

Data Visualization

Visualizing 1D categorical and continuous variables

June 8th, 2023

New dataset - 2021 MVP Shohei Ohtani's batted balls

Created dataset of batted balls by the American League MVP Shohei Ohtani in 2021 season using `baseballr`:

```
library(tidyverse)
ohtani_batted_balls <-
  read_csv("http://www.stat.cmu.edu/cmsac/sure/2022/materials/data/sports/xy_examples/ohtani_2021.csv")
head(ohtani_batted_balls)
```

```
## # A tibble: 6 × 7
##   pitch_type batted_ball_type hit_x hit_y exit_velocity launch_angle outcome
##   <chr>      <chr>          <dbl> <dbl>         <dbl>         <dbl> <chr>
## 1 FC        line_drive        89.7  144.         113.           20 home_run
## 2 CH        fly_ball          3.35  83.9         83.9           55 field_out
## 3 CH        fly_ball       -65.6  126.         102.           38 field_out
## 4 CU        ground_ball        39.2   50.4         82.5            8 field_out
## 5 FC        fly_ball       -37.6  138.         101.           23 field_out
## 6 KC        popup          -51.9   41.6         84             65 field_out
```

- each row / observation is a batted ball from Ohtani's 2021 season
- **Categorical** / qualitative variables: `pitch_type`, `batted_ball_type`, `outcome`
- **Continuous** / quantitative variables: `hit_x`, `hit_y`, `exit_velocity`, `launch_angle`

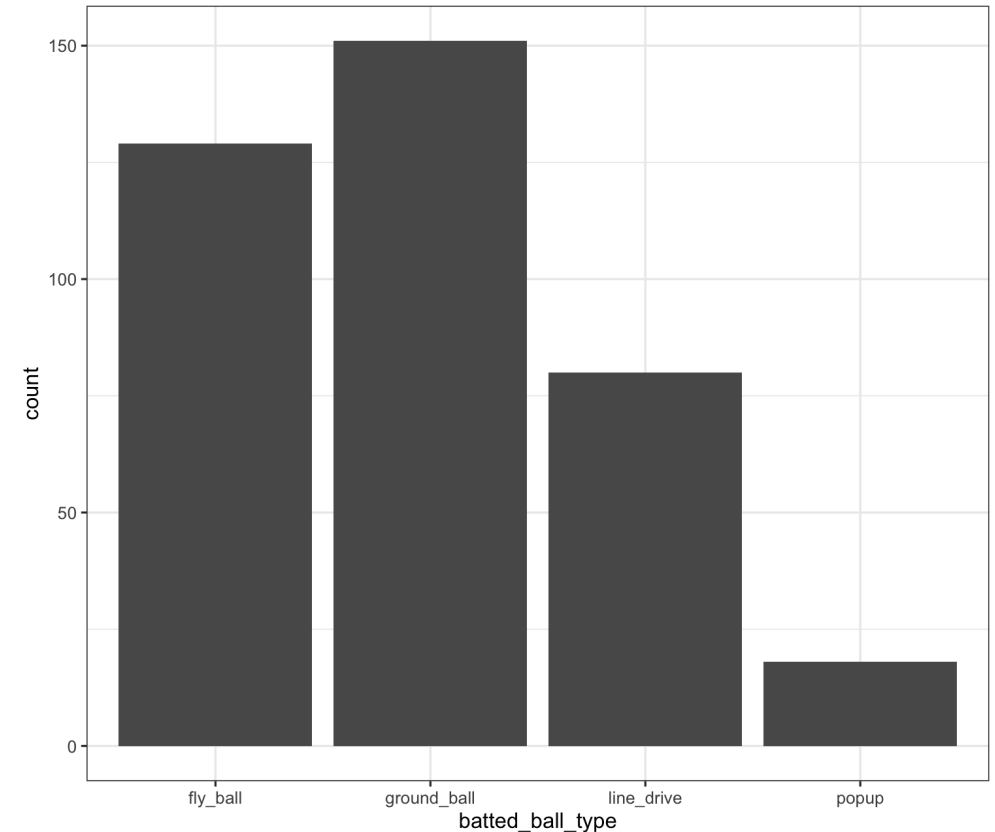
Visualizing 1D categorical data

How can we summarize `batted_ball_type` and other categorical variables?

- We make a **bar chart** with `geom_bar()`

```
ohtani_batted_balls %>%  
  ggplot(aes(x = batted_ball_type)) +  
  geom_bar() +  
  theme_bw()
```

- Only map `batted_ball_type` to the x-axis
- Counts of each type are displayed on y-axis...



Remember statistical summaries!

1. **geom_bar()** begins with the **diamonds** data set

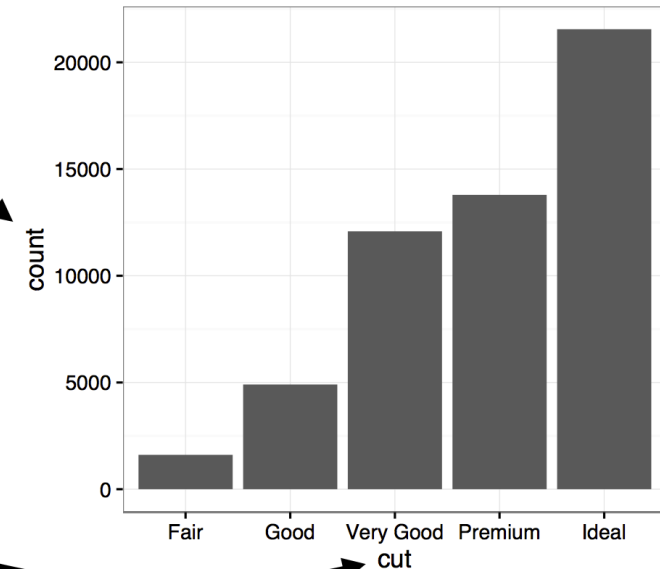
carat	cut	color	clarity	depth	table	price	x	y	z
0.23	Ideal	E	SI2	61.5	55	326	3.95	3.98	2.43
0.21	Premium	E	SI1	59.8	61	326	3.89	3.84	2.31
0.23	Good	E	VS1	56.9	65	327	4.05	4.07	2.31
0.29	Premium	I	VS2	62.4	58	334	4.20	4.23	2.63
0.31	Good	J	SI2	63.3	58	335	4.34	4.35	2.75
...

stat_count()

2. **geom_bar()** transforms the data with the "count" stat, which returns a data set of cut values and counts.

cut	count	prop
Fair	1610	1
Good	4906	1
Very Good	12082	1
Premium	13791	1
Ideal	21551	1

3. **geom_bar()** uses the transformed data to build the plot. cut is mapped to the x axis, count is mapped to the y axis.



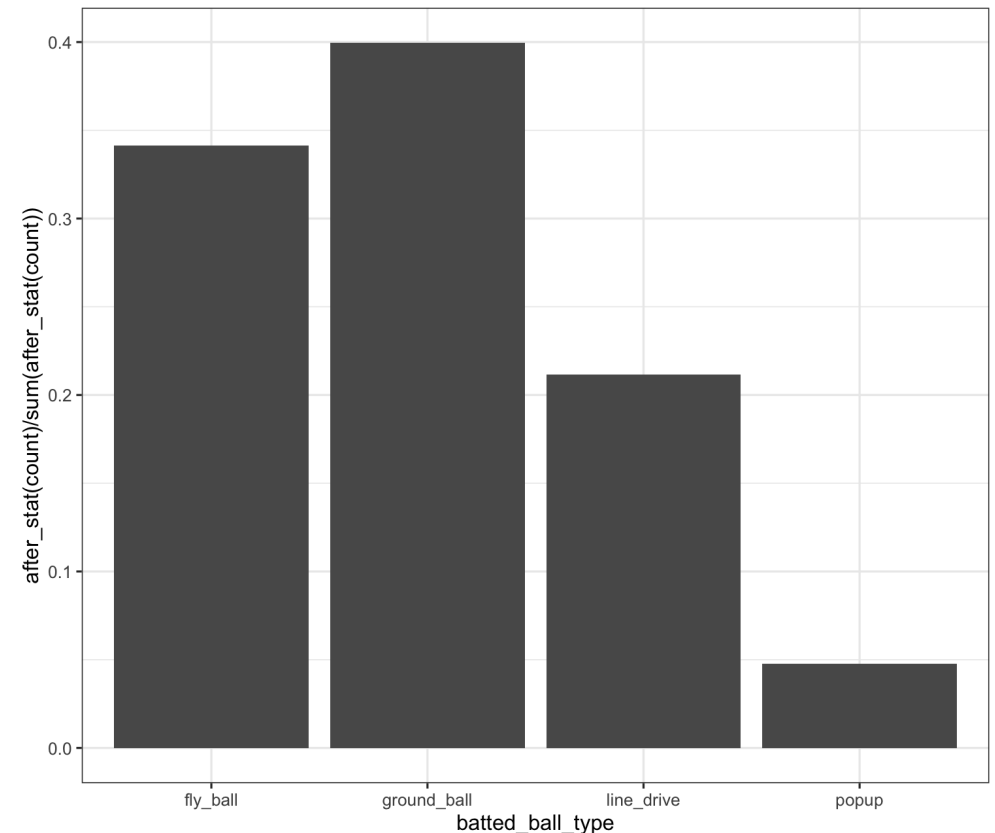
From Chapter 3 of R for Data Science

What does a bar chart show?

Marginal distribution: probability that categorical variable X (e.g., `batted_ball_type`) takes each particular value x (e.g. `fly_ball`). *So how do we display the individual probabilities?*

```
ohtani_batted_balls %>%  
  ggplot(aes(x = batted_ball_type)) +  
  geom_bar(aes(y = after_stat(count) / sum(after_stat(count)))  
  theme_bw()
```

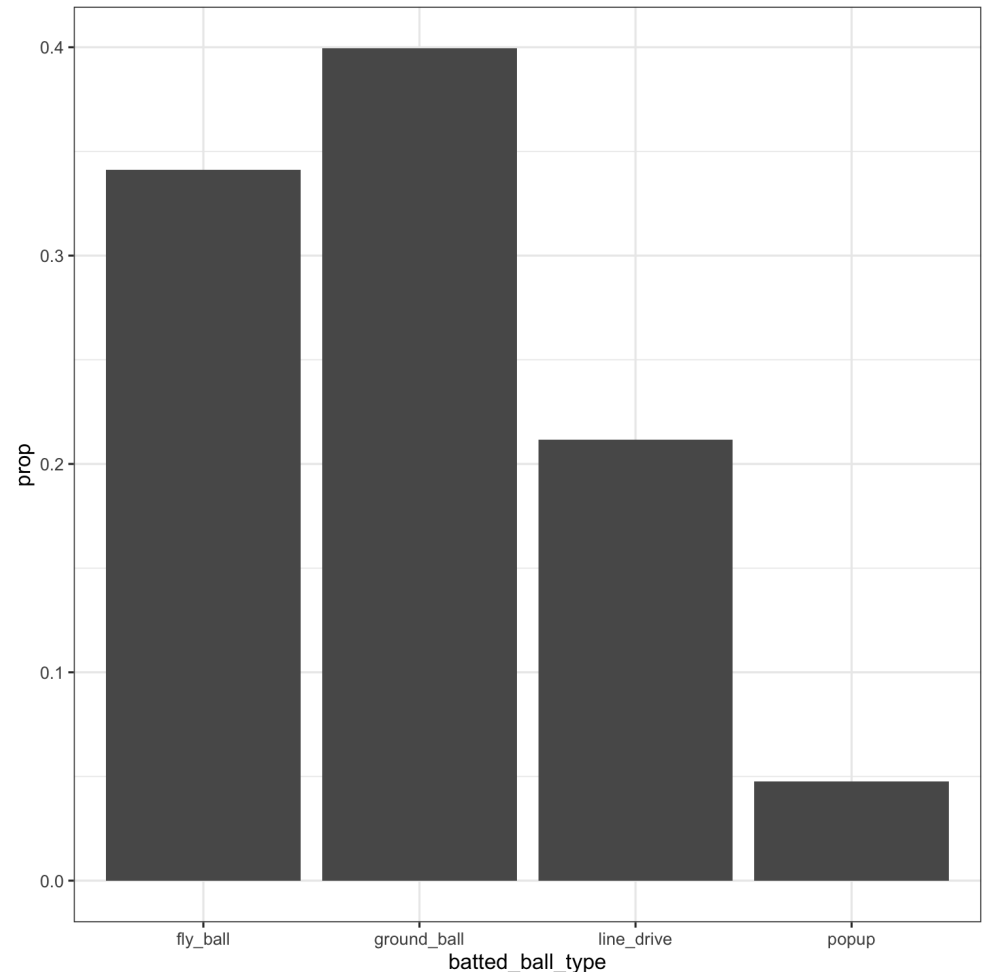
- `after_stat()` indicates the aesthetic mapping is performed after the statistical transformation
- Use `after_stat(count)` to access the `stat_count()` called by `geom_bar()`
- **We can code this in a more clear way**



Compute and display the proportions directly

```
ohtani_batted_balls %>%  
  group_by(batted_ball_type) %>%  
  summarize(count = n()) %>%  
  ungroup() %>%  
  mutate(total = sum(count),  
         prop = count / total) %>%  
  ggplot(aes(x = batted_ball_type)) +  
  geom_bar(aes(y = prop),  
          stat = "identity") +  
  theme_bw()
```

- Category counts give info about sample size, but this could be labeled in the chart
- Proportions = the **probability mass function** (PMF) for **discrete** variables
 - e.g. $P(\text{batted_ball_type} = \text{fly_ball})$



Population versus sample...

We have the **population** of Ohtani's batted balls in the 2021 season \Rightarrow **we know the true probabilities:**

- $P(\text{batted_ball_type} = \text{fly_ball})$
- $P(\text{batted_ball_type} = \text{ground_ball})$
- $P(\text{batted_ball_type} = \text{line_drive})$
- $P(\text{batted_ball_type} = \text{popup})$

What if we pretend this is a sample from all hypothetical Ohtani 2021 seasons?

Empirical distribution: We **estimate** the **true marginal** distribution with **observed (sample) data**

\Rightarrow Estimate $P(\text{batted_ball_type} = C_j)$ with \hat{p}_j for each category C_j (e.g. $\hat{p}_{\text{fly_ball}}$)

Compute **standard error** for each \hat{p}_j :

$$SE(\hat{p}_j) = \sqrt{\frac{\hat{p}_j(1 - \hat{p}_j)}{n}}$$

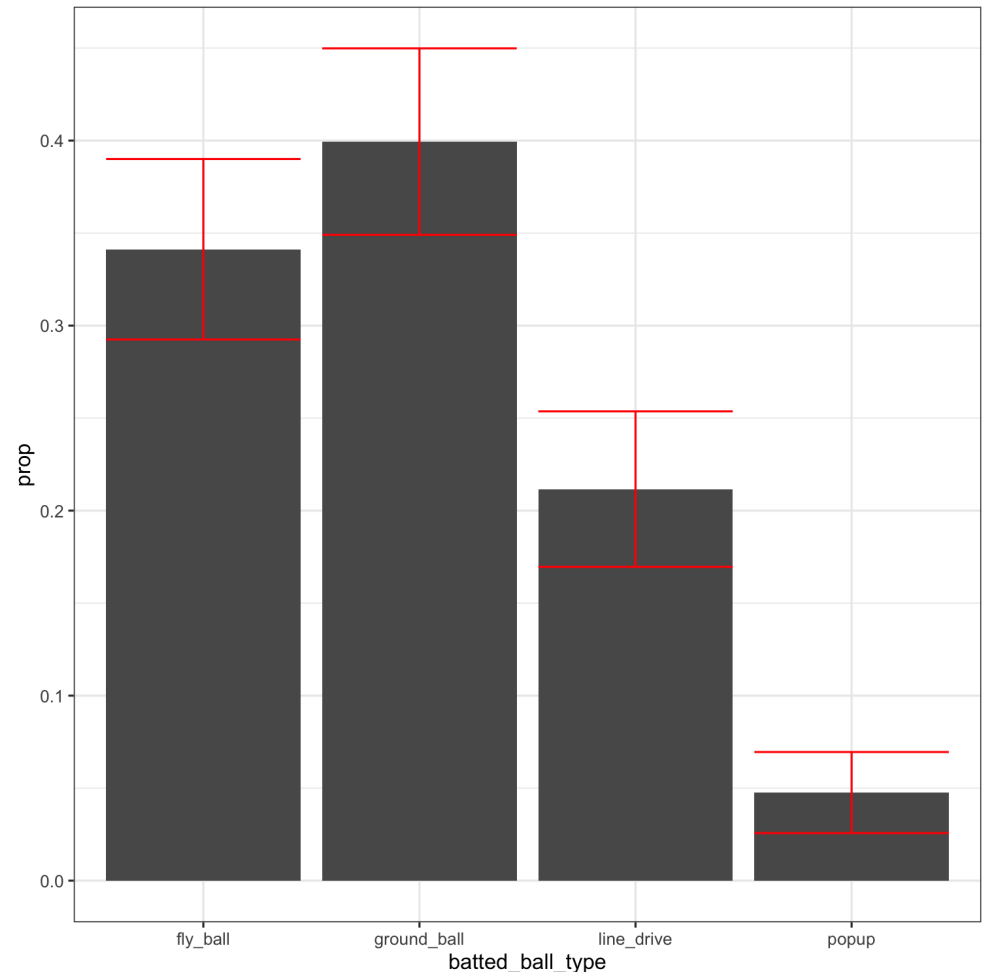
For large $n \Rightarrow \approx 95\%$ **confidence interval (CI):** $\hat{p}_j + / - 2 \cdot SE(\hat{p}_j)$

Add confidence intervals to bar chart

```
ohtani_batted_balls %>%
  group_by(batted_ball_type) %>%
  summarize(count = n()) %>%
  ungroup() %>%
  mutate(total = sum(count),
         prop = count / total,
         se = sqrt(prop * (1 - prop) / total),
         lower = prop - 2 * se,
         upper = prop + 2 * se) %>%
  ggplot(aes(x = batted_ball_type)) +
  geom_bar(aes(y = prop),
          stat = "identity") +
  geom_errorbar(aes(ymin = lower,
                   ymax = upper),
               color = "red") +
  theme_bw()
```

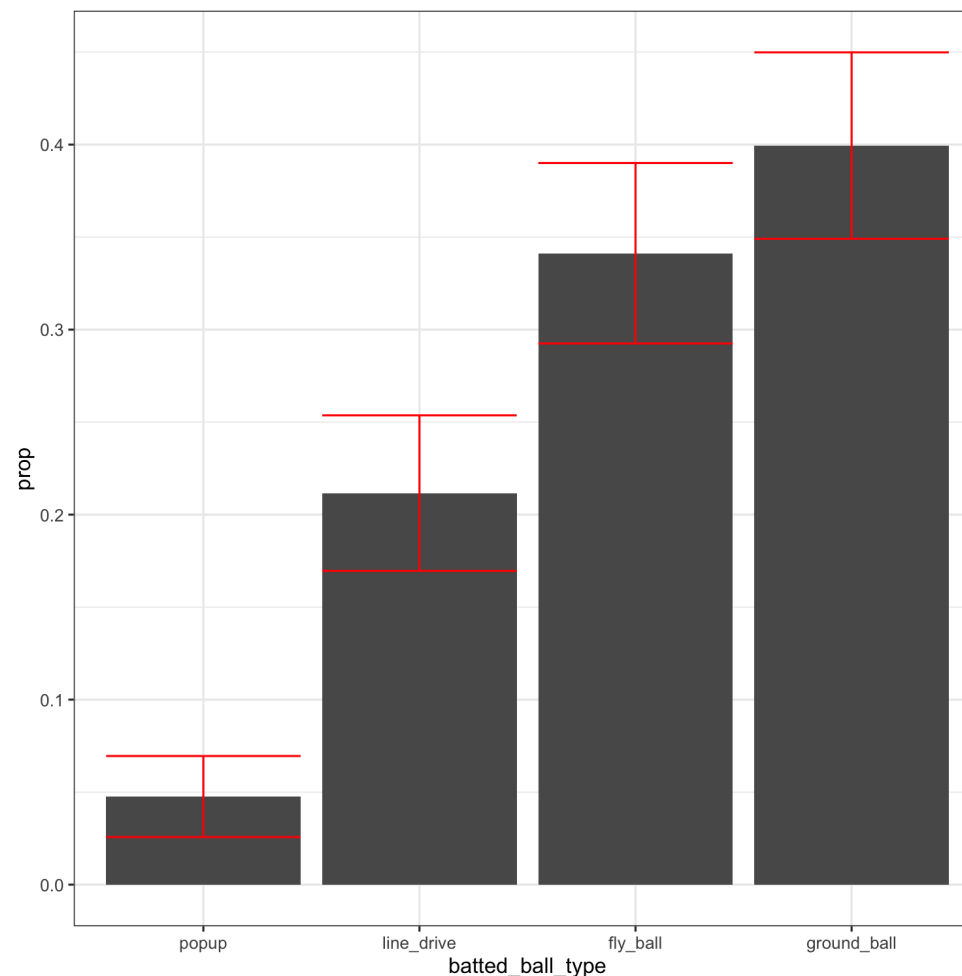
Be careful about your interpretation of CIs...

You should remember to label your charts!



Fun with factors using **forcats**

```
ohtani_batted_balls %>%
  group_by(batted_ball_type) %>%
  summarize(count = n()) %>%
  ungroup() %>%
  mutate(total = sum(count),
         prop = count / total,
         se = sqrt(prop * (1 - prop) / total),
         lower = prop - 2 * se,
         upper = prop + 2 * se,
         batted_ball_type =
           fct_reorder(batted_ball_type,
                       prop)) %>%
  ggplot(aes(x = batted_ball_type)) +
  geom_bar(aes(y = prop),
          stat = "identity") +
  geom_errorbar(aes(ymin = lower,
                   ymax = upper),
               color = "red") +
  theme_bw()
```



Did you say pie chart?



This is the only pie chart I will show you all summer

(Note: These slides originally come from Professor Yurko, a known hater of pie charts)

Describing 1D continuous data

How can we summarize `exit_velocity` and other continuous variables?

- **Center:** mean, median, number and location of modes
- **Spread:** range (max - min), quantiles, variance (standard deviation), etc.
- **Shape:** skew vs symmetry, outliers, heavy vs light tails, etc.
- Compute basic summary statistics

```
summary(ohtani_batted_balls$exit_velocity)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   NA's  
##    27.50   83.75   96.00   93.26  105.55  119.00     27
```

```
sd(ohtani_batted_balls$exit_velocity)
```

```
## [1] NA
```

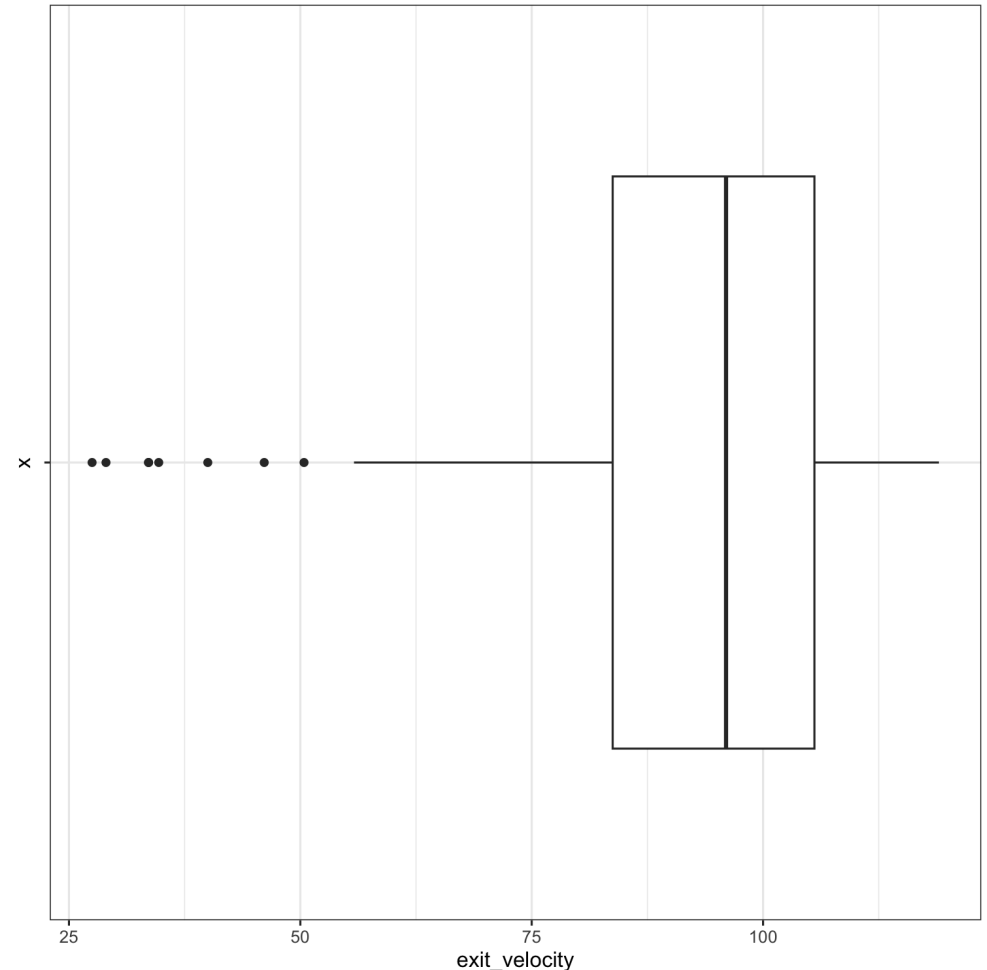
Box plots visualize summary statistics

- We make a **box plot** with `geom_boxplot()`

```
ohtani_batted_balls %>%  
  ggplot(aes(y = exit_velocity)) +  
  geom_boxplot(aes(x = "")) +  
  theme_bw() +  
  coord_flip()
```

- **Pros:**
 - Displays outliers, percentiles, spread, skew
 - Useful for side-by-side comparison (tomorrow)
- **Cons:**
 - Does not display the full distribution shape!
 - Does not display modes

Why use `aes(x = "")` inside `geom_boxplot()`?



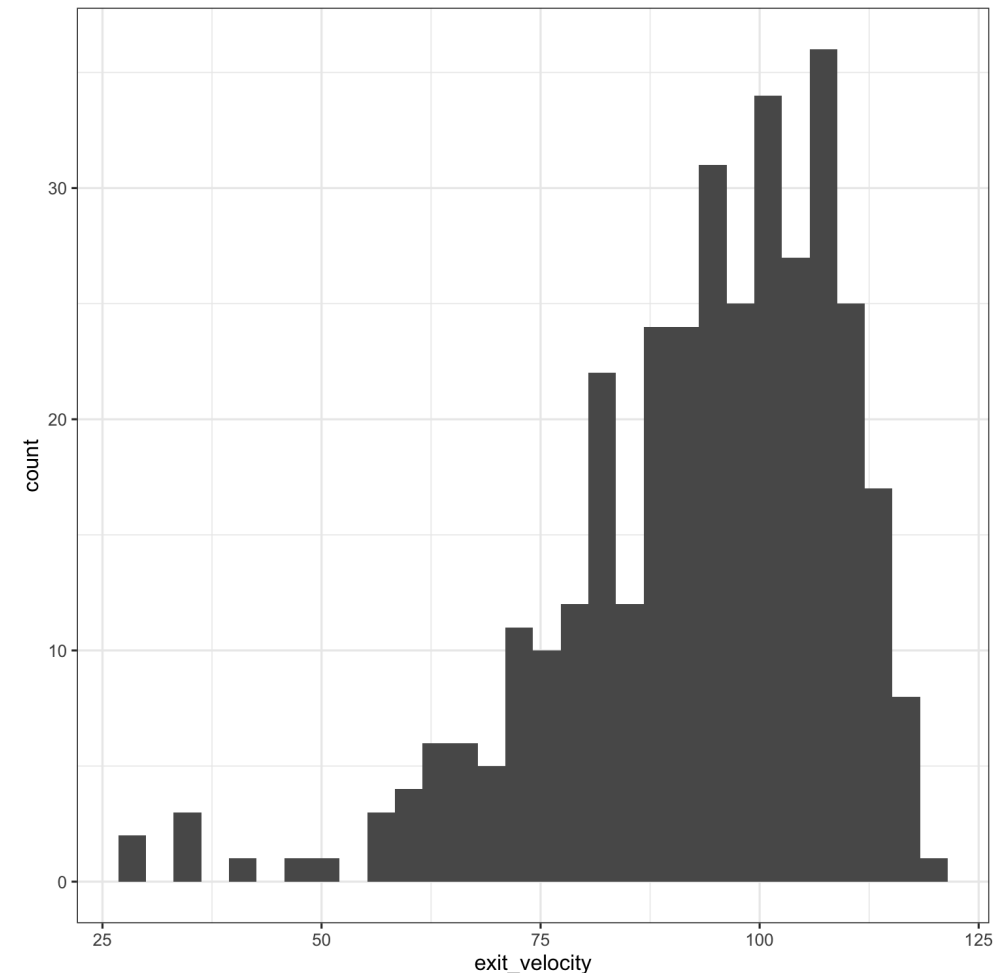
Histograms display 1D continuous distributions

- We make **histograms** with `geom_histogram()`

```
ohtani_batted_balls %>%  
  ggplot(aes(x = exit_velocity)) +  
  geom_histogram() +  
  theme_bw()
```

$$\# \text{ total obs.} = \sum_{j=1}^k \# \text{ obs. in bin } j$$

- **Pros:**
 - Displays full shape of distribution
 - Easy to interpret
- **Cons:**
 - Have to choose number of bins and bin locations (will revisit later)

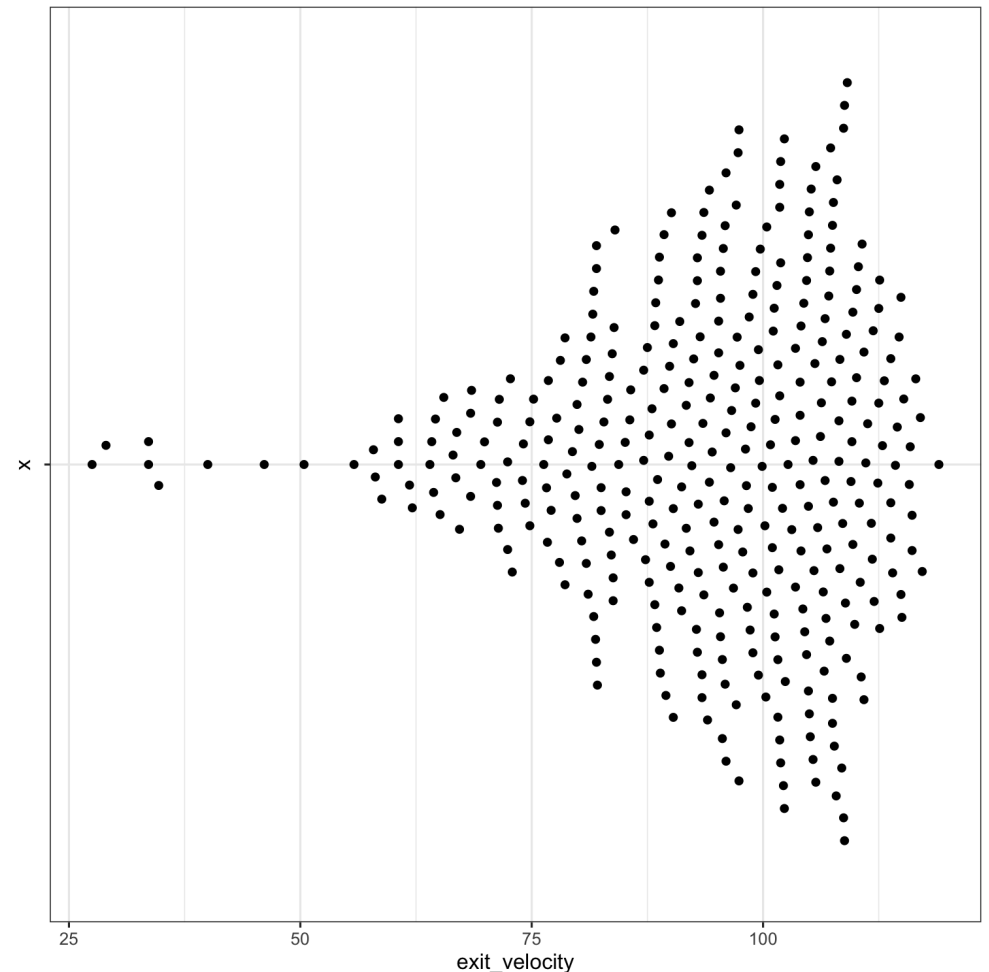


Display the data points directly with beeswarm plots

- We make a **beeswarm plot** using the **ggbeeswarm** package

```
library(ggbeeswarm)
ohtani_batted_balls %>%
  ggplot(aes(y = exit_velocity)) +
  geom_beeswarm(aes(x = "" ),
               cex = 3) +
  theme_bw() +
  coord_flip()
```

- **Pros:**
 - Displays each data point
 - Easy to view full shape of distribution
- **Cons:**
 - Can be overbearing with large datasets
 - Which algorithm for arranging points?



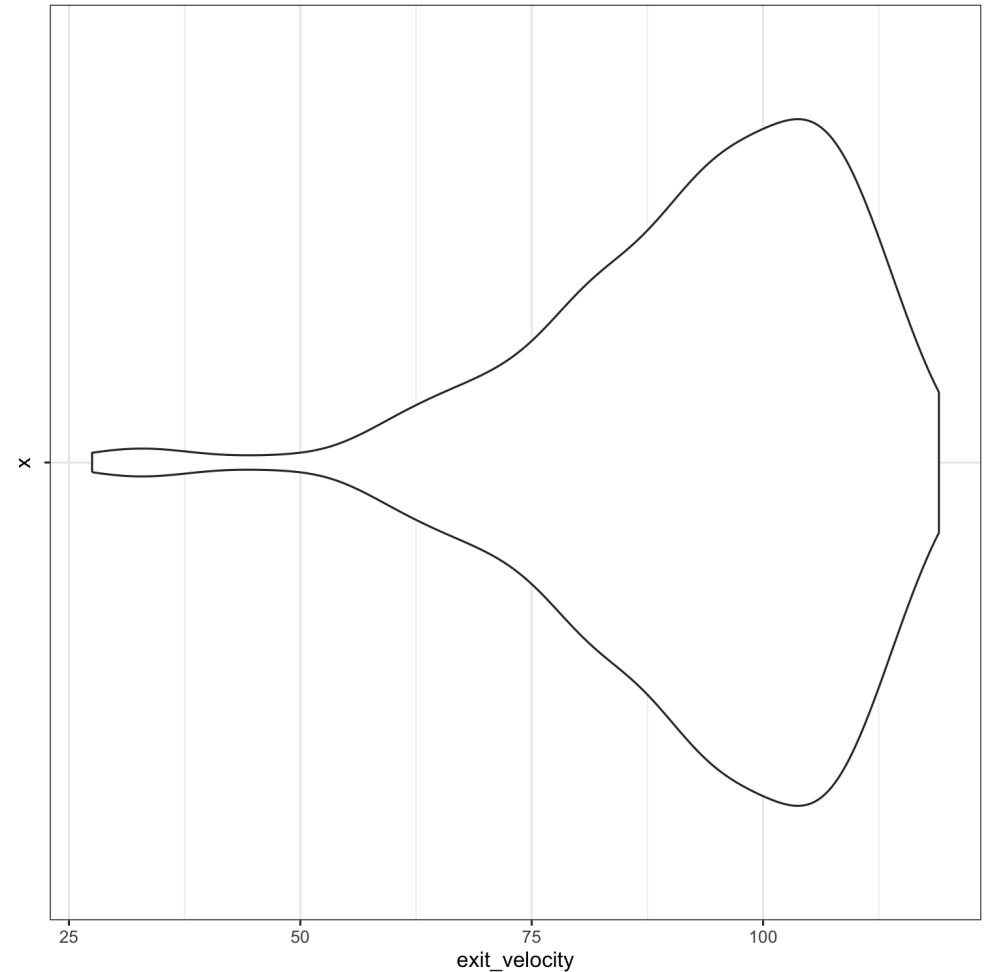
What does `cex = 3` do?

Smooth summary with violin plots

- We make **violin plots** with `geom_violin()`

```
ohtani_batted_balls %>%  
  ggplot(aes(y = exit_velocity)) +  
  geom_violin(aes(x = "")) +  
  theme_bw() +  
  coord_flip()
```

- **Pros:**
 - Displays full shape of distribution
 - Can easily layer...

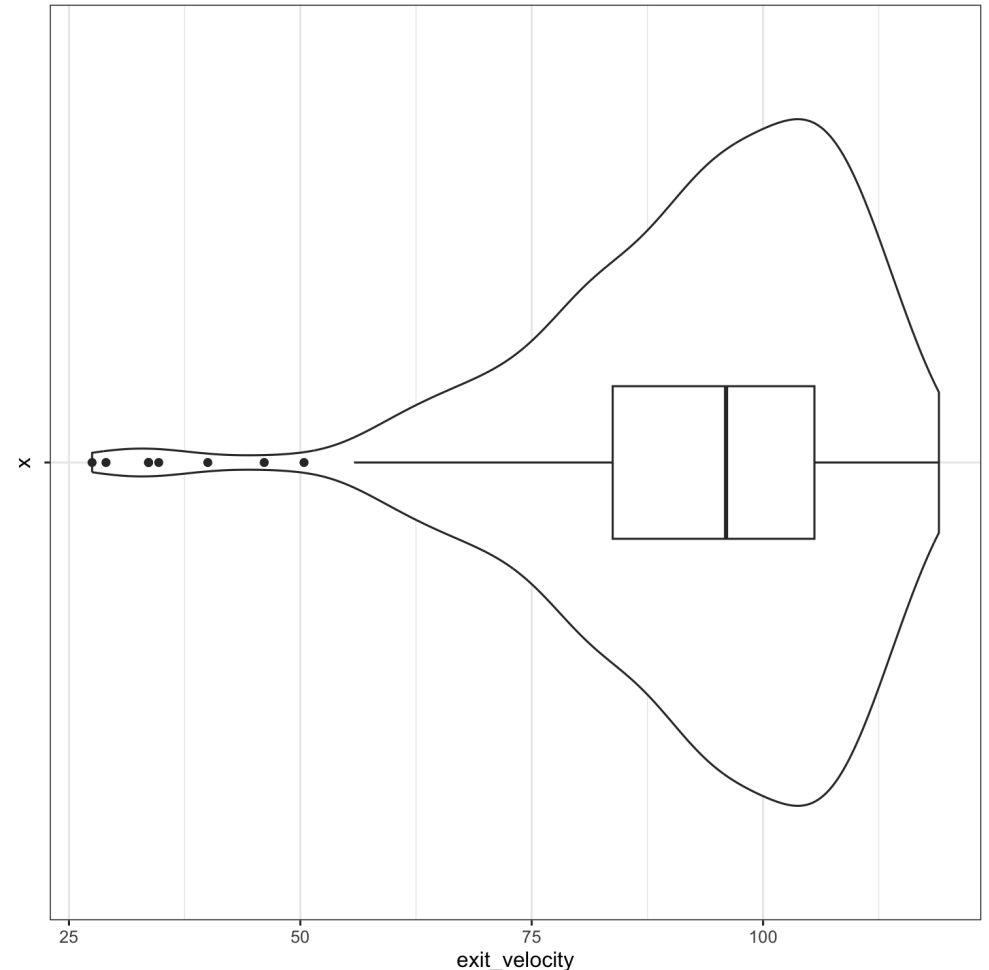


Smooth summary with violin plots + box plots

- We make **violin plots** with `geom_violin()`

```
ohtani_batted_balls %>%  
  ggplot(aes(y = exit_velocity,  
             x = "")) +  
  geom_violin() +  
  geom_boxplot(width = .2) +  
  theme_bw() +  
  coord_flip()
```

- **Pros:**
 - Displays full shape of distribution
 - Can easily layer... with box plots on top
- **Cons:**
 - Summary of data via **density estimate**
 - Mirror image is duplicate information



What do visualizations of continuous distributions display?

Probability that continuous variable X takes a particular value is 0

e.g. $P(\text{exit_velocity} = 100) = 0$, *why?*

Instead we use the **probability density function (PDF)** to provide a **relative likelihood**

- Density estimation is the focus of lecture next Monday

For continuous variables we can use the **cumulative distribution function (CDF)**,

$$F(x) = P(X \leq x)$$

For n observations we can easily compute the **Empirical CDF (ECDF)**:

$$\hat{F}_n(x) = \frac{\# \text{ obs. with variable } \leq x}{n} = \frac{1}{n} \sum_{i=1}^n 1(x_i \leq x)$$

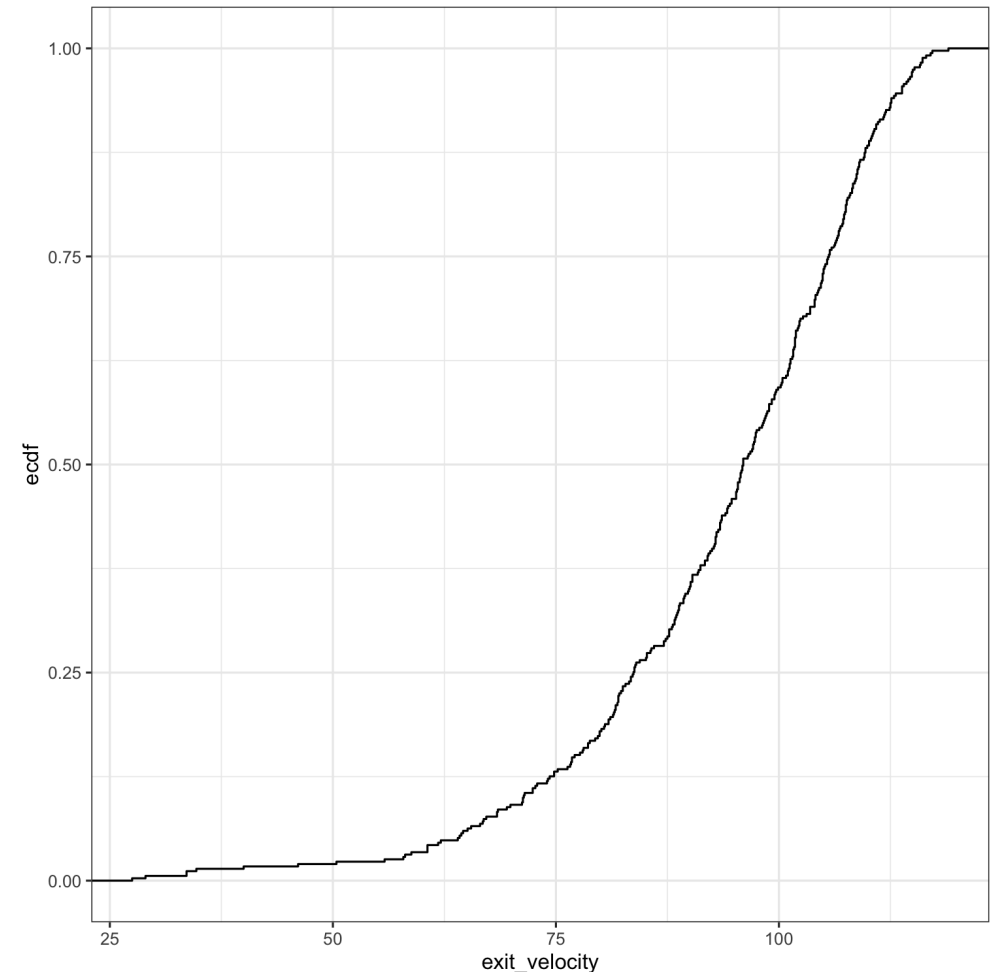
- where $1()$ is the indicator function, i.e. `ifelse(x_i <= x, 1, 0)`

Display full distribution with ECDF plot

- We make **ECDF plots** with `stat_ecdf()`

```
ohtani_batted_balls %>%  
  ggplot(aes(x = exit_velocity)) +  
  stat_ecdf() +  
  theme_bw()
```

- **Pros:**
 - ECDF displays all information in data (except for order)
 - As $n \rightarrow \infty$, our ECDF $\hat{F}_n(x)$ converges to the true CDF $F(x)$
 - Easy to interpret...
- **Cons:**
 - ... and yet it's not as popular!

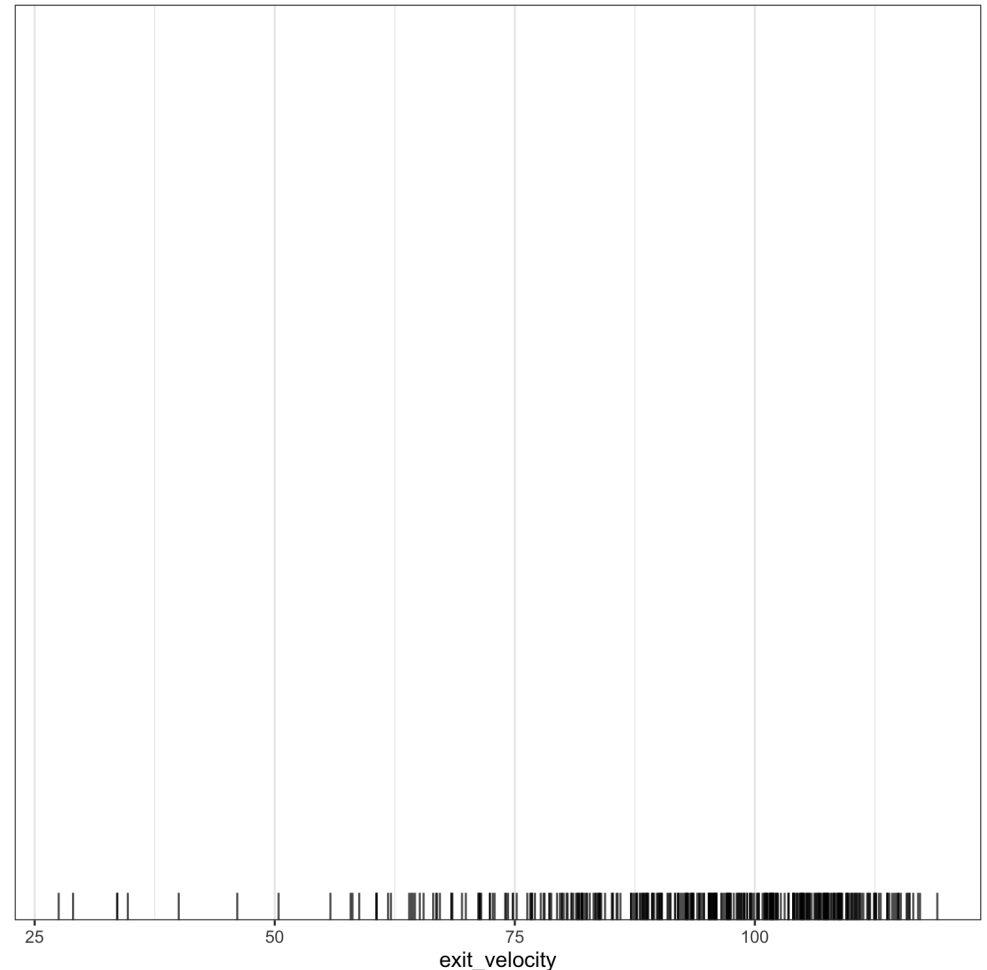


Rug plots display raw data

- We make a **rug plot** with `geom_rug()`

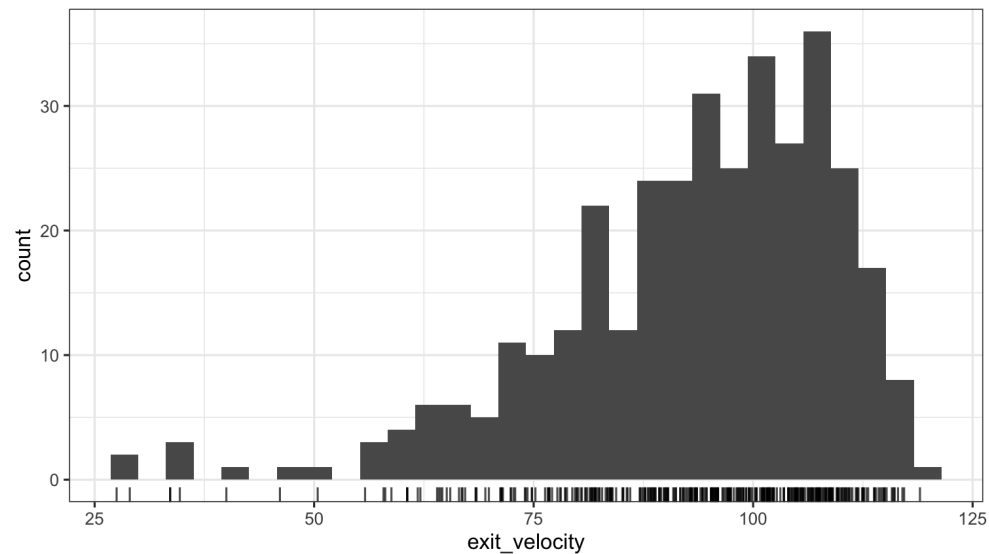
```
ohtani_batted_balls %>%  
  ggplot(aes(x = exit_velocity)) +  
  geom_rug(alpha = 0.7) +  
  theme_bw()
```

- **Pros:**
 - Displays raw data points
 - Useful supplement for summaries and 2D plots...
- **Cons:**
 - Can be overbearing for larger datasets

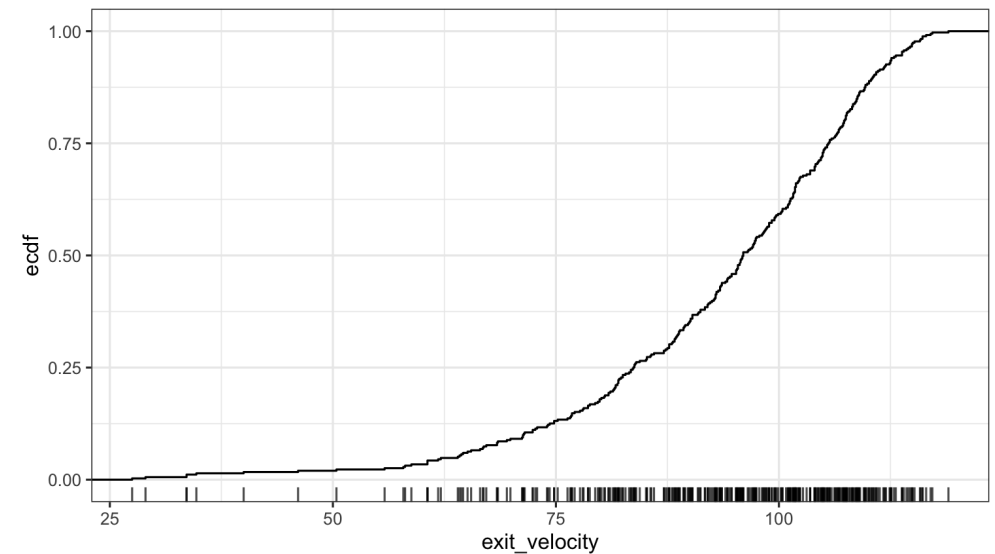


Rug plots supplement other displays

```
ohtani_batted_balls %>%  
  ggplot(aes(x = exit_velocity)) +  
  geom_rug(alpha = 0.7) +  
  geom_histogram() +  
  theme_bw()
```



```
ohtani_batted_balls %>%  
  ggplot(aes(x = exit_velocity)) +  
  geom_rug(alpha = 0.7) +  
  stat_ecdf() +  
  theme_bw()
```



Scatterplots for 2D continuous data

- We make a **scatterplot** with `geom_point()`

```
ohtani_batted_balls %>%  
  ggplot(aes(x = exit_velocity,  
             y = launch_angle)) +  
  geom_point() +  
  geom_rug(alpha = 0.4) +  
  theme_bw()
```

Easy to supplement with rug plots

Look at the plot: what question would you want to ask, assuming you know something about baseball?

To be continued...

