

- Other arguments: -`labs` sets the titles for the plot (used `xlab` and `ylab` in previous plot) -`scale_x_continuous` sets the scale for the

Histograms in R: adding a vertical `median` line

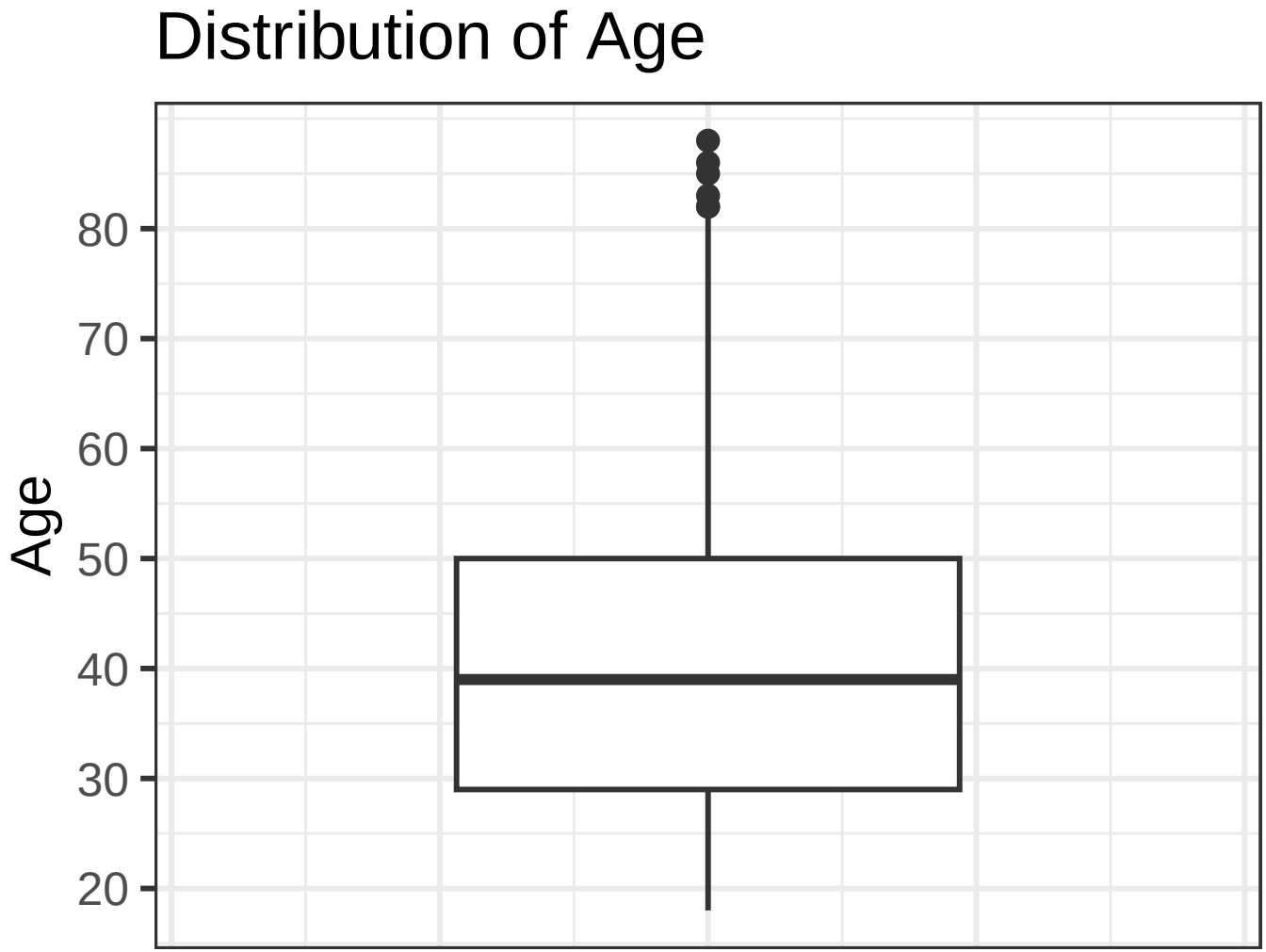


Histograms in R: adding a vertical `median` line

```
vignettes %>%
  ggplot(aes(x = age,
             y = ..density..)) +
  geom_histogram(binwidth = 5, # how wide for each bin
                 boundary = 0, # bin position
                 alpha=0.75) + # reduces opaqueness
  geom_vline(xintercept = median(vignettes$age)) +
  annotate(geom = "text", x = median(vignettes$age),
           y=.03, label = "Median", hjust = -0.1) +
  scale_x_continuous(breaks = seq(20, 80,by = 10)) +
  labs(title = "Distribution of Respondent's Age",
       y = "Density",
       x = "Age") +
  theme_bw()
```

Boxplot

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



Boxplots in R

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

Boxplots in R

```
vignettes %>%
  ggplot(aes(y = age)) +
  geom_boxplot() +
  scale_y_continuous(breaks = seq(20, 80,by = 10)) +
  xlim(-.75,.75) +
  labs(title = "Distribution of Age", y = "Age") +
  theme_bw() +
  theme(axis.text.x=element_blank(),
        axis.ticks.x=element_blank())
```

- Added options:
  - `scale_y_continuous`: scale the y axis
  - `xlim`: alter range of x-axis so box is less wide
  - `theme_bw`: removes grey background
  - `theme`: allows you to adjust other parts of figure

Review

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

Review

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