

# Distributions: part one

VISUALIZATION BEST PRACTICES IN R



**Nick Strayer**  
Instructor

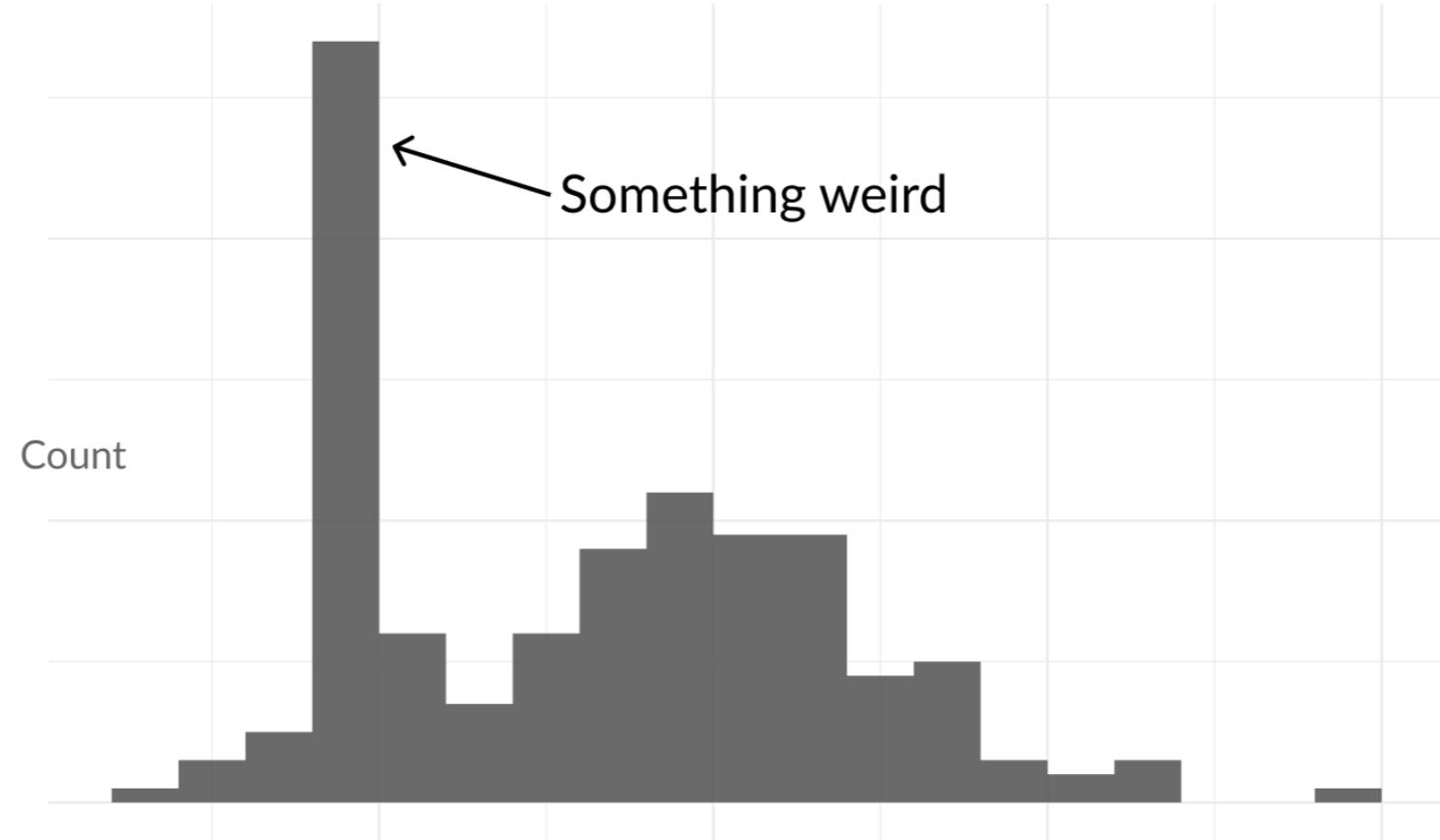
# What is distribution data?

- Multiple 'observations'
- Usually a sample of some population



# Why distributions are important

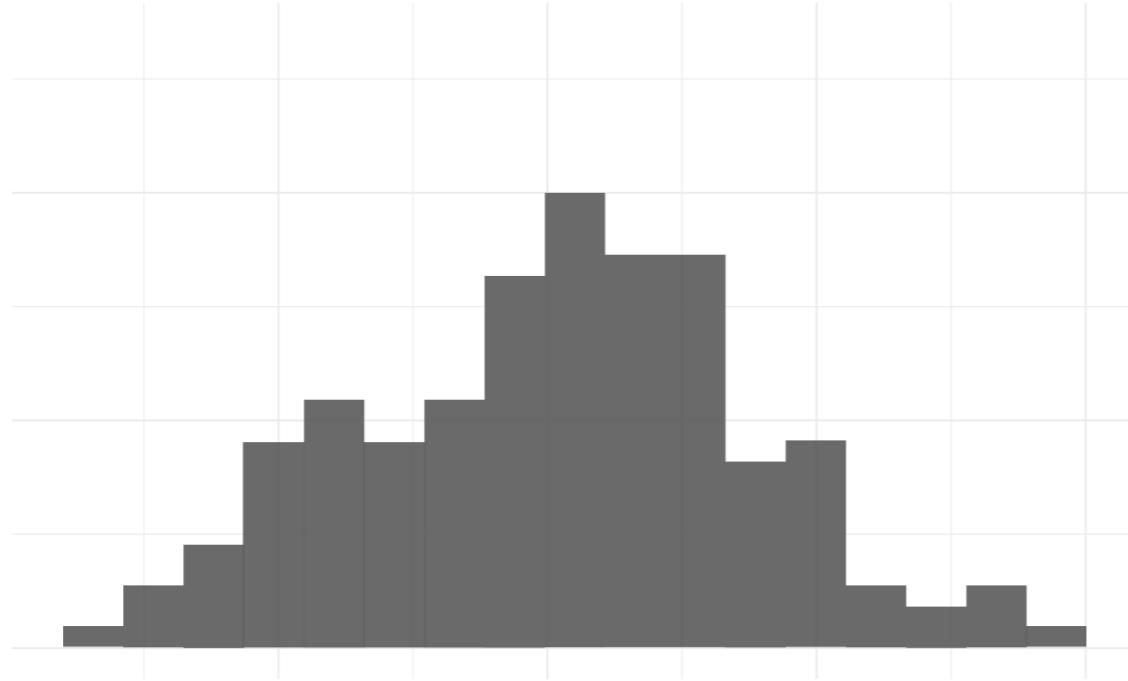
- Data collection or cleaning errors can become apparent
- Could indicate the need to control for a variable in a model
- Being true to the data



# Standard plots

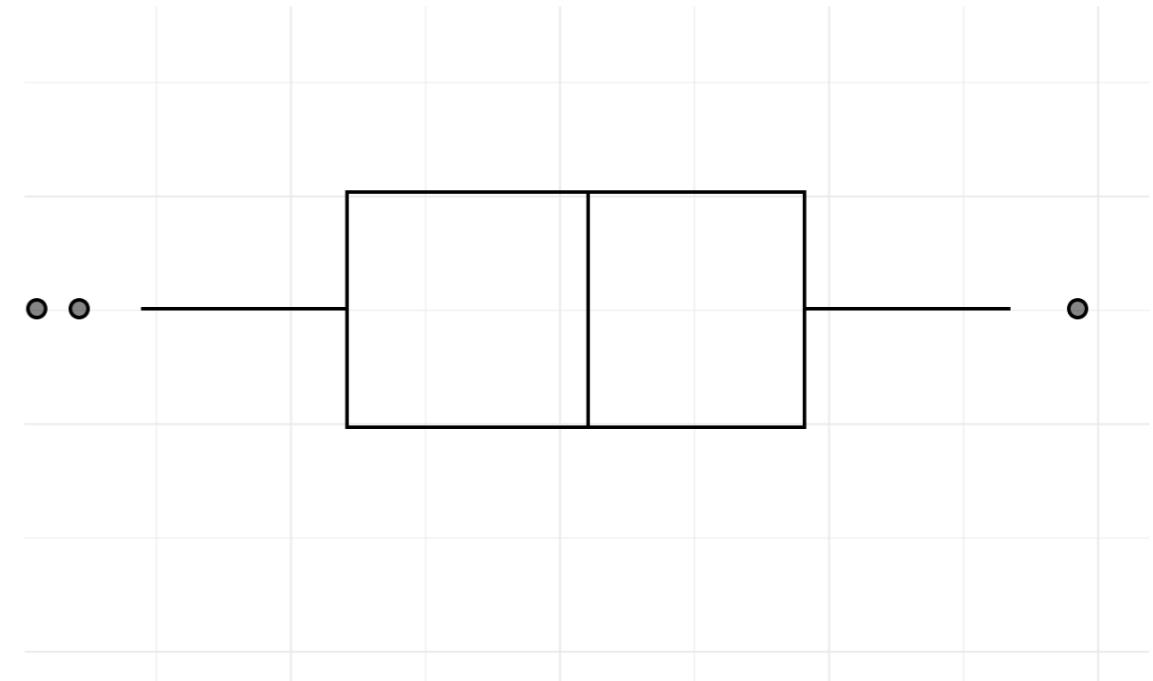
## Histogram

- Good for one distribution at a time
- This chapter



## boxplot

- For comparing multiple distributions
- Next chapter



# Maryland speeding data

- Speeding tickets given in Montgomery County, Maryland for 2017
- Retrieved from [data.montgomerycountymd.gov](http://data.montgomerycountymd.gov)

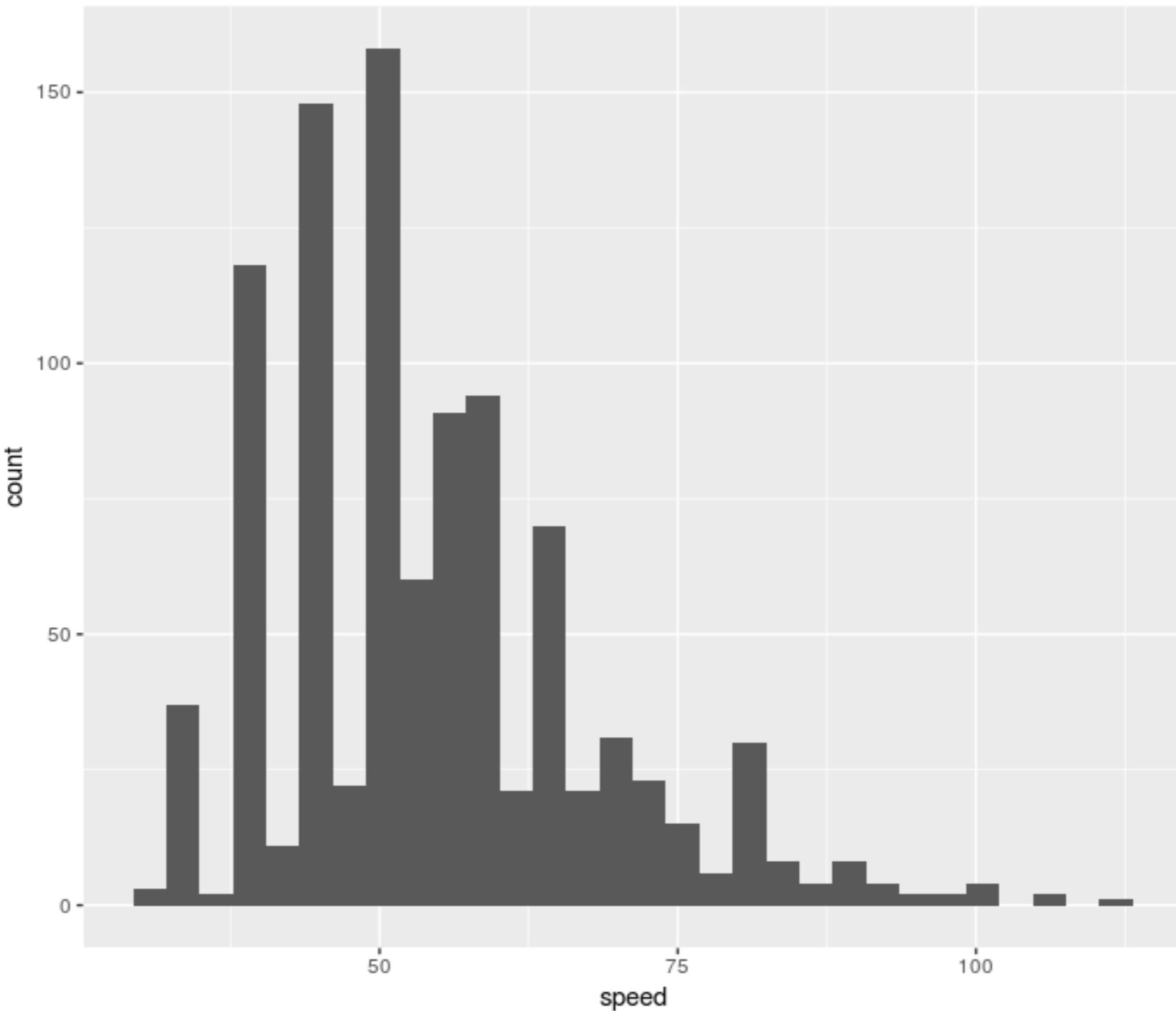
md\_speeding

```
# A tibble: 10,499 x 15
  work_zone vehicle_type vehicle_year vehicle_color race    gender driver_state speed_limit speed
  <lgl>     <chr>        <int>      <chr>       <chr>    <chr>   <chr>      <int>      <int>
1 F          Automobile    2003  BLUE    HISPANIC  F      MD            30        39
2 F          Automobile    2017  GREY    HISPANIC  M      MD            35        45
3 F          Automobile    2016  WHITE   WHITE     M      MD            35        50
4 F          Automobile    2006  RED     HISPANIC  M      MD            35        60
5 F          Automobile    2013  GREY    OTHER     F      MD            40        49
6 F          Automobile    2017  RED     WHITE     M      MD            40        49
7 F          Automobile    2003  GREY    BLACK     M      MD            40        49
8 F          Automobile    2004  GREY    OTHER     M      MD            40        49
9 F          Automobile    2000  WHITE   ASIAN     M      MD            55        90
10 F         Automobile    2007  BLACK   BLACK     F      MD            35        59
# ... with 10,489 more rows, and 6 more variables: day_of_week <chr>, day_of_month <int>,
#   month <chr>, hour_of_day <dbl>, speed_over <int>, percentage_over_limit <dbl>
```

# Making a histogram in ggplot2

- `geom_histogram()`
- Automatically bins data for you
- Just supply `x` `aes` thetic

```
md_speeding %>%  
  filter(vehicle_color == 'BLUE') %>%  
  ggplot(aes(x = speed)) +  
  geom_histogram()
```



# **Let's make some histograms!**

**VISUALIZATION BEST PRACTICES IN R**

# Histogram nuances

VISUALIZATION BEST PRACTICES IN R

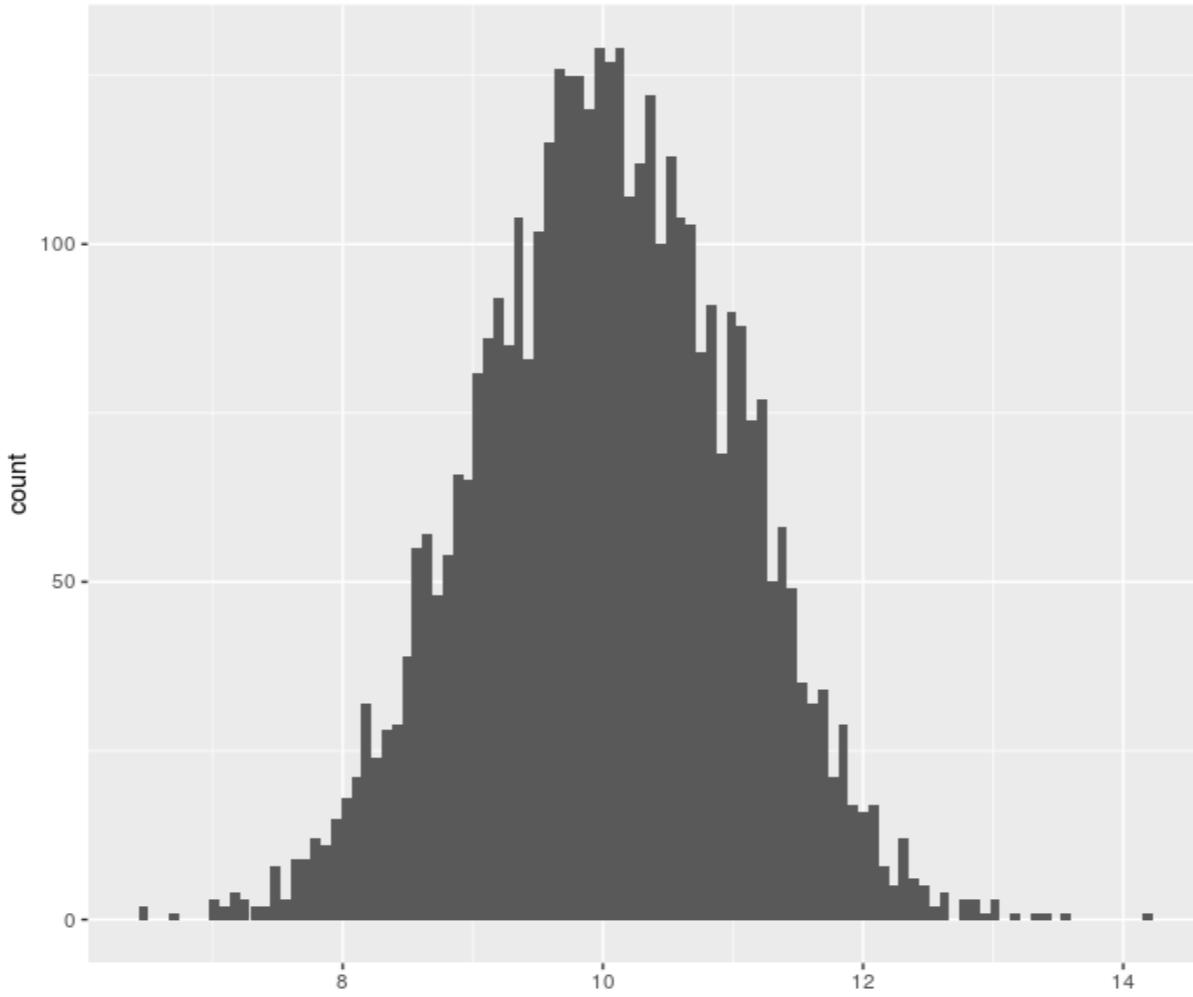


Nick Strayer

Instructor

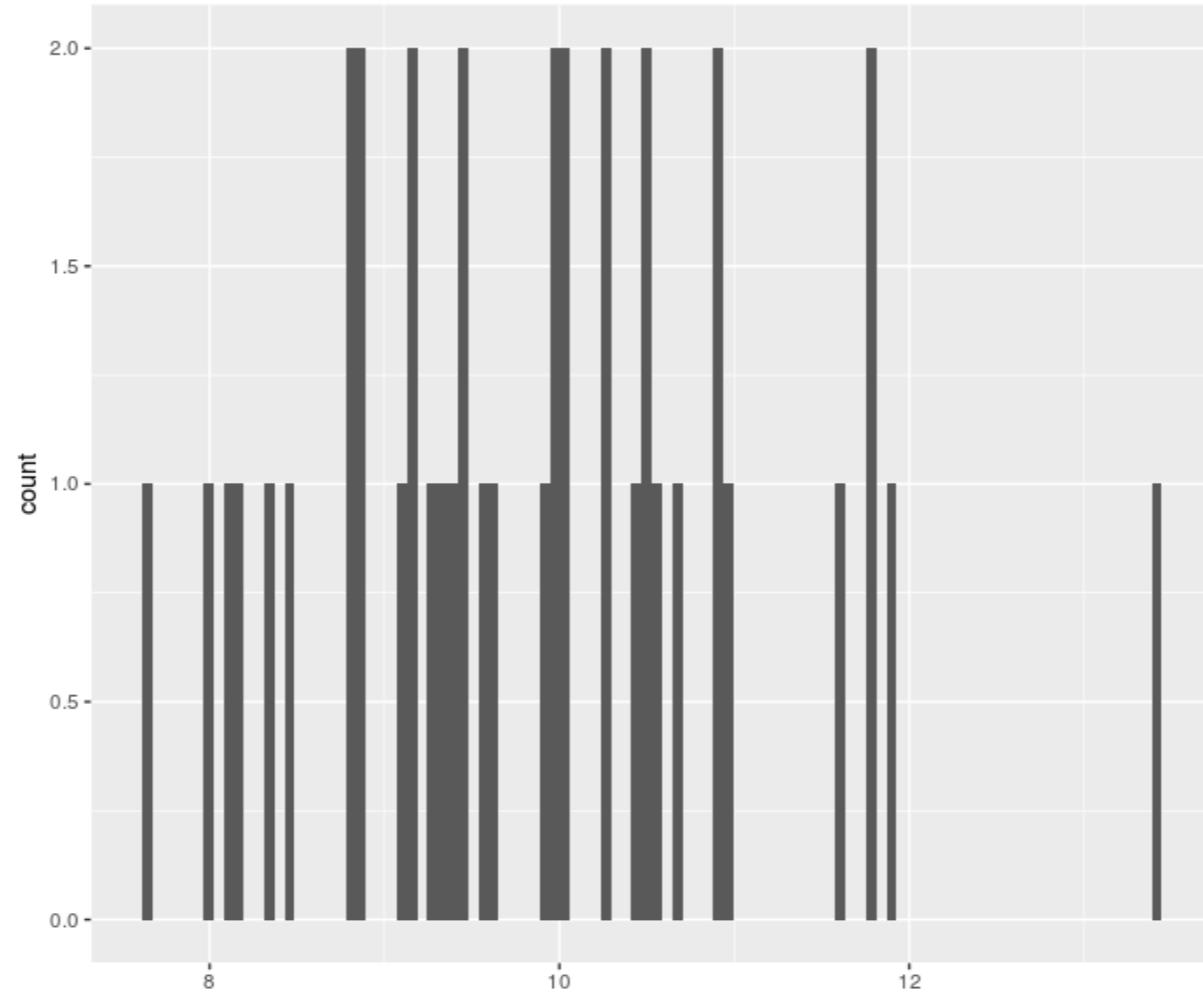
# Histogram positives

- Intuitive
- Interpretable



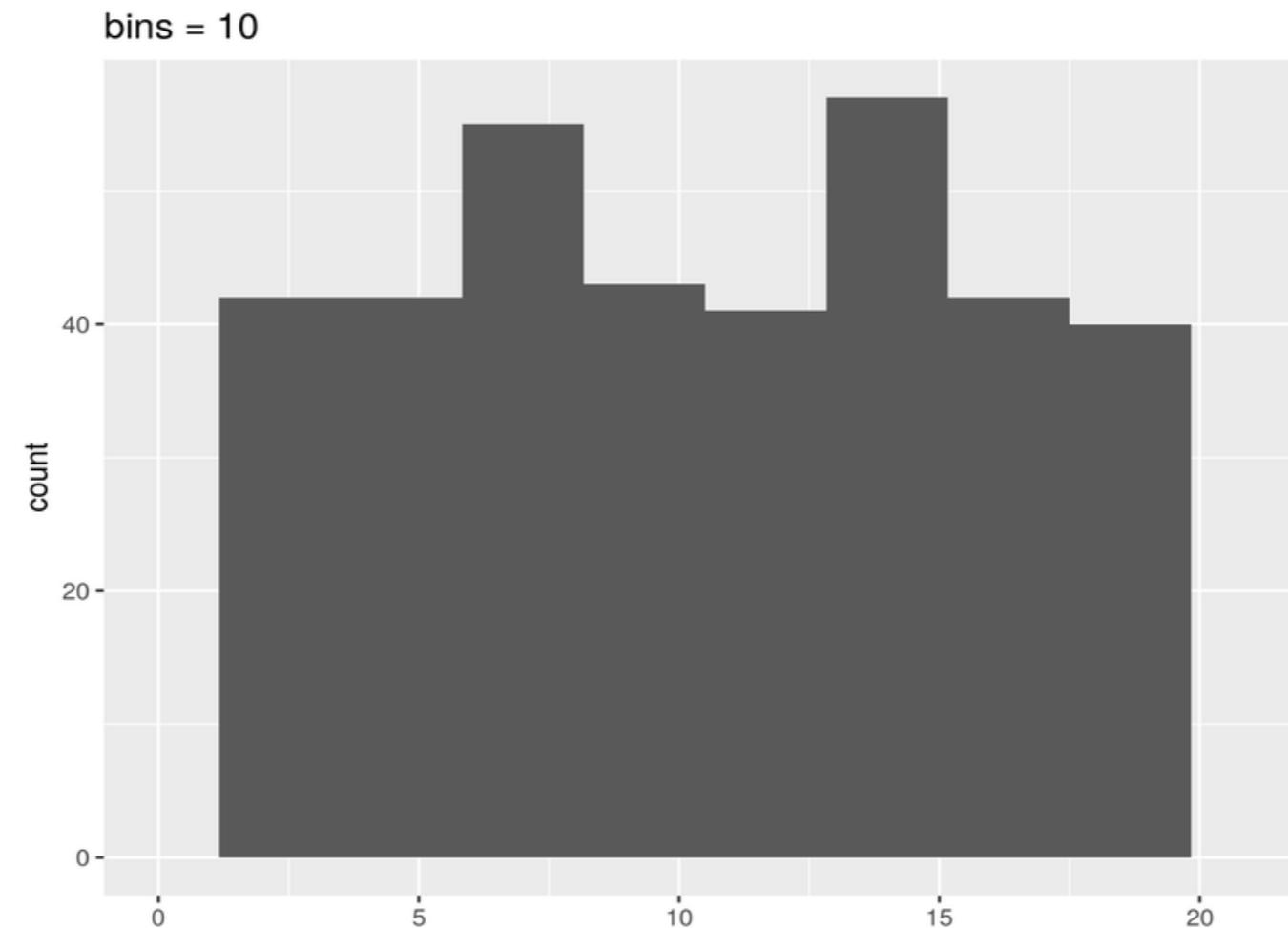
# Histogram negatives

- Sensitive to bin placements
- Iffy with small amounts of data



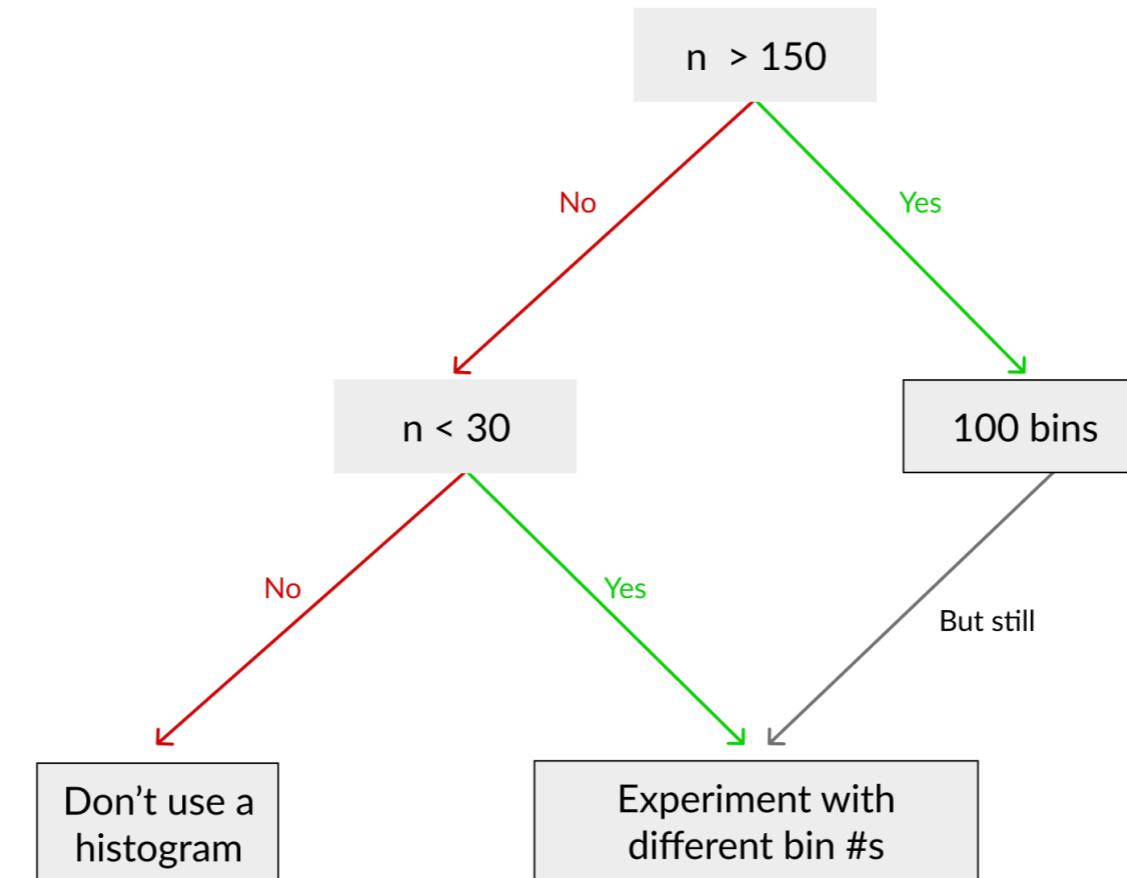
# Adjusting number of bins

- Exact same data
- Varying bin-numbers (`geom_histogram(bins = n)`) from 10 to 55



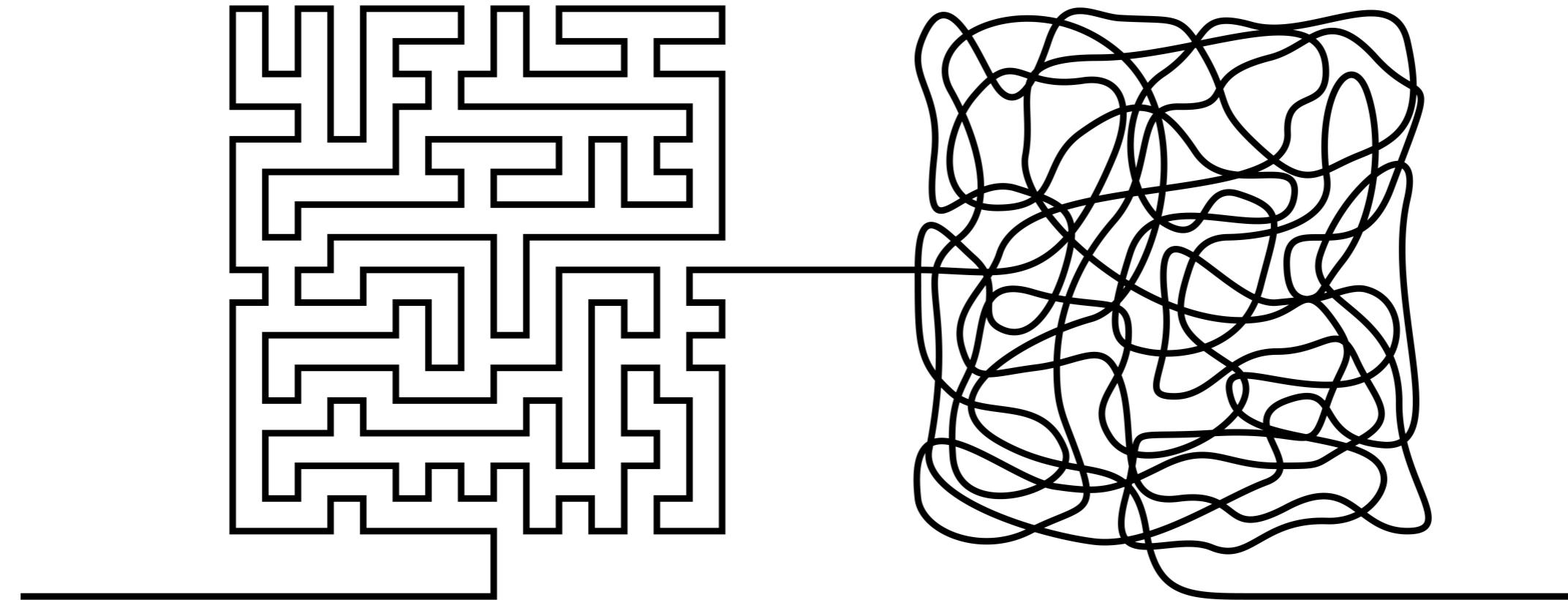
# Bin number best practices

- If `length(data$x) > 150` → `bins = 100`
- Otherwise, play around to get a good sense of the data



# Reality

- Beware of digit preferences
- Data from automated sources are less likely to be problematic



# **Let's improve some histograms!**

VISUALIZATION BEST PRACTICES IN R

# The kernel density estimator

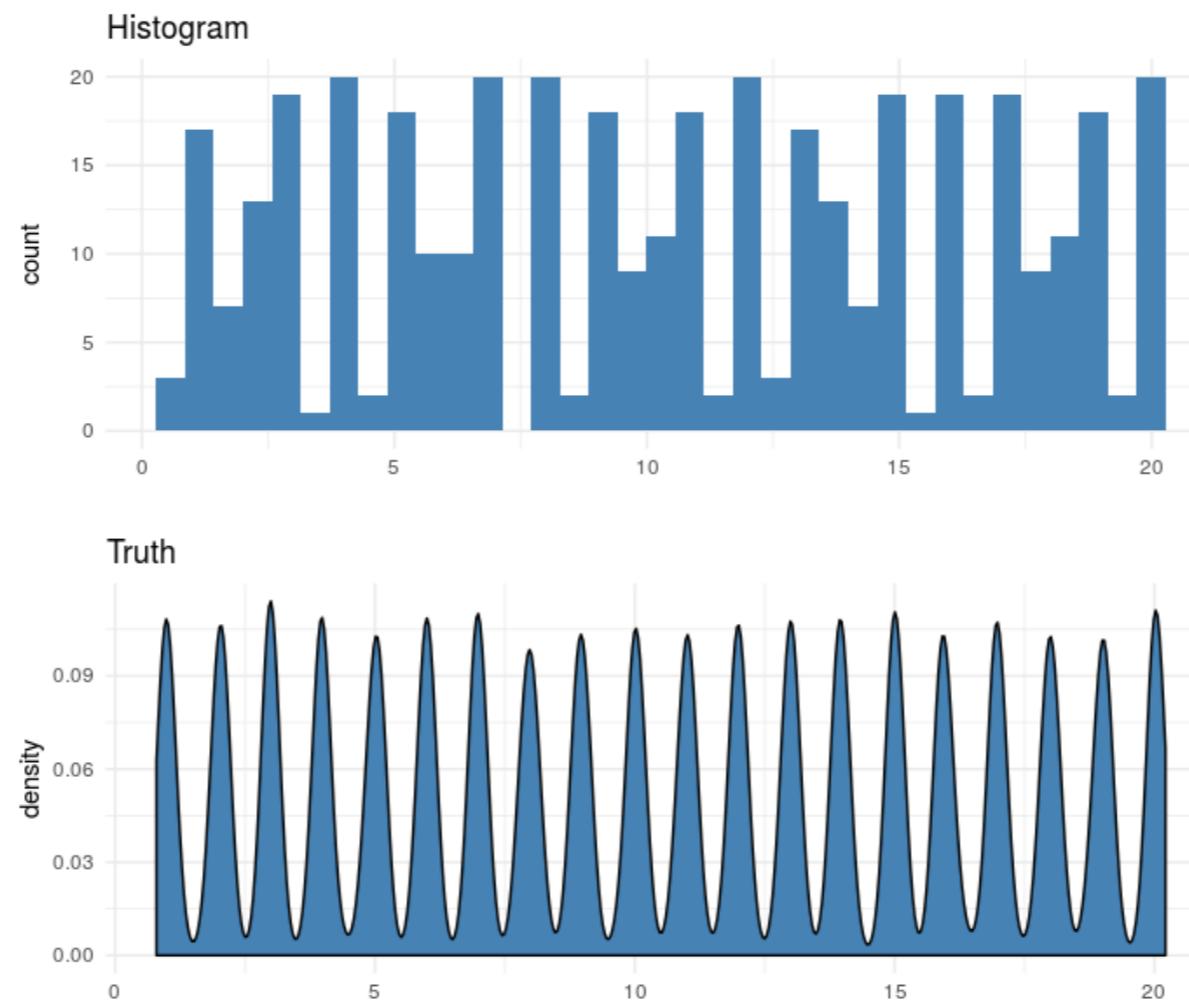
VISUALIZATION BEST PRACTICES IN R



**Nick Strayer**  
Instructor

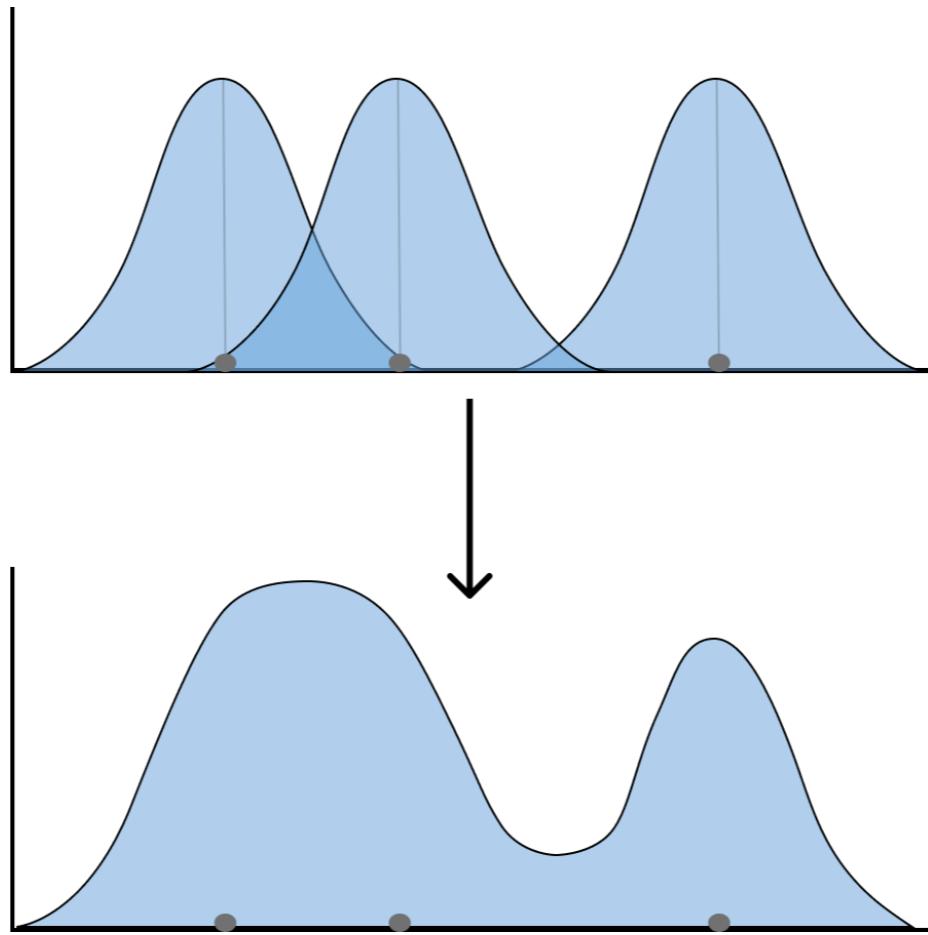
# Where histograms struggle

- Data with multiple strong peaks
- Small data



# Kernel density plots

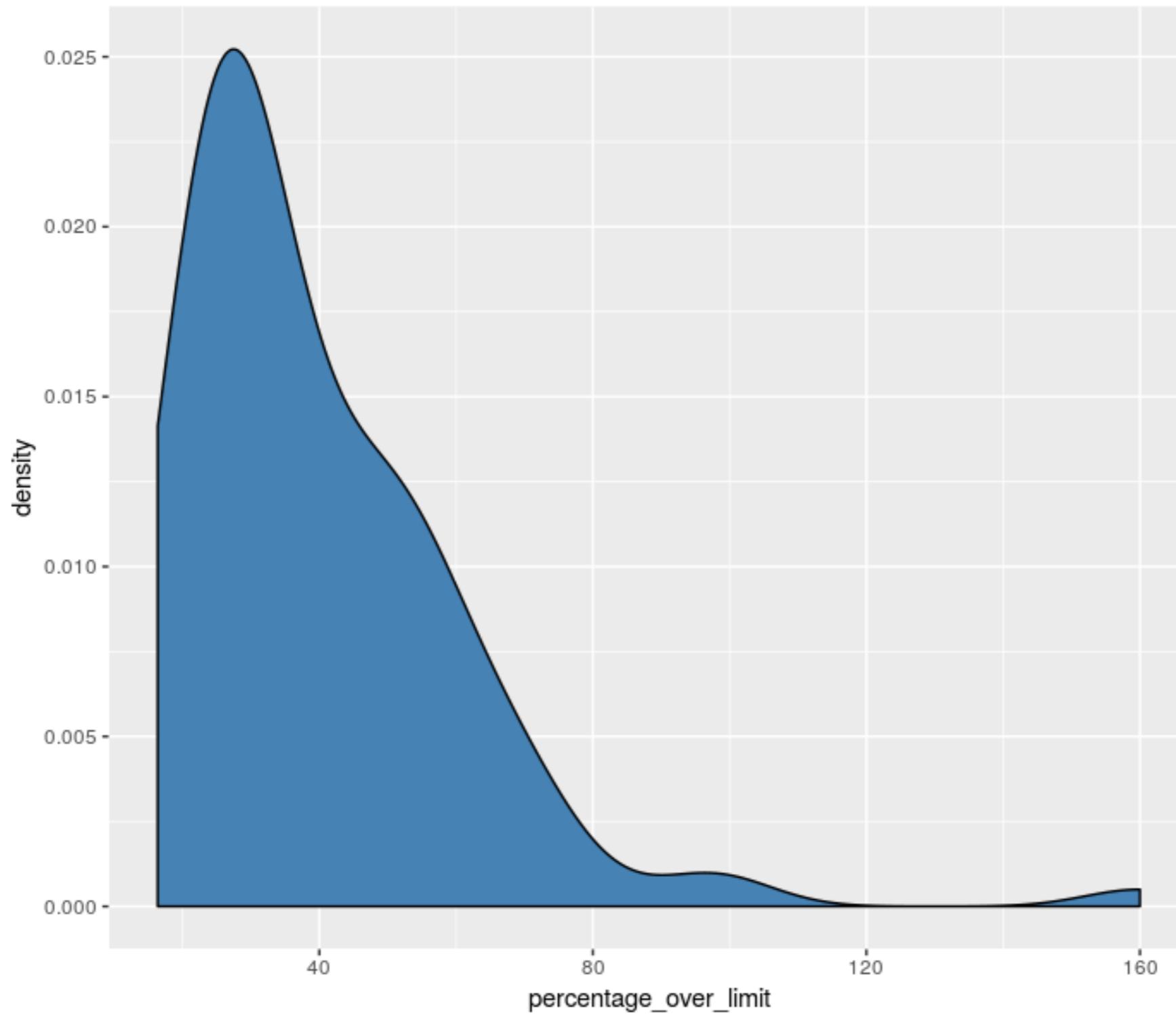
- Place "kernel" on top of every data point
- Add up heights of all overlapping kernels



# Making a KDE in ggplot

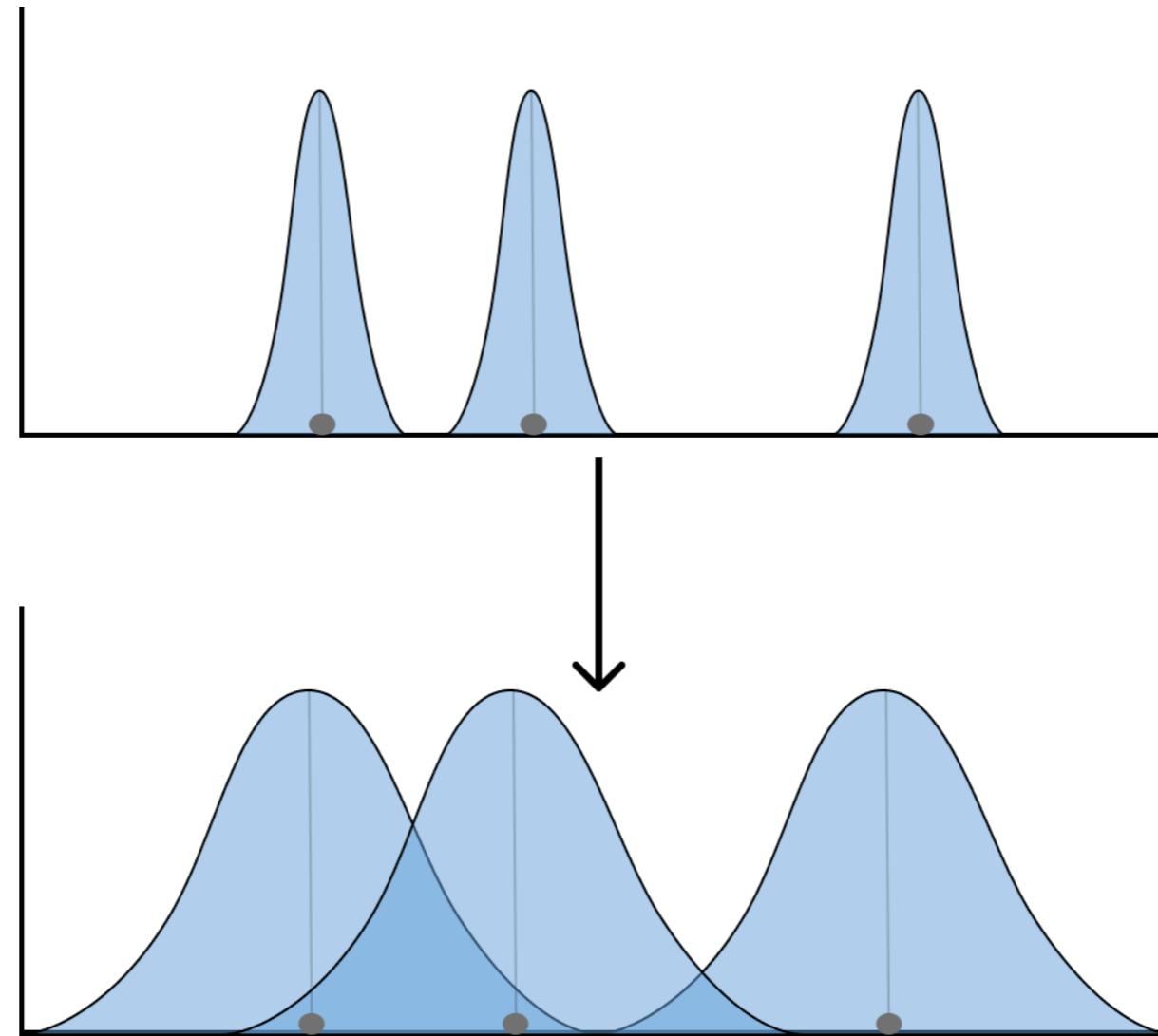
- Just swap `geom_histogram()` for `geom_density()`

```
sample_n(md_speeding, 100) %>%  
  ggplot(aes(x = percentage_over_limit)) +  
    # Swap out geom_histogram()  
    geom_density(  
      # Fill in curve with color  
      fill = 'steelblue',  
      # Standard deviation of kernel  
      bw = 8  
    )
```

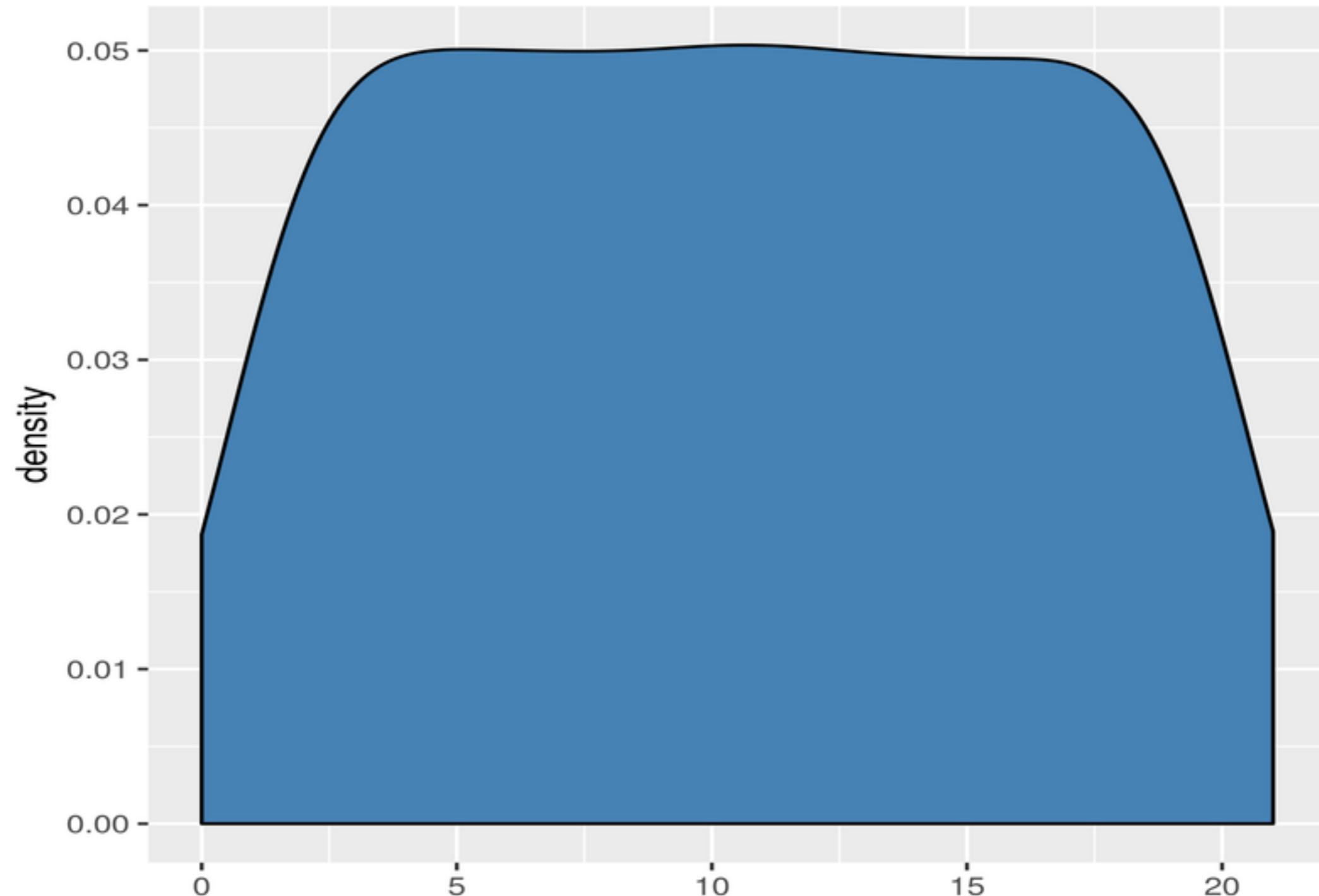


# A new width to worry about

- Need to adjust the standard deviation of the kernel placed on each point



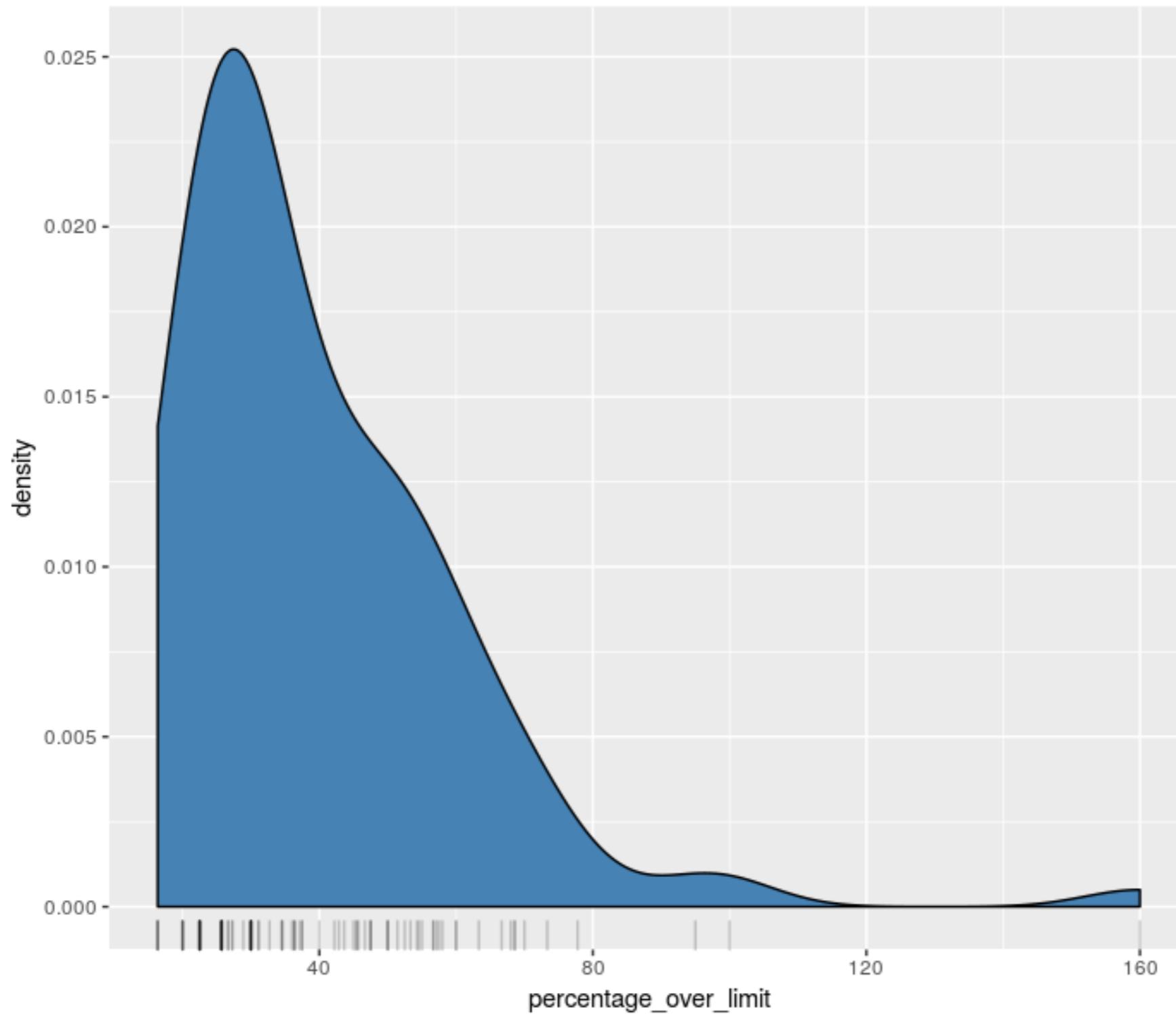
Size adjust = 100%



# Show all the data

Use `geom_rug()` to show all data below KDE with lines

```
p <- sample_n(md_speeding, 100) %>%
  ggplot(aes(x = percentage_over_limit)) +
  geom_density(
    fill = 'steelblue', # fill in curve with color
    bw = 8 # standard deviation of kernel
  )
p + geom_rug(alpha = 0.4)
```



# **Let's stack some gaussians!**

**VISUALIZATION BEST PRACTICES IN R**