Data Visualization

Visualizing 2D categorical and continuous by categorical

June 10th, 2022

Revisiting MVP Shohei Ohtani's batted balls in 2021

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_202
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`

First - more fun with `forcats`

Variables of interest: `pitch_type` and `batted_ball_type` - but how many levels does `pitch_type` have?

```
table(ohtani_batted_balls$pitch_type)
```

```
##
## CH CU FC FF FS KC SI SL
## 62 37 30 87  8 11 57 62
```

We can manually `fct_recode` `pitch_type` (see Chapter 15 of R for Data Science for more on factors)

```
ohtani_batted_balls <- ohtani_batted_balls %>%
  filter(pitch_type != "null") %>%
  mutate(pitch_type = fct_recode(pitch_type, "Changeup" = "CH", "Breaking ball" = "CU",
                    "Fastball" = "FC", "Fastball" = "FF", "Fastball" = "FS",
                    "Breaking ball" = "KC",  "Fastball" = "SI",  "Breaking ball" = "SL"))
table(ohtani_batted_balls$pitch_type)
```

```
##
##      Changeup Breaking ball      Fastball
##            62           110           182
```

Inference for categorical data

The main test used for categorical data is the **chi-square test**:

- **Null hypothesis**: $H_0 : p_1 = p_2 = \cdots = p_K$ and we compute the **test statistic**:

$$\chi^2 = \sum_{j=1}^{K} \frac{(O_j - E_j)^2}{E_j}$$

- O_j: observed counts in category j
- E_j: expected counts under H_0 (i.e., $\frac{n}{K}$ or each category is equally likely to occur)

```
chisq.test(table(ohtani_batted_balls$pitch_type))
```

```
##
##     Chi-squared test for given probabilities
##
## data:  table(ohtani_batted_balls$pitch_type)
## X-squared = 61.831, df = 2, p-value = 3.747e-14
```

Statistical inference in general

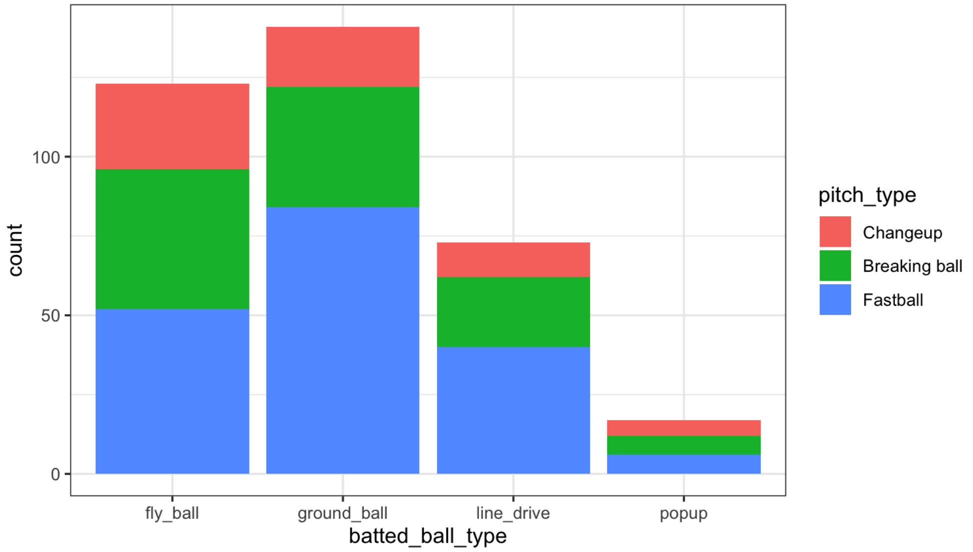
Computing p-values works like this:

- Choose a test statistic.
- Compute the test statistic in your dataset.
- Is test statistic "unusual" compared to what I would expect under H_0?
- Compare p-value to **target error rate** α (typically referred to as target level α)
- Typically choose $\alpha = 0.05$

2D Categorical visualization (== more bar charts!)

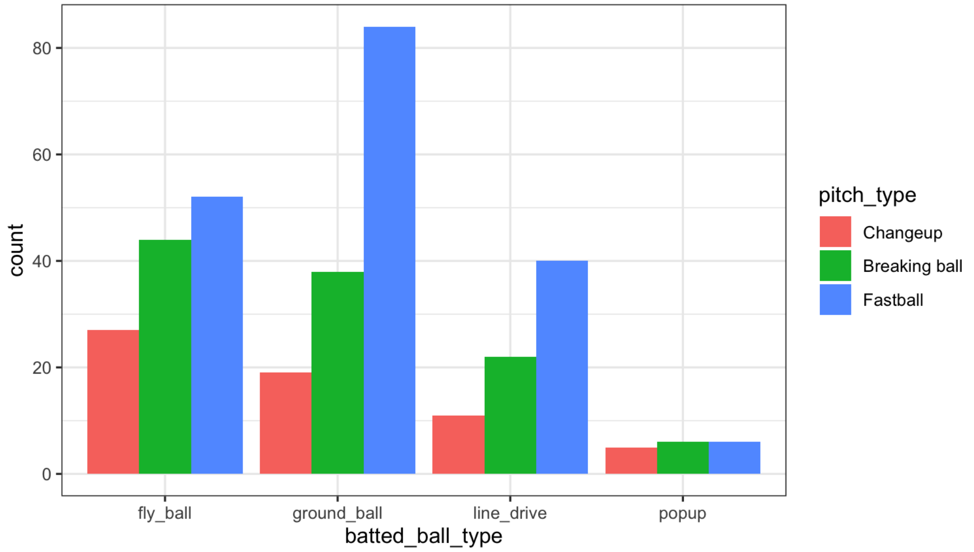
Stacked: a bar chart of *spine* charts

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

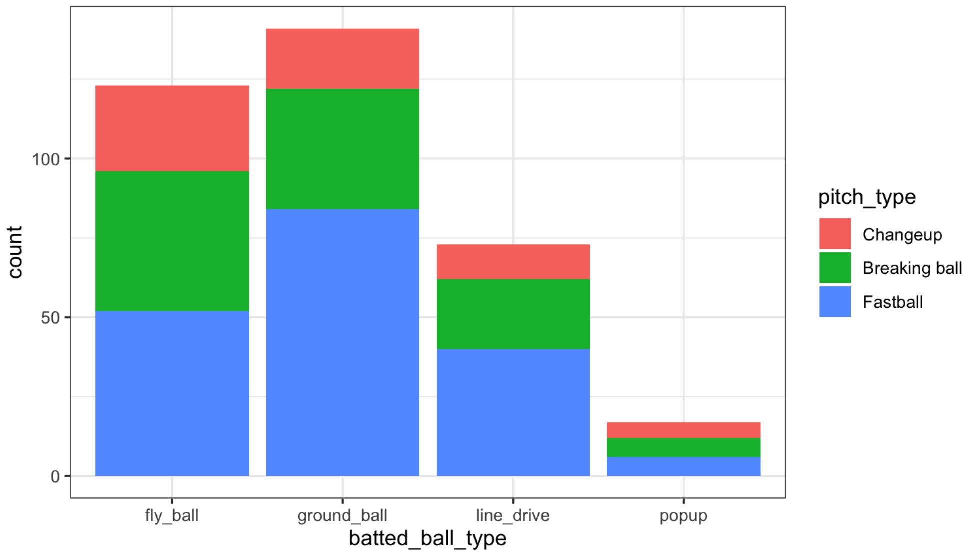


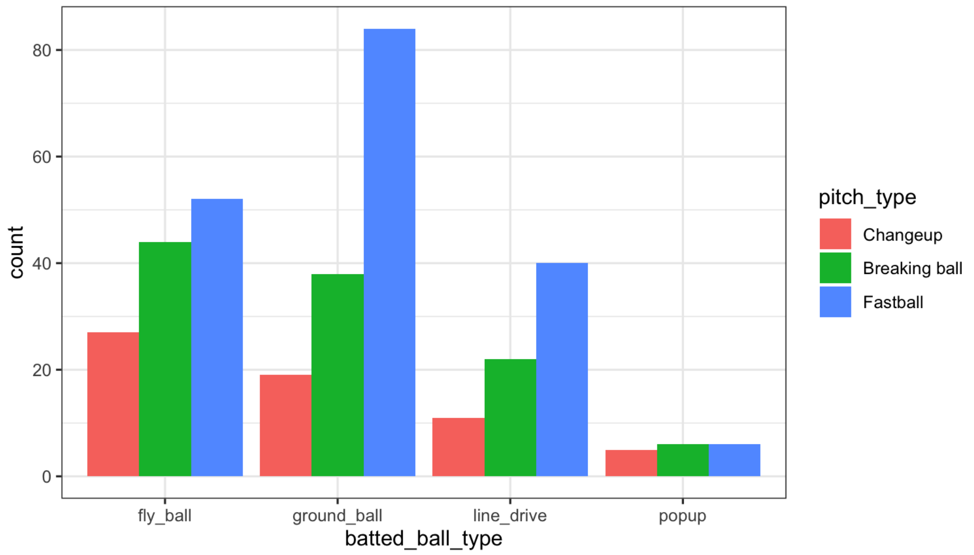
Side-by-Side: a bar chart *of bar charts*

```
ohtani_batted_balls %>%
  ggplot(aes(x = batted_ball_type,
             fill = pitch_type)) +
  geom_bar(position = "dodge") + theme_bw()
```



Which do you prefer?





- Stacked bar charts emphasize **marginal** distribution of `x` variable,
  - e.g. $P\,(\texttt{batted\_ball\_type} = \text{fly\_ball})$
- Side-by-side bar charts are useful to show the **conditional** distribution of `fill` variable given `x`,
  - e.g. $P\,(\texttt{pitch\_type} = \text{Fastball} \mid \texttt{batted\_ball\_type} = \text{fly\_ball})$

Contingency tables

Can provide `table()` with more than one variable

```r
table("Pitch type" = ohtani_batted_balls$pitch_type,
      "Batted ball type" = ohtani_batted_balls$batted_ball_type)
```

```
##                 Batted ball type
## Pitch type        fly_ball ground_ball line_drive popup
##   Changeup              27          19         11     5
##   Breaking ball         44          38         22     6
##   Fastball              52          84         40     6
```

Easily compute `proportions()`:

```r
proportions(table(ohtani_batted_balls$pitch_type, ohtani_batted_balls$batted_ball_type))
```

```
##
##                   fly_ball ground_ball line_drive      popup
##   Changeup      0.07627119  0.05367232 0.03107345 0.01412429
##   Breaking ball 0.12429379  0.10734463 0.06214689 0.01694915
##   Fastball      0.14689266  0.23728814 0.11299435 0.01694915
```

Review of joint, marginal, and conditional probabilities

Joint distribution: frequency of intersection, $P(X = x, Y = y)$

```
proportions(table(ohtani_batted_balls$pitch_type, ohtani_batted_balls$batted_ball_type))
```

```
##
##                   fly_ball ground_ball line_drive      popup
##   Changeup      0.07627119  0.05367232 0.03107345 0.01412429
##   Breaking ball 0.12429379  0.10734463 0.06214689 0.01694915
##   Fastball      0.14689266  0.23728814 0.11299435 0.01694915
```

Marginal distribution: row / column sums, e.g. $P(X = \text{popup}) = \sum_{y \in \text{pitch types}} P(X = \text{popup}, Y = y)$

Conditional distribution: probability event X **given** second event Y,

- e.g. $P(X = \text{popup} | Y = \text{Fastball}) = \frac{P(X = \text{popup}, Y = \text{Fastball})}{P(Y = \text{Fastball})}$

BONUS: `pivot_wider` example

Manually construct this table for practice…

```
library(gt)
ohtani_batted_balls %>%
  group_by(batted_ball_type, pitch_type) %>%
  summarize(joint_prob = n() / nrow(ohtani_batted_balls)) %>%
  pivot_wider(names_from = batted_ball_type, values_from = joint_prob,
              values_fill = 0)
```

```
## # A tibble: 3 × 5
##   pitch_type    fly_ball ground_ball line_drive  popup
##   <fct>            <dbl>       <dbl>      <dbl>  <dbl>
## 1 Changeup        0.0763      0.0537     0.0311 0.0141
## 2 Breaking ball   0.124       0.107      0.0621 0.0169
## 3 Fastball        0.147       0.237      0.113  0.0169
```

Inference for 2D categorical data

We AGAIN use the **chi-square test**:

- **Null hypothesis**: H_0: Variables A and B are independent,
    - e.g., `batted_ball_type` and `pitch_type` are independent of each other, no relationship
- And now we compute the **test statistic** as:

$$\chi^2 = \sum_i^{k_1} \sum_j^{k_2} \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$$

- O_{ij}: observed counts in contingency table j
- E_{ij}: expected counts under H_0 where **under the null**:

$$\begin{aligned} E_{ij} &= n \cdot P(A = a_i, B = b_j) \\ &= n \cdot P(A = a_i) P(B = b_j) \\ &= n \cdot \left(\frac{n_{i\cdot}}{n}\right) \left(\frac{n_{\cdot j}}{n}\right) \end{aligned}$$

Inference for 2D categorical data

We AGAIN use the **chi-square test**:

- **Null hypothesis**: H_0: Variables A and B are independent,
    - e.g., `batted_ball_type` and `pitch_type` are independent of each other, no relationship
- And now we compute the **test statistic** as:

$$\chi^2 = \sum_i^{k_1} \sum_j^{k_2} \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$$

```r
chisq.test(table(ohtani_batted_balls$pitch_type, ohtani_batted_balls$batted_ball_type))
```

```
##
##      Pearson's Chi-squared test
##
## data:  table(ohtani_batted_balls$pitch_type, ohtani_batted_balls$batted_ball_type)
## X-squared = 10.928, df = 6, p-value = 0.09062
```

Can we visualize independence?

Two variables are **independent** if knowing the level of one tells us nothing about the other

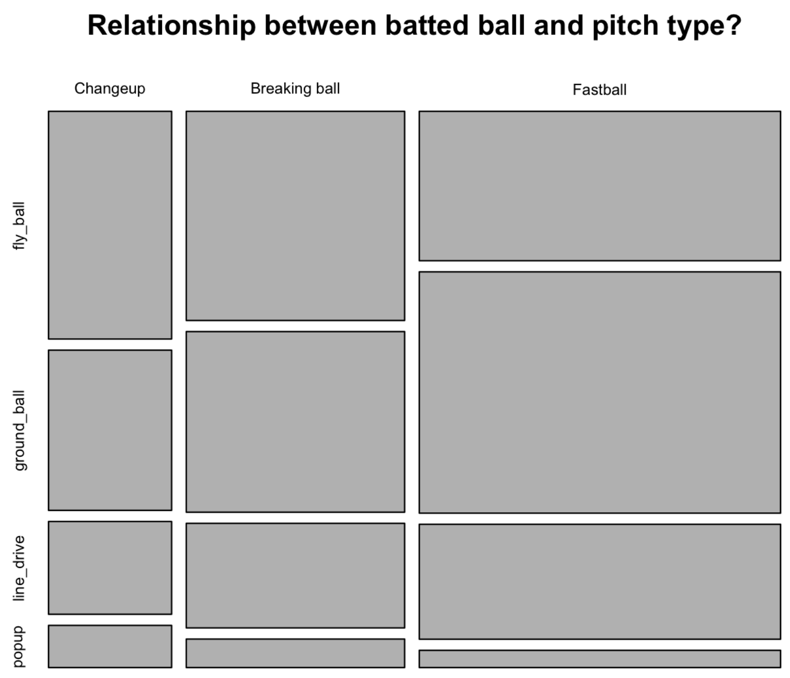
- i.e. $P(X = x | Y = y) = P(X = x)$, and that $P(X = x, Y = y) = P(X = x) \times P(Y = y)$

Create a **mosaic** plot using **base `R`**

```
mosaicplot(table(ohtani_batted_balls$pitch_ty
           main = "Relationship between batte
```

- spine chart *of spine charts*
- width $\propto$ marginal distribution of `pitch_type`
- height $\propto$ conditional distribution of `batted_ball_type | pitch_type`
- area $\propto$ joint distribution

`ggmosaic` has issues...



Shade by *Pearson residuals*

- The **test statistic** is:

$$\chi^2 = \sum_i^{k_1} \sum_j^{k_2} \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$$

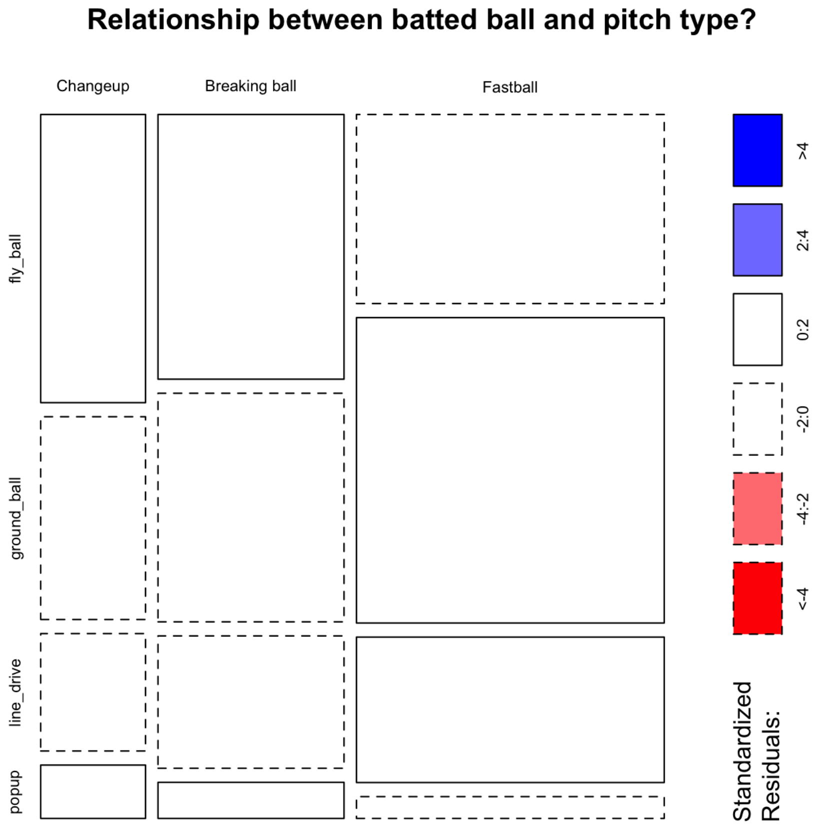
- Define the *Pearson residuals* as:

$$r_{ij} = \frac{O_{ij} - E_{ij}}{\sqrt{E_{ij}}}$$

- Sidenote: In general, Pearson residuals are $\frac{\text{residuals}}{\sqrt{\text{variance}}}$
- $r_{ij} \approx 0 \rightarrow$ observed counts are close to expected counts
- $|r_{ij}| > 2 \rightarrow$ "significant" at level $\alpha = 0.05$.
- Very positive $r_{ij} \rightarrow$ more than expected, while very negative $r_{ij} \rightarrow$ fewer than expected
- Mosaic plots: Color by Pearson residuals to tell us which combos are much bigger/smaller than expected.

Shade by *Pearson residuals*

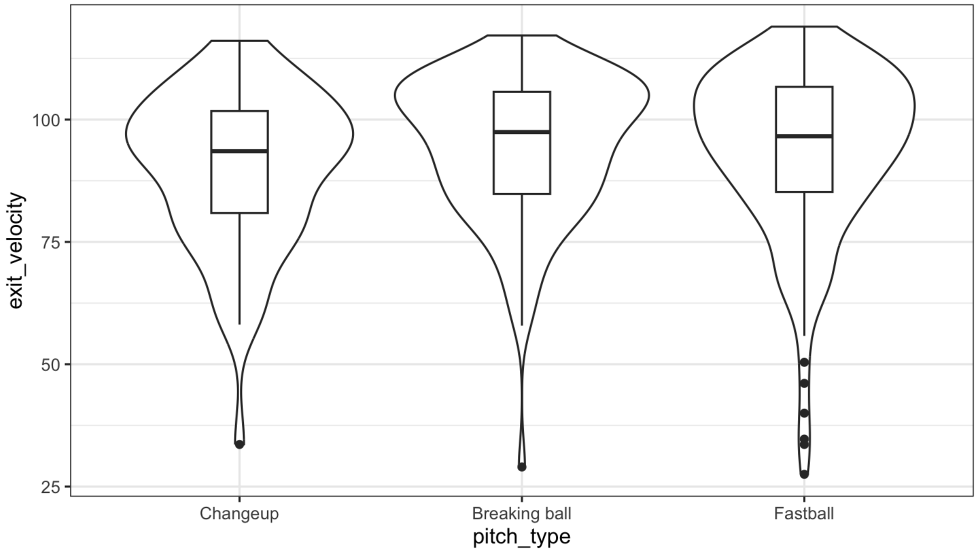
```
mosaicplot(table(ohtani_batted_balls$pitch_type, ohtani_batted_balls$batted_ball_type),
           shade = TRUE, main = "Relationship between batted ball and pitch type?")
```

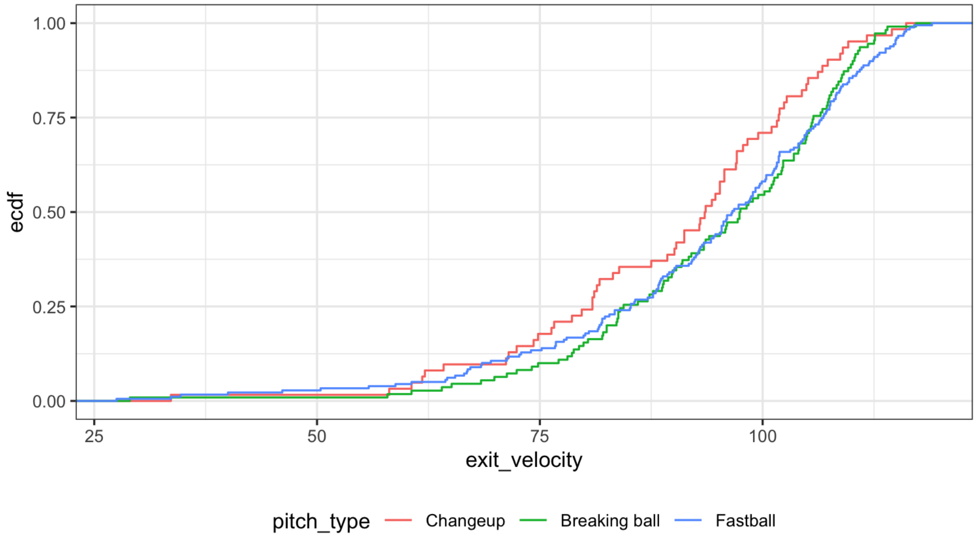


Continuous by categorical: side-by-side and color

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

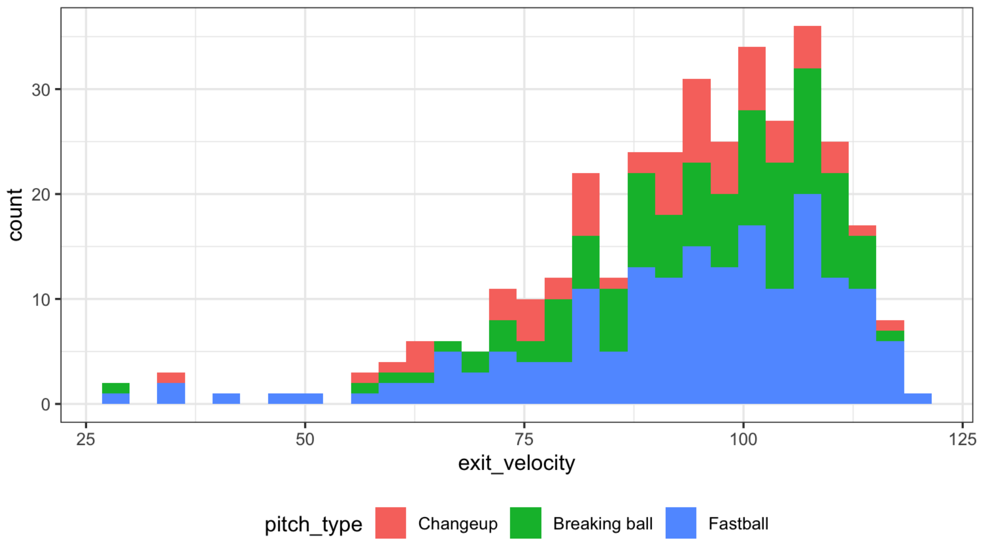
```
ohtani_batted_balls %>%
  ggplot(aes(x = exit_velocity,
             color = pitch_type)) +
  stat_ecdf() +
  theme_bw() +
  theme(legend.position = "bottom")
```



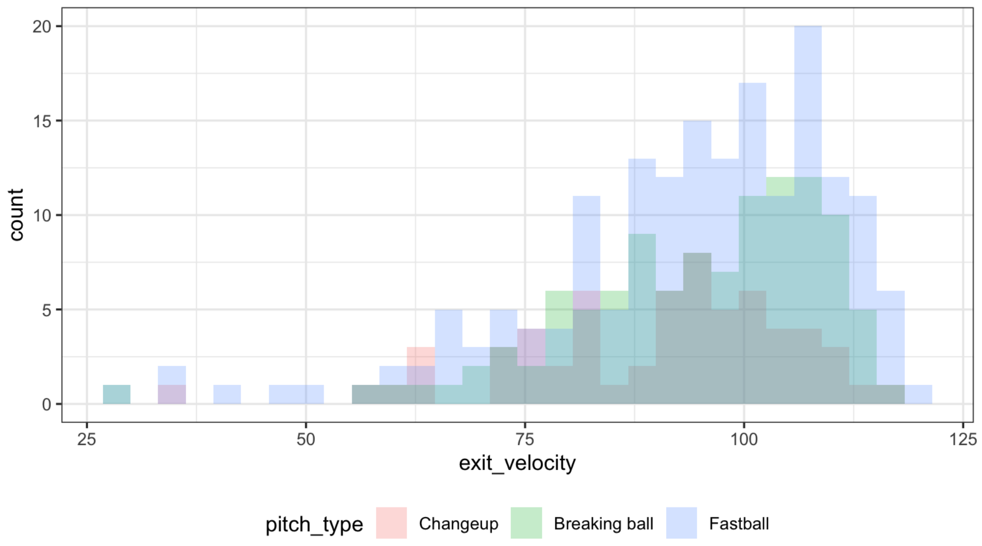


What about for histograms?

```
ohtani_batted_balls %>%
  ggplot(aes(x = exit_velocity,
             fill = pitch_type)) +
  geom_histogram() +
  theme_bw() + theme(legend.position = "botto
```

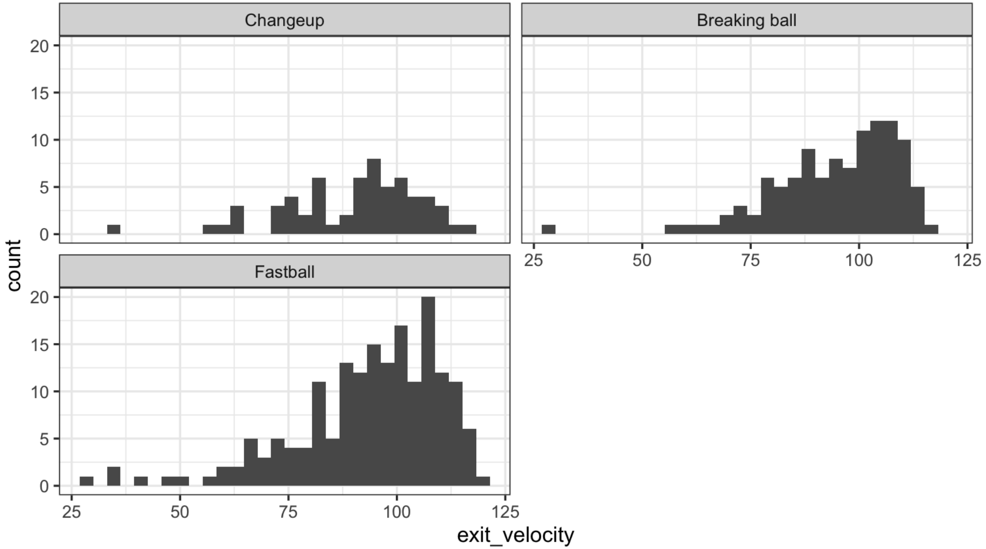



```
ohtani_batted_balls %>%
  ggplot(aes(x = exit_velocity,
             fill = pitch_type)) +
  geom_histogram(alpha = .25, position = "ide
  theme_bw() + theme(legend.position = "botto
```

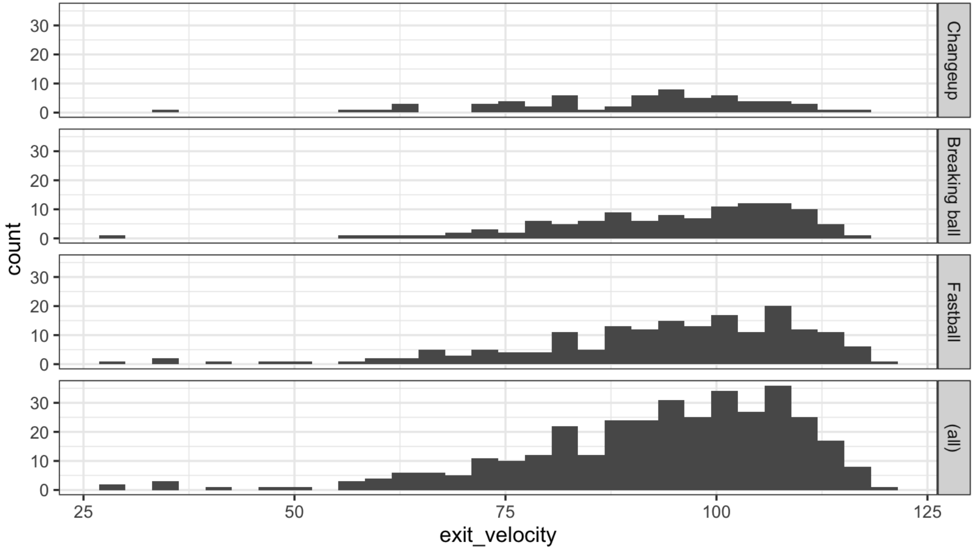


We can always facet instead...

```
ohtani_batted_balls %>%
  ggplot(aes(x = exit_velocity)) +
  geom_histogram() +
  theme_bw() +
  facet_wrap(~ pitch_type, ncol = 2)
```




```
ohtani_batted_balls %>%
  ggplot(aes(x = exit_velocity)) +
  geom_histogram() +
  theme_bw() +
  facet_grid(pitch_type ~., margins = TRUE)
```



Facets make it easy to move beyond 2D

```
ohtani_batted_balls %>%
  ggplot(aes(x = pitch_type,
             fill = batted_ball_type)) +
  geom_bar() + theme_bw() +
  facet_wrap(~ outcome, ncol = 5) +
  theme(legend.position = "bottom")
```