Data visualization

Graphs for a single continuous variable

Jenine Harris
Brown School

Graphs for a single continuous variable

- Three commonly used options are **histograms**, **density plots**, and **boxplots**.
- Histograms and density plots are very similar to each other and show the overall shape of the data.
    - These two types of graphs are especially useful in determining whether a variable has a **normal distribution** or not.
    - Boxplots show the central tendency and spread of the data, which are another way to determine whether a variable is normally distributed or skewed.
- **Violin plots** are also useful when looking at a continuous variable, and are like a combination of boxplots and density plots.
    - Violin plots are commonly used to examine the distribution of a continuous variable for different levels (or groups) of a factor (or categorical) variable.

Importing the data

- The data are from a 2017 study about funding and publication of gun research.
  - David E. Stark, Nigam H. Shah. Funding and Publication of Research on Gun Violence and Other Leading Causes of Death. JAMA, 2017; 317 (1): 84 DOI: 10.1001/jama.2016.16215

```
# import the data
research.funding <-
  read.csv(file = "/Users/harrisj/Box/teaching/Teaching/Fall2020/week-3/
# check out the data
summary(object = research.funding)
```

```
##  Cause.of.Death     Mortality.Rate.per.100.000.Population  Publications
##  Length:30          Min.   :  0.590                       Min.   :   1034
##  Class :character   1st Qu.:  1.775                       1st Qu.:  12550
##  Mode  :character   Median :  7.765                       Median :  39498
##                     Mean   : 22.419                       Mean   :  93914
##                     3rd Qu.: 14.812                       3rd Qu.:  54064
##                     Max.   :201.540                       Max.   :1078144
##     Funding           Predicted.Publications
##  Min.   :3.475e+06   Length:30
##  1st Qu.:3.580e+08   Class :character
##  Median :1.660e+09   Mode  :character
##  Mean   :4.137e+09
##  3rd Qu.:4.830e+09
##  Max.   :2.060e+10
##  Publications..Studentized.Residuals. Predicted.Funding
```

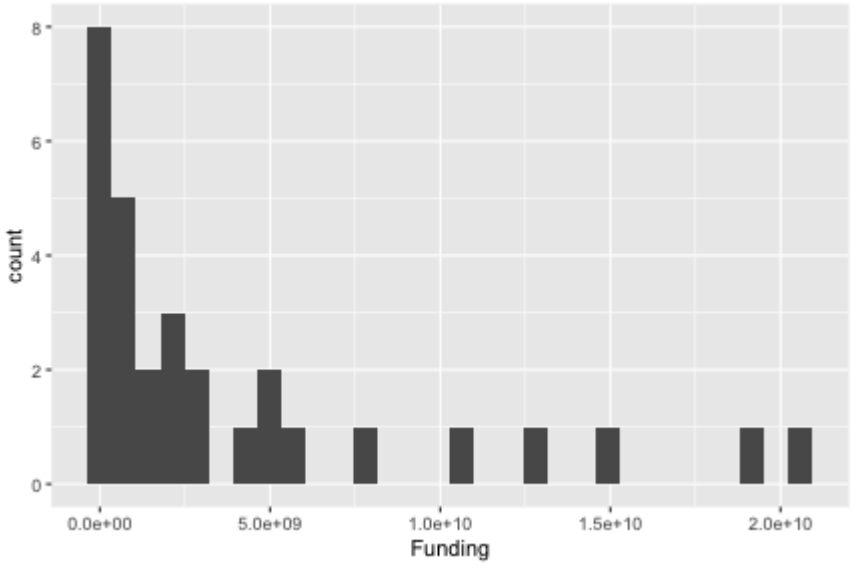
Creating a histogram with ggplot()

- Histograms can be developed with ggplot2().
- The geometry for a histogram is `geom_histogram()`.
- Graph the `Funding` variable to examine its distribution.

```
# open the tidyverse
library(package = "tidyverse")

# make a histogram of funding
histo.funding <- research.funding %>%
  ggplot(aes(x = Funding)) +
  geom_histogram()
histo.funding
```

Interpreting the histogram



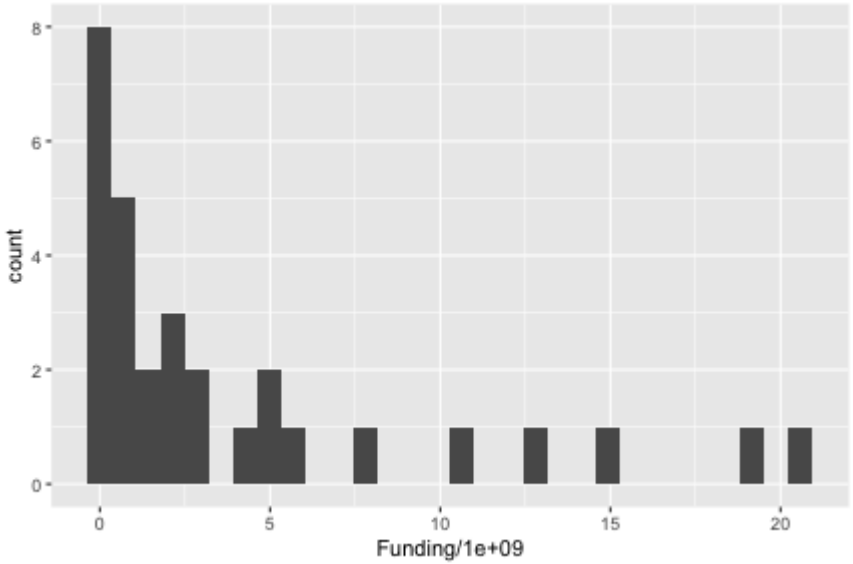
- The frequency is on the y-axis and funcing on the x-axis.
- The x-axis is shown using scientific notation because the numbers are so large.

Changing the axis values

- The histogram would be easier to read by changing the axis to show numbers that can be more easily interpreted.
- One strategy is to convert the numbers from *dollars* to *billions of dollars* by dividing the Funding variable by 1,000,000,000 within the `aes()` for the `ggplot()`.

```
# make a histogram of funding
histo.funding <- research.funding %>%
  ggplot(aes(x = Funding/1000000000)) +
  geom_histogram()
histo.funding
```

Viewing the graph with new axis values



- From the histogram it appears that most mortality causes are funded at between 0 and 5 billion dollars annually.
- However, several causes receive more than 5 billion and up to over 25 billion. The very large values on the right of the graph suggested that the **distribution** of funding is right skewed.

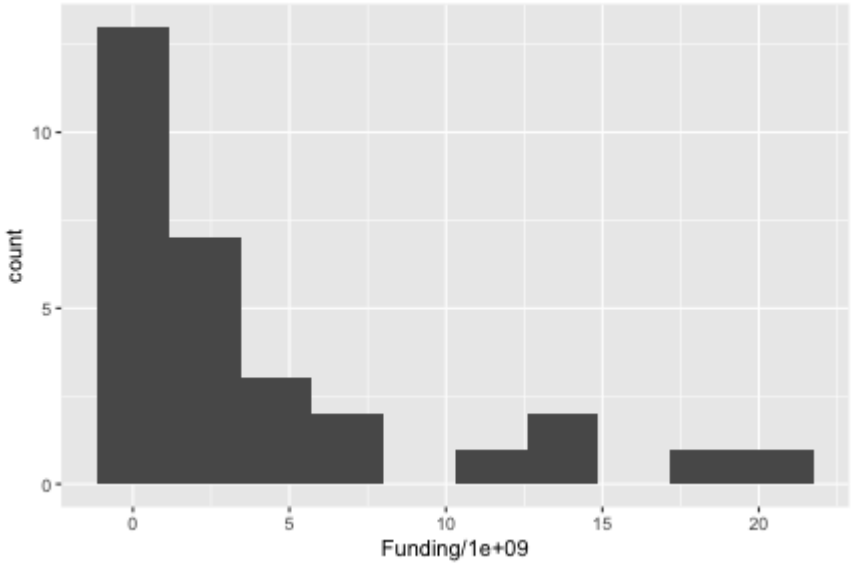
Working with bins in a histogram

- Each of the bars shown in the histogram is called a bin and contains a certain proportion of the observations.
- To show more bins, which may help to clarify the shape of the distribution, specify how many *bins* to see by adding `bins =` to the `geom_histogram()` layer.

```
# make a histogram of funding
# adjust the number of bins to 10
histo.funding <- research.funding %>%
  ggplot(aes(x = Funding/1000000000)) +
  geom_histogram(bins = 10)
histo.funding
```

Examining 10 bins

- The histogram with `bins = 10`.

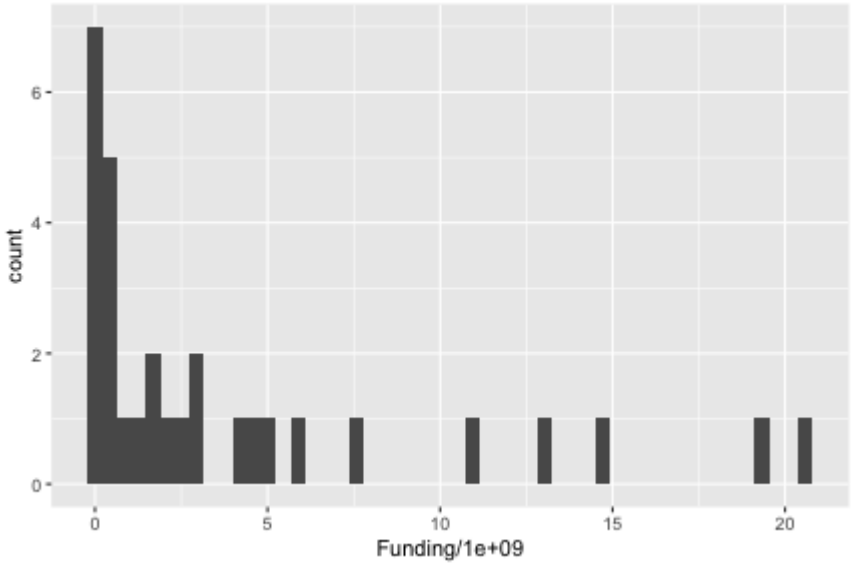


Using more bins

- Try 50 bins next to see if it changed the shape.

```
# make a histogram of funding
# adjust the number of bins to 50
histo.funding <- research.funding %>%
  ggplot(aes(x = Funding/1000000000)) +
  geom_histogram(bins = 50)
histo.funding
```

Examining the shape of the histogram with 50 bins

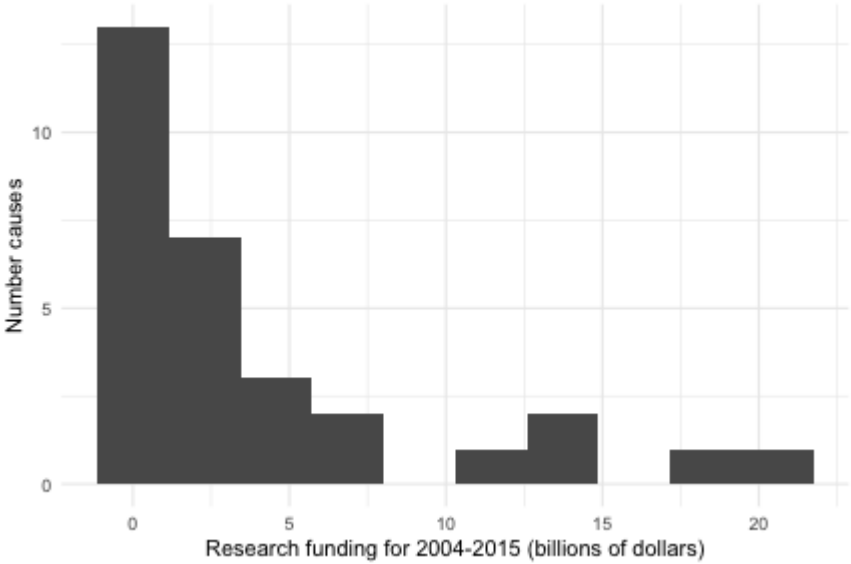


Formatting the histogram for reading and printing

- Add better titles for the axes in a `labs()` layer.
- Make the graph printer-friendly to use less ink by adding a `theme_minimal()` layer.

```
# make a histogram of funding
# adjust the number of bins to 10
histo.funding <- research.funding %>%
  ggplot(aes(x = Funding/1000000000)) +
  geom_histogram(bins = 10) +
  labs(x = "Research funding for 2004-2015 (billions of dollars)",
       y = "Number causes") +
  theme_minimal()
histo.funding
```

Reviewing the formatted histogram

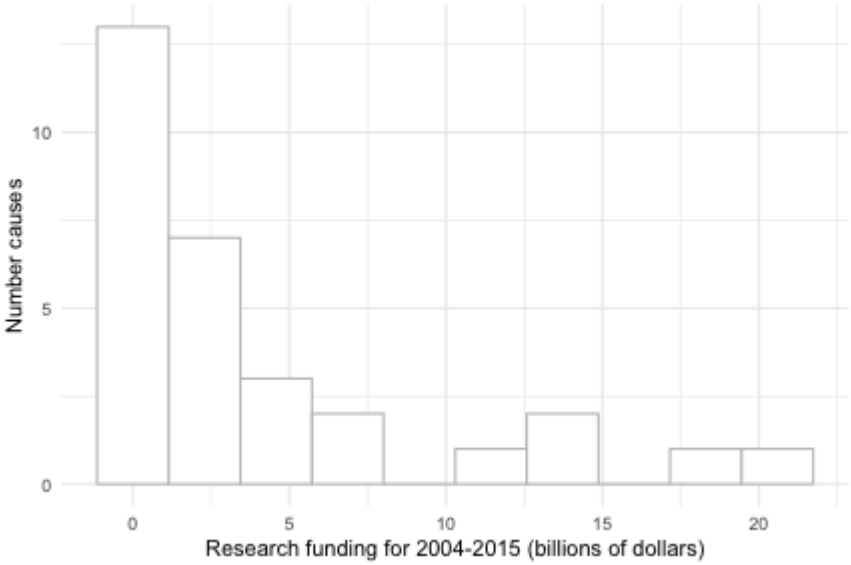


Additional formatting options for histograms

- Adding thin borders around the bins and filling the bins with white is another option.
- The geometry layer `geom_histogram()` can take arguments for `fill =`, which takes a color to fill each bin, and `color =`, which takes a color for the border of each bin.
- To add a `color` and `fill` not based on a variable, add these arguments to `geom_histogram()` *without* putting them in `aes()`.

```
# make a histogram of funding
# adjust the number of bins to 10
histo.funding <- research.funding %>%
  ggplot(aes(x = Funding/1000000000)) +
  geom_histogram(bins = 10, fill = "white", color = "gray") +
  labs(x = "Research funding for 2004-2015 (billions of dollars)",
       y = "Number causes") +
  theme_minimal()
histo.funding
```

Reviewing the histogram



- The histogram is right skewed and funding would therefore be best described using the median rather than the mean.
- The IQR would also probably be better than the range for reporting spread given how wide the range is.

Density plots

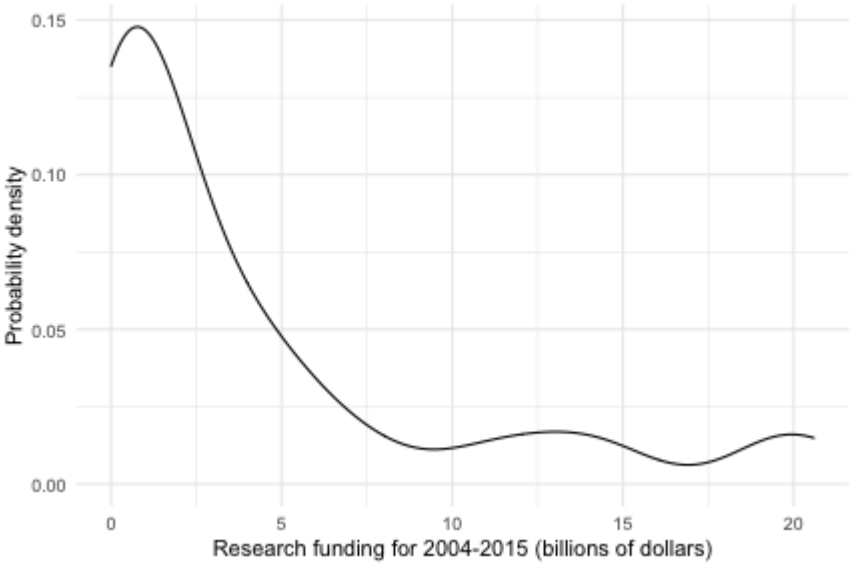
- A **density plot** is similar to a histogram but more fluid in appearance because it does not have the separate bins.
- Density plots can be created using `ggplot()` with a `geom_density()` layer.
- To create the density plot, replace the `geom_histogram()` layer.
- The y-axis is a different measure for this type of plot.
    - Instead of frequency it is the probability density, which is the probability of each value on the x-axis.
    - The probability density is not very useful for interpreting what is happening at any given value of the variable on the x-axis, but it is useful in computing the percentage of values that are within a range along the x-axis.
- The area under the curve adds up to 100 percent of the data and the height of the curve is determined by the distribution of the data, which is scaled so that the area will be 100 percent (or 1).

Labeling the y-axis for density plots

Change the `y =` option in the `labs()` layer to label the y-axis "Probability density"

```
# density plot of research funding
dens.funding <- research.funding %>%
  ggplot(aes(x = Funding/1000000000)) +
  geom_density() +
  labs(x = "Research funding for 2004-2015 (billions of dollars)",
       y = "Probability density") +
  theme_minimal()
dens.funding
```

Reviewing the density plot



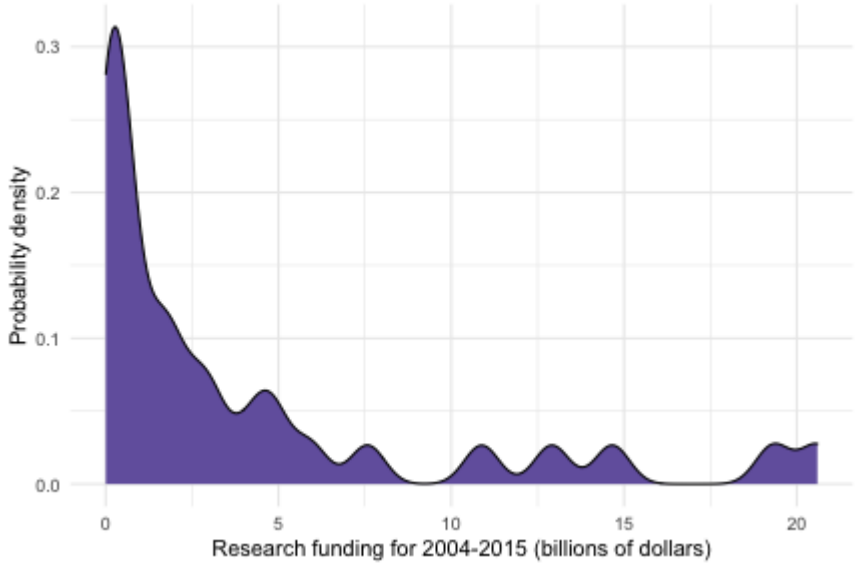
- The **area under the curve** in a density plot could be interpreted as the probability of a single observation or a range of observations.
- Probabilities are most useful when we have a **sample** taken from a **population** where a sample is a subgroup selected from a larger group.
- In this case we have *all* the top 30 mortality causes, so it is not a sample of the top mortality clauses, it is all of them.

Bandwidth and color in density plots

- Add color in order to be able to see the shape a little more.
- Try a few values of `bw =` within the `geom_density()`, noting that bw usually takes much smaller values than bin.
- The `bw` stands for *bandwidth* in a density plot, which is similar to the bin width in a histogram.

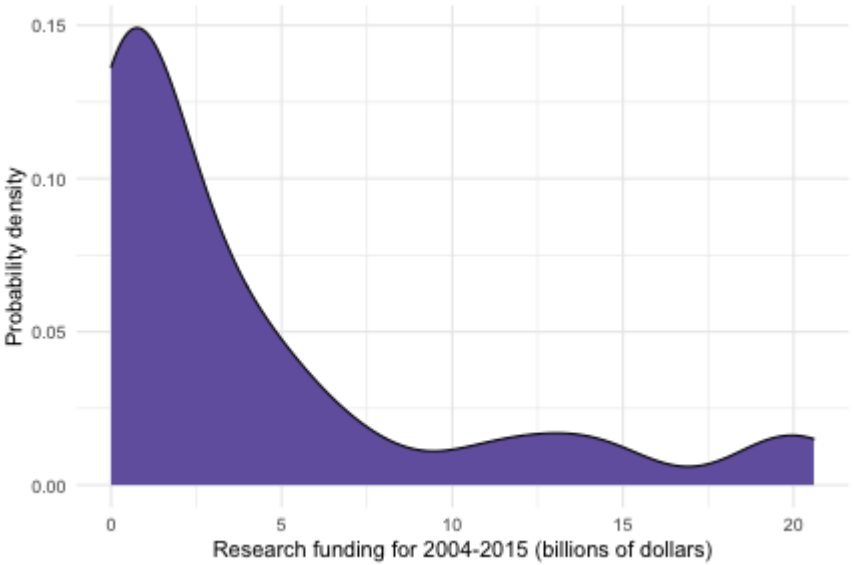
```
# density plot of research funding
# bw = .5
dens.funding <- research.funding %>%
  ggplot(aes(x = Funding/1000000000)) +
  geom_density(bw = .5, fill = "#7463AC") +
  labs(x = "Research funding for 2004-2015 (billions of dollars)",
       y = "Probability density") +
  theme_minimal()
dens.funding
```

Reviewing density plot with bw = .5



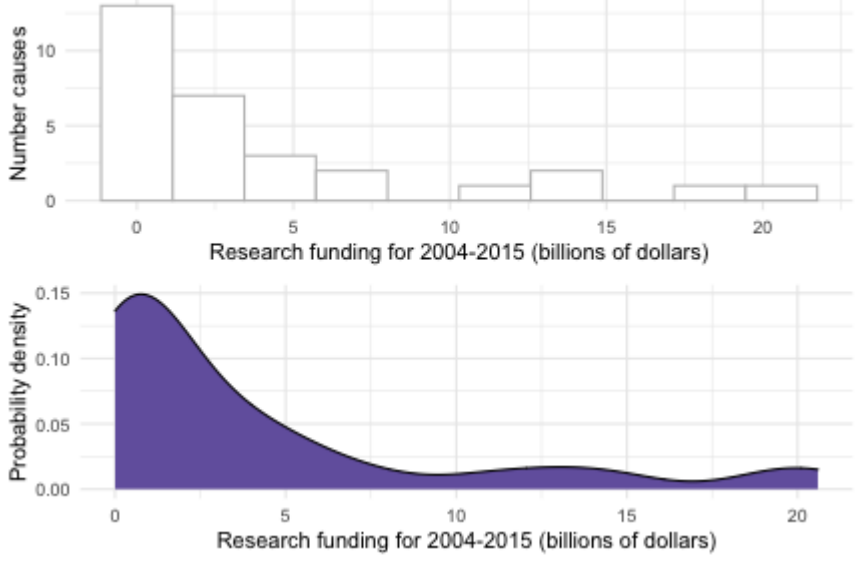
Density plot with bw = 1.5

```
# density plot of research funding
# bw = 1.5
dens.funding <- research.funding %>%
  ggplot(aes(x = Funding/1000000000)) +
  geom_density(bw = 1.5, fill = "#7463AC") +
  labs(x = "Research funding for 2004-2015 (billions of dollars)", y = "
  theme_minimal()
dens.funding
```



Density plot caveats

- The higher the value used as a bandwith in `bw =`, the smoother the graph looks.
- Although the bandwidth of 1.5 looks good, there is one feature that some data scientists suggest is misleading.
- Compare the density plot with the histogram, both from the same data;
    - The histogram shows gaps where there are no observations, while the density plot has the appearance of data continuing without gaps across the full range of values.
    - For this reason, data scientists sometimes recommend histograms over density plots, especially for small data sets where gaps in the data are more likely.

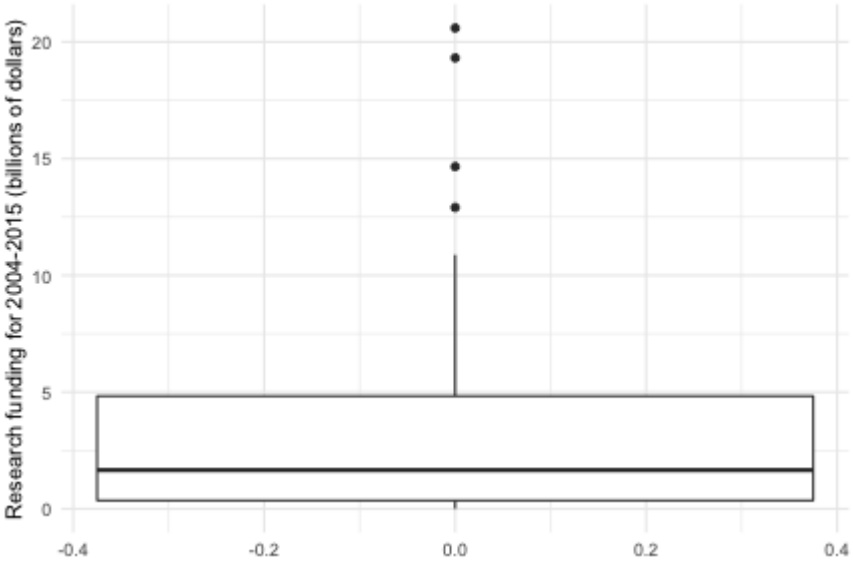


Boxplots

- Histograms and density plots are great for examining the overall shape of the data for a continuous variable, but that the boxplot was useful for identifying the middle value and the boundaries around the middle half of the data.
- Typically, boxplots consist of several parts:
    - A line representing the median value
    - A box containing the middle 50% of values
    - Whiskers extending to 1.5 times the IQR
    - Outliers more than 1.5 times the IQR away from the median
- In `ggplot()`, the boxplot uses the `geom_boxplot()`.
- Copy the density plot commands and change the `geom_` type.
- The boxplot will show the values of the variable along the y-axis by default, so instead of `x = Funding/1000000000`, use `y = Funding/1000000000` in the plot aesthetics, `aes()`.

R code for a boxplot

```
# boxplot of research funding
box.funding <- research.funding %>%
  ggplot(aes(y = Funding/1000000000)) +
  geom_boxplot() +
  theme_minimal() +
  labs(y = "Research funding for 2004-2015 (billions of dollars)")
box.funding
```



Flipping the coordinates of the boxplot

- For easier interpretation, add a new layer of `coord_flip()` to flip the coordinates so that what used to be on the y-axis is now on the x-axis and vice-versa.

```
# boxplot of research funding
box.funding <- research.funding %>%
  ggplot(aes(y = Funding/1000000000)) +
  geom_boxplot() +
  theme_minimal() +
  labs(y = "Research funding for 2004-2015 (billions of dollars)") +
  coord_flip()
box.funding
```

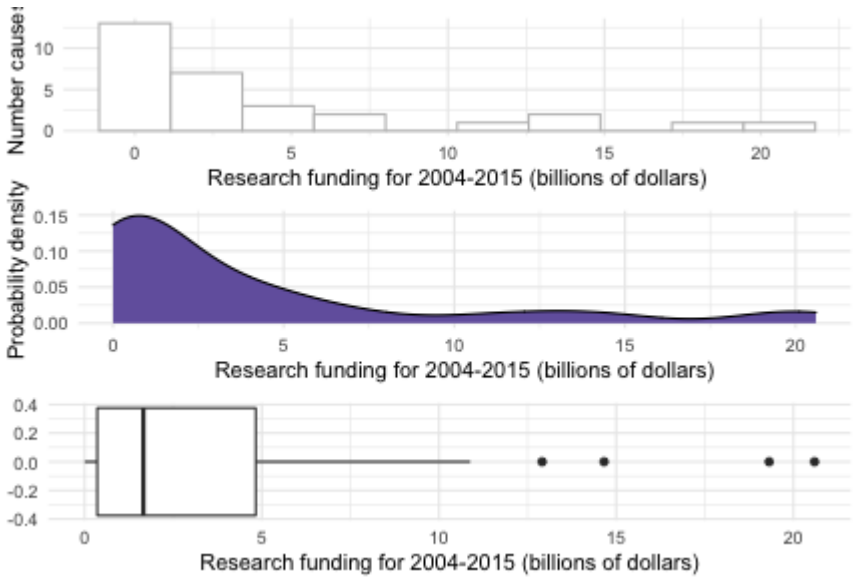
Interpreting the boxplot

- The median funding level was about 2 billion based on the location of the thick black line in the middle of the box.
- Based on the boundaries of the box, the middle half of the data appeared to be between about 1 and 5 billion dollars.
- Right skew shown in the histogram and density plot can also be seen in this graph, with the long whisker to the right of the box and the outliers on the far right.
- The left whisker coming from the box and the right whisker coming from the box both extend to 1.5 times the value of the IQR away from the median (the box ends at 1 IQR from the median).

Comparing the plots

- Use `gridExtra::grid.arrange()` to arrange the histogram, density plot, and boxplot together in order to see the similarities and differences between the three.
  - Use the option `nrow = 3` to display one graph per row rather than side-by-side in columns.

```
# plot all three options together
gridExtra::grid.arrange(histo.funding,
                        dens.funding,
                        box.funding,
                        nrow = 3)
```



Comparing and interpreting the graphs

- Looking at the three graphs together, it was clear that they tell a consistent story but there are some different pieces of information to be learned from the different types of graphs.
- All three graphs show the right skew clearly, while the histogram and boxplot show gaps in the data toward the end of the tail.
- The boxplot is the only one of the three that clearly identifies the central tendency and spread of the variable.

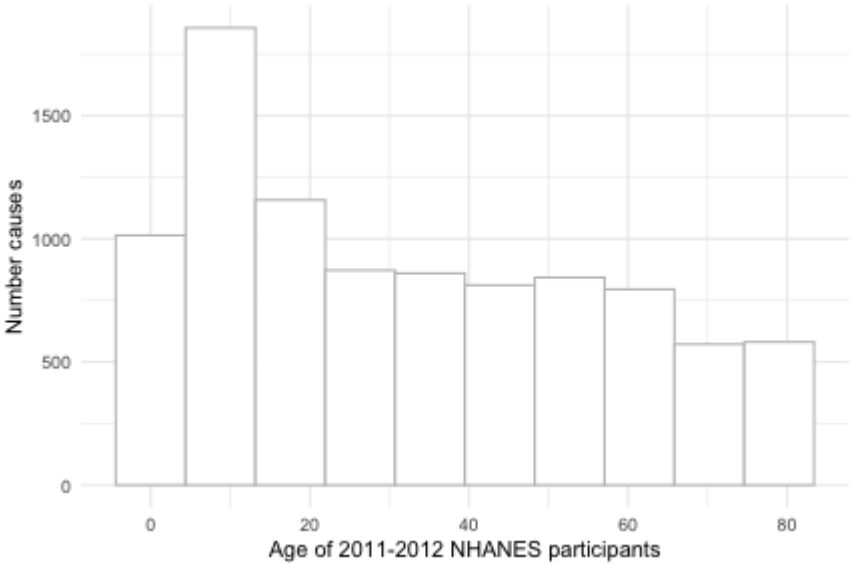
Check your understanding

Create a histogram, a boxplot, and a density plot to show the distribution of the age variable (`RIDAGEYR`) from the NHANES 2012 data set. Explain the distribution including an approximate value of the median, what the boundaries are around the middle 50% of the data, and a description of the skew (or lack of skew).

```
# import the data
nhanes.2012 <- read.csv(file = "data/nhanes_2011_2012_ch3.csv")
```

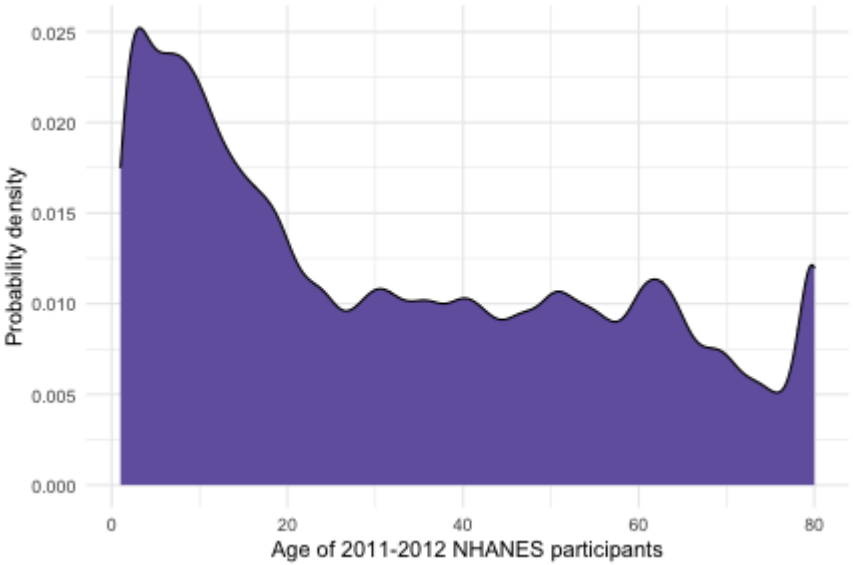
Histogram

```
# histogram of age
histo.age <- nhanes.2012 %>%
  ggplot(aes(x = RIDAGEYR)) +
  geom_histogram(bins = 10, fill = "white", color = "gray") +
  labs(x = "Age of 2011-2012 NHANES participants",
       y = "Number causes") +
  theme_minimal()
histo.age
```



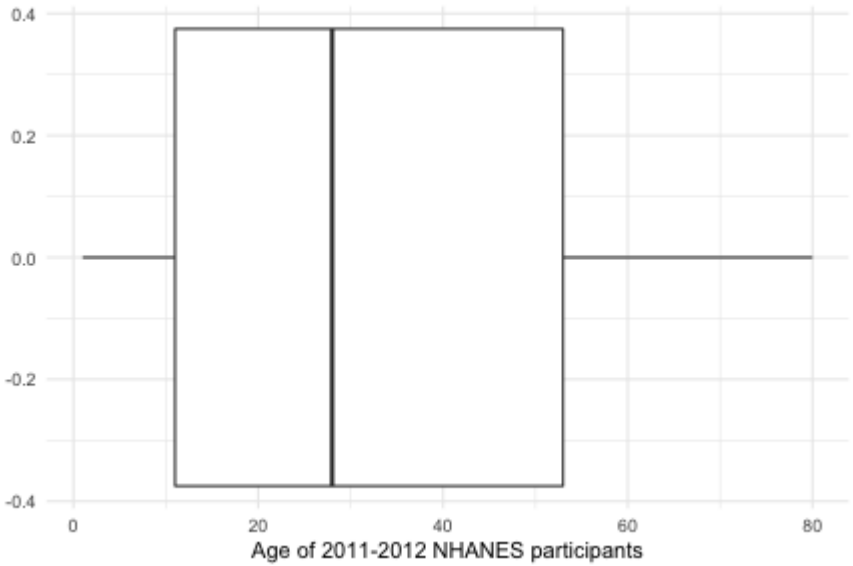
Density plot

```
# density plot of age
dens.age <- nhanes.2012 %>%
  ggplot(aes(x = RIDAGEYR)) +
  geom_density(bw = 1.5, fill = "#7463AC") +
  labs(x = "Age of 2011-2012 NHANES participants",
       y = "Probability density") +
  theme_minimal()
dens.age
```



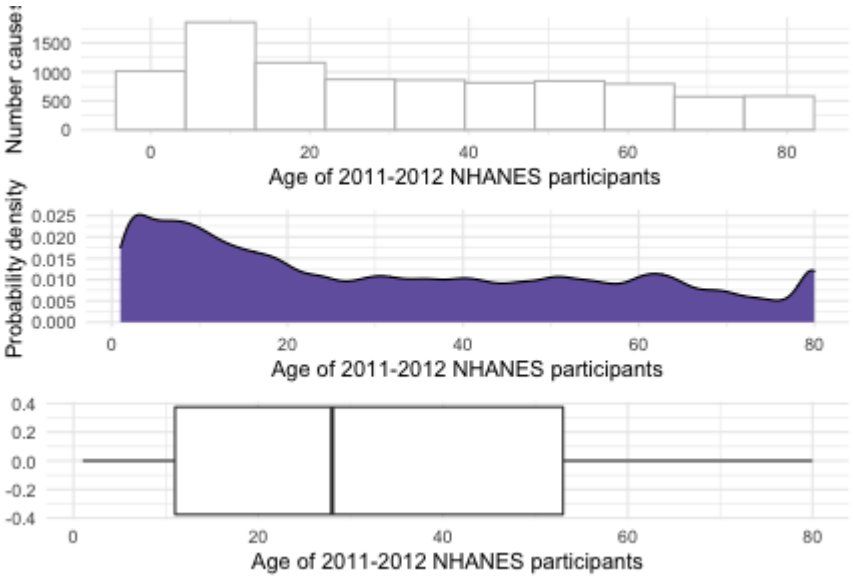
Boxplot

```
# boxplot of age
box.age <- nhanes.2012 %>%
  ggplot(aes(y = RIDAGEYR)) +
  geom_boxplot() +
  theme_minimal() +
  labs(y = "Age of 2011-2012 NHANES participants") +
  coord_flip()
box.age
```



Plot three graphs

```
# plot all three options together
gridExtra::grid.arrange(histo.age,
                        dens.age,
                        box.age,
                        nrow = 3)
```



Interpret the graphs

- Age looks a little right skewed
- The median is near 27 years old
- The boundaries around the middle 50% look like 11 years old to 52 years old approximately
- The age seems to range from near 0 to 80 years old