DataFest Training Baseball Data Set - Royals Analysis

Matthew Wolz

2025-03-29

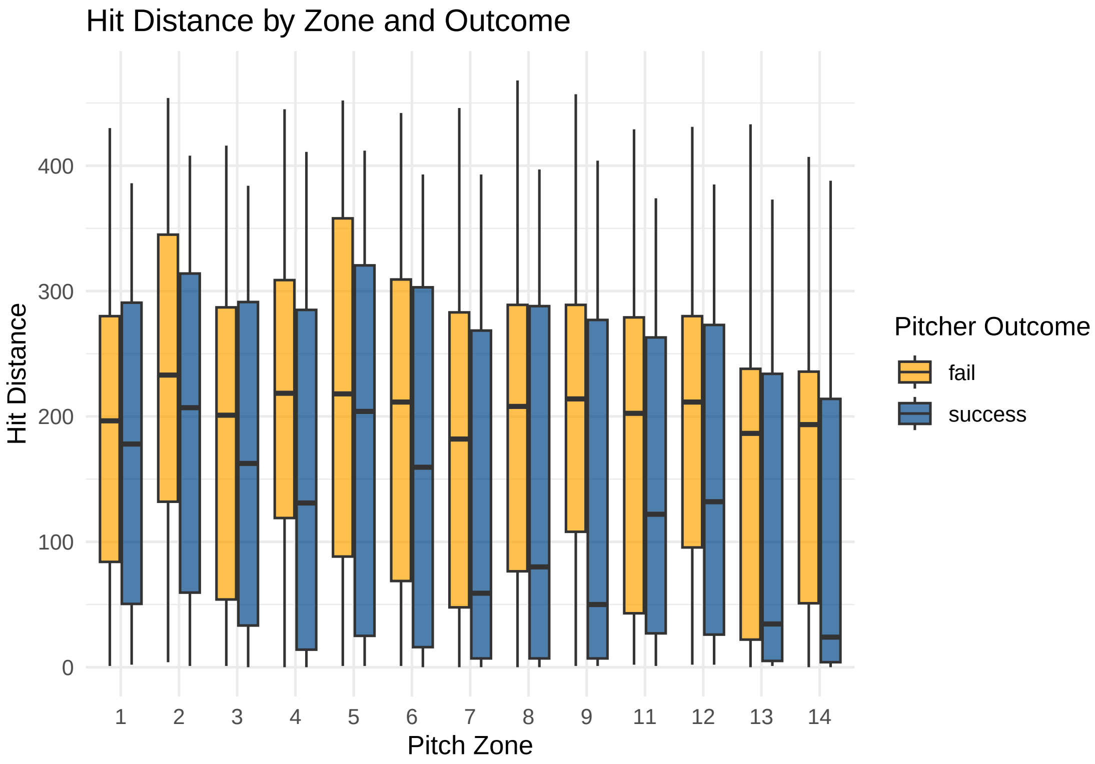
Contents

```r
library(tidyverse)
library(ggplot2)
library(scales)
library(randomForest)
library(vip)
library(caret)
library(knitr)
library(kableExtra)
```

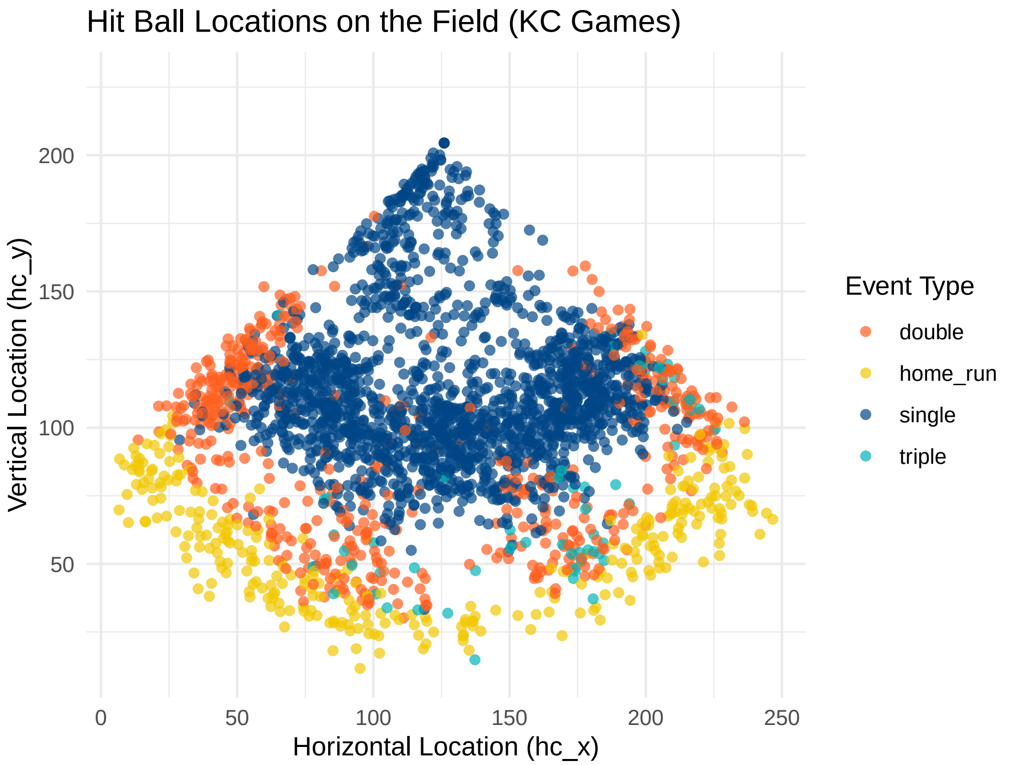
```r
data_baseball <- read_csv("statcast_pitch_swing_data_20240402_20241030_with_arm_angle.csv")

# Vector of Kansas City Royals Hitters
kc_batters <- c(
  "Witt Jr., Bobby", "Perez, Salvador", "Garcia, Maikel",
  "Pasquantino, Vinnie", "Melendez, MJ", "Renfroe, Hunter",
  "Isbel, Kyle", "Fermin, Freddy", "Massey, Michael",
  "Frazier, Adam", "Hampson, Garrett", "Velazquez, Nelson",
  "Loftin, Nick", "Blanco, Dairon", "DeJong, Paul",
  "Pham, Tommy", "Gurriel, Yuli", "Grossman, Robbie",
  "Waters, Drew", "Alexander, CJ", "Gentry, Tyler"
)
```

```r
kc_data <- data_baseball %>%
  filter(home_team == "KC" | away_team == "KC") %>%
  mutate(
    vx0_normalized = scale(vx0),
    vy0_normalized = scale(vy0),
    vz0_normalized = scale(vz0),
    ax_normalized = scale(ax),
    ay_normalized = scale(ay),
    az_normalized = scale(az)
  ) %>%
  mutate(
    combined_normalized_velocity = rowMeans(select(., vx0_normalized, vy0_normalized, vz0_normalized)),
    combined_normalized_acceleration = rowMeans(select(., ax_normalized, ay_normalized, az_normalized))
  ) %>%
  # Exclude rows where the batter is on the KC 2024 roster
  filter(!(batter %in% kc_batters)) %>%
  select(
    events, pitcher, batter, release_speed, release_spin_rate, effective_speed,
    pfx_x, pfx_z, zone, plate_x, plate_z, events, hit_distance_sc,
    woba_value, delta_pitcher_run_exp, game_date, outs_when_up,
    home_team, away_team, hc_x, hc_y, description, combined_normalized_velocity, combined_normalized_acc
  ) %>%
  mutate(pitcher_outcome = case_when(
    events %in% c("strikeout", "field_out", "double_play", "strikeout_double_play", "force_out", "fielde
    events %in% c("single", "double", "triple", "home_run", "walk", "hit_by_pitch",
                  "fielders_choice", "sac_bunt", "sac_fly", "sac_fly_double_play") ~ "fail"  )) %>%
  filter(!is.na(pitcher_outcome))
```

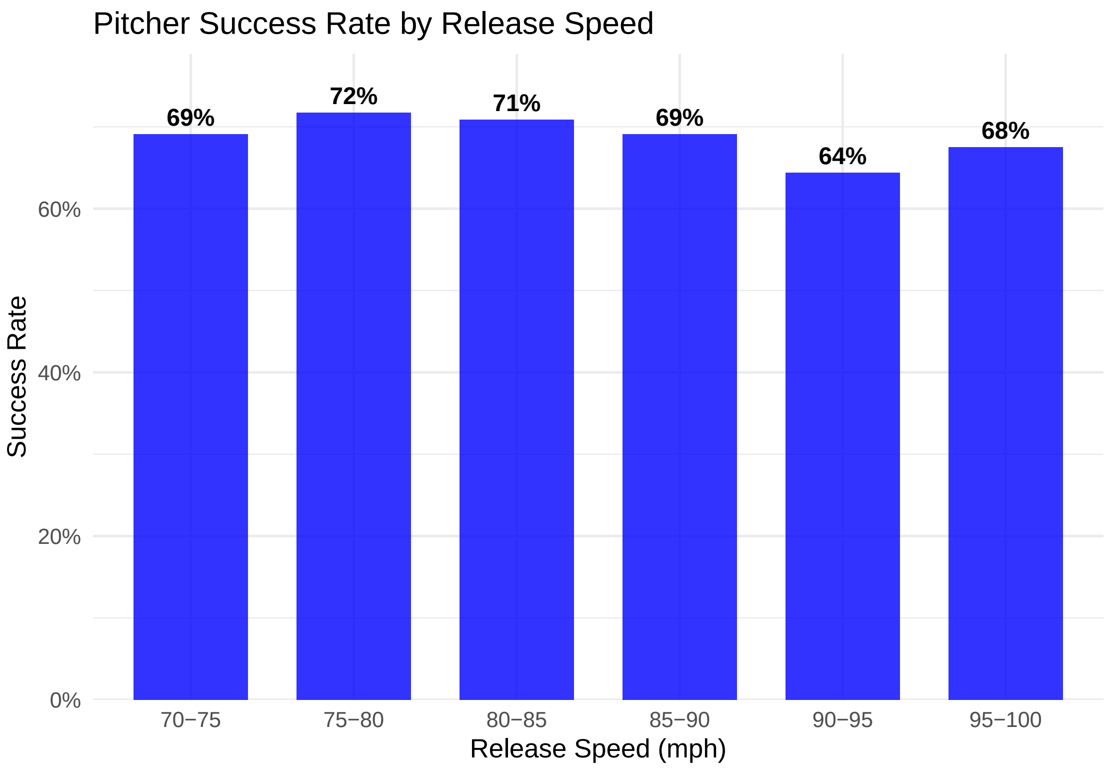
```
ggplot(kc_data, aes(x = factor(zone), y = hit_distance_sc, fill = pitcher_outcome)) +
  geom_boxplot(alpha = 0.7) +
  labs(
    title = "Hit Distance by Zone and Outcome",
    x = "Pitch Zone",
    y = "Hit Distance",
    fill = "Pitcher Outcome"
  ) +
  scale_fill_manual(values = c("success" = "#004687", "fail" = "#FFA500")) +
  theme_minimal()
```



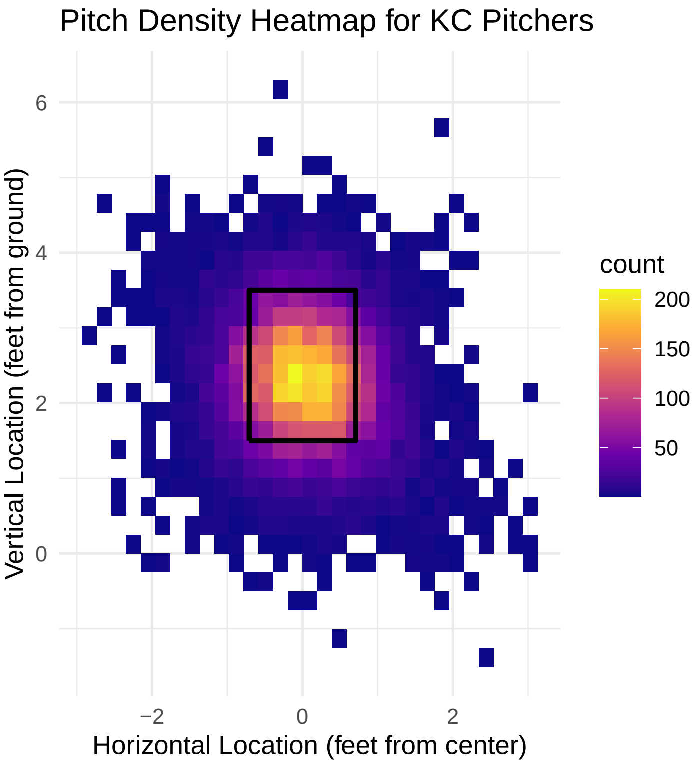
```
# Clean version without NA points
ggplot(kc_data %>% filter(!is.na(hc_x), !is.na(hc_y), !is.na(events)),
       aes(x = hc_x, y = hc_y)) +
  geom_point(aes(color = events), alpha = 0.7) +
  geom_curve(x = 50, xend = 200, y = -100, yend = -100, curvature = -.65) +
  geom_segment(x = 100, xend = 50, y = -150, yend = -100) +
  geom_segment(x = 100, xend = 200, y = -150, yend = -100) +
  geom_curve(x = 75, xend = 150, y = -120, yend = -121, curvature = -.65, linetype = "dotted") +
  coord_fixed() +
  theme_minimal() +
  labs(
    title = "Hit Ball Locations on the Field (KC Games)",
    x = "Horizontal Location (hc_x)",
    y = "Vertical Location (hc_y)",
    color = "Event Type"
  ) +
  scale_color_manual(
    values = c(
      "single" = "#004687",
      "double" = "#FF5F1F",
      "triple" = "#00B5B8",
      "home_run" = "#F2C800"
    ),
    na.value = NA  # This explicitly removes NA values from the plot
  )
```



```
speed_bins_clean <- kc_data %>%
  filter(!is.na(release_speed), !is.na(pitcher_outcome),
         release_speed >= 70, release_speed <= 100) %>%
  mutate(
    speed_bin = cut(
      release_speed,
      breaks = seq(70, 100, by = 5),
      labels = paste0(seq(70, 95, by = 5), "-", seq(75, 100, by = 5)),
      include.lowest = TRUE
    )
  ) %>%
  group_by(speed_bin) %>%
  summarise(success_rate = mean(pitcher_outcome == "success"), .groups = "drop")

ggplot(speed_bins_clean, aes(x = speed_bin, y = success_rate)) +
  geom_col(fill = "blue", width = 0.7, alpha = 0.8) +
  geom_text(
    aes(label = percent(success_rate, accuracy = 1)),
    vjust = -0.5,
    size = 3.5,
    fontface = "bold"
  ) +
  scale_y_continuous(
    labels = percent_format(),
    limits = c(0, max(speed_bins_clean$success_rate) * 1.1),
    expand = c(0, 0)
  ) +
  labs(
    title = "Pitcher Success Rate by Release Speed",
    x = "Release Speed (mph)",
    y = "Success Rate"
  ) +
  theme_minimal()
```



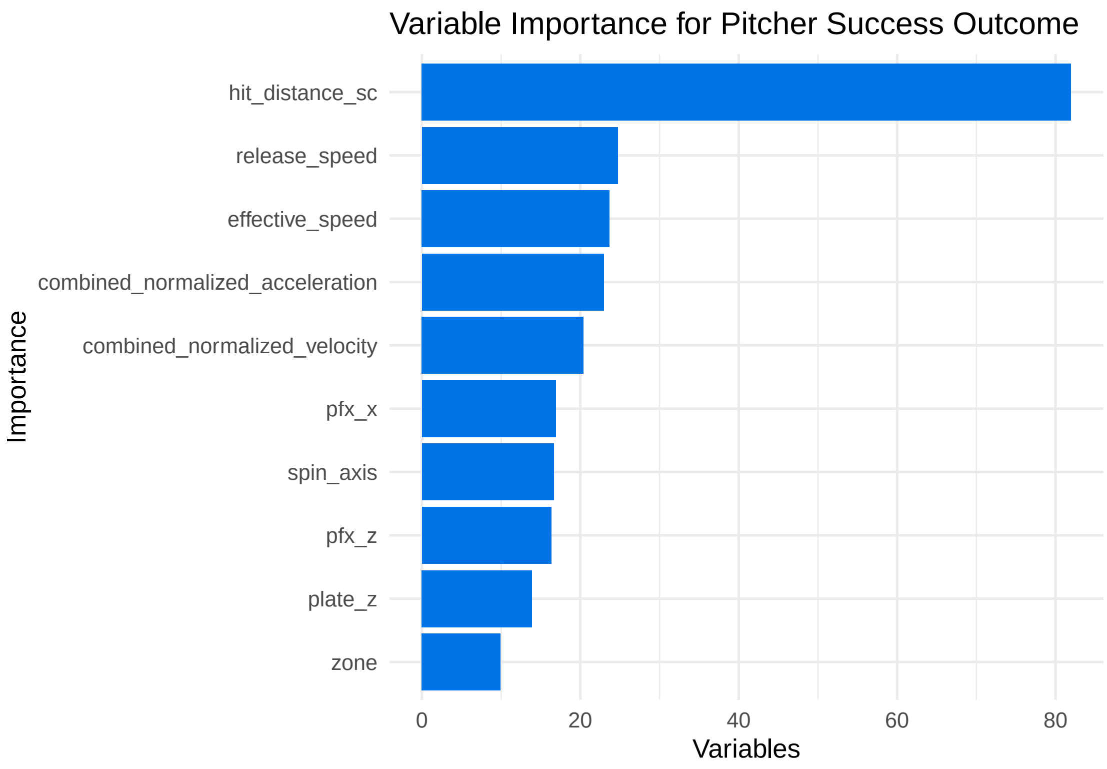
```r
# Create strike zone coordinates
strike_zone <- data.frame(
  x = c(-0.708, 0.708, 0.708, -0.708, -0.708),  # Horizontal edges (ft from center)
  y = c(1.5, 1.5, 3.5, 3.5, 1.5)                # Vertical edges (ft from ground)
)

# Create the heatmap with proper strike zone overlay
ggplot(kc_data, aes(x = plate_x, y = plate_z)) +
  geom_bin2d(bins = 30) +
  scale_fill_viridis_c(option = "plasma") +
  geom_path(
    data = strike_zone,
    aes(x = x, y = y),
    color = "black",
    linewidth = 1,
    inherit.aes = FALSE  # Important: don't inherit main plot aesthetics
  ) +
  coord_fixed() +
  labs(
    title = "Pitch Density Heatmap for KC Pitchers",
    x = "Horizontal Location (feet from center)",
    y = "Vertical Location (feet from ground)"
  ) +
  theme_minimal()
```



```r
rf_data <- kc_data %>%
  mutate(pitcher_outcome = factor(pitcher_outcome, levels = c("fail", "success"))) %>%
  select(-c("pitcher", "batter", "game_date", "events", "woba_value",
           "delta_pitcher_run_exp", "description", "hc_x", "hc_y"))

set.seed(123)
train_index <- createDataPartition(rf_data$pitcher_outcome, p = 0.8, list = FALSE)
train_data <- rf_data[train_index, ]
test_data <- rf_data[-train_index, ]

rf_model <- randomForest(pitcher_outcome ~ .,
                         data = train_data,
                         importance = TRUE,
                         na.action = na.omit)

predictions <- predict(rf_model, test_data)
conf_matrix <- confusionMatrix(predictions, test_data$pitcher_outcome)
print(conf_matrix)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction fail success
##    fail     136      73
##    success  436    1004
##
##                Accuracy : 0.6913
##                  95% CI : (0.6684, 0.7136)
##     No Information Rate : 0.6531
##     P-Value [Acc > NIR] : 0.0005555
##
##                   Kappa : 0.1997
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.23776
##             Specificity : 0.93222
##          Pos Pred Value : 0.65072
##          Neg Pred Value : 0.69722
##              Prevalence : 0.34688
##          Detection Rate : 0.08247
##    Detection Prevalence : 0.12674
##       Balanced Accuracy : 0.58499
##
##        'Positive' Class : fail
##
```

```r
vip(rf_model) +
  geom_bar(stat = "identity", fill = "#0073e6") +
  labs(
    title = "Variable Importance for Pitcher Success Outcome",
    x = "Importance",
    y = "Variables"
  ) +
  theme_minimal()
```



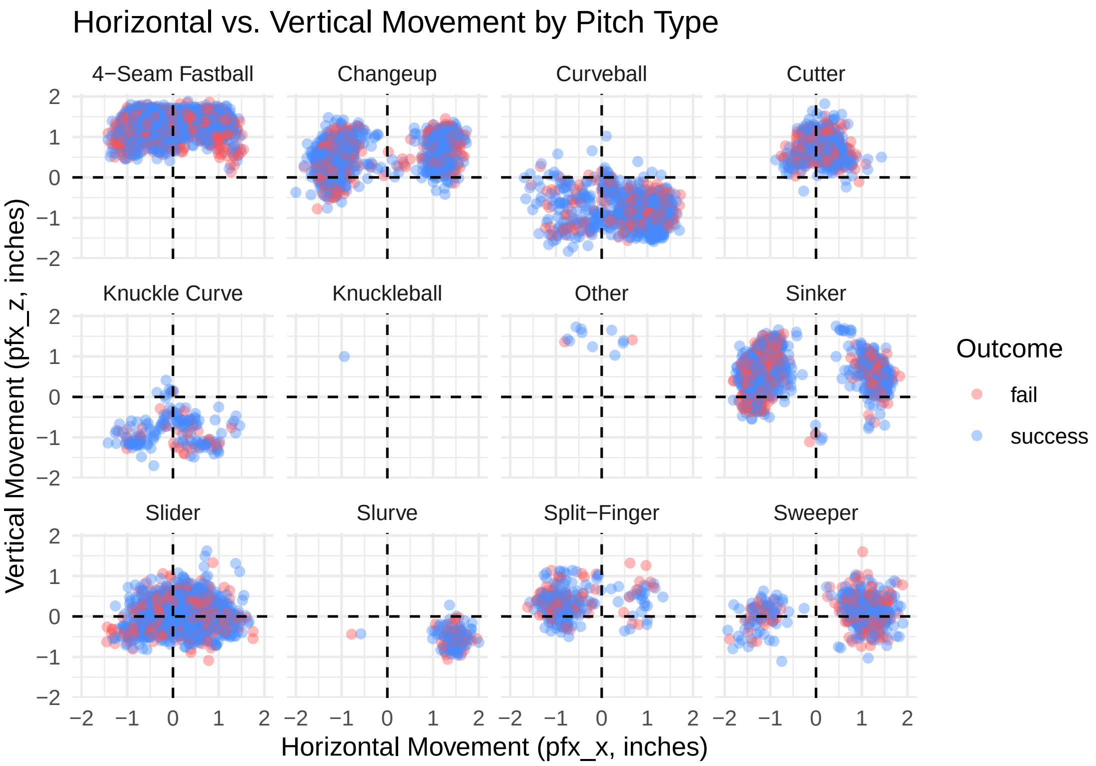
```r
# Prepare the data with normalized metrics
pp_data <- data_baseball %>%
  filter(home_team == "KC" | away_team == "KC") %>%
  mutate(
    # Convert scaled values to numeric vectors
    vx0_normalized = as.numeric(scale(vx0)),
    vy0_normalized = as.numeric(scale(vy0)),
    vz0_normalized = as.numeric(scale(vz0)),
    ax_normalized = as.numeric(scale(ax)),
    ay_normalized = as.numeric(scale(ay)),
    az_normalized = as.numeric(scale(az))
  ) %>%
  mutate(
    combined_normalized_velocity = rowMeans(cbind(vx0_normalized, vy0_normalized, vz0_normalized), na.r
    combined_normalized_acceleration = rowMeans(cbind(ax_normalized, ay_normalized, az_normalized), na.
  ) %>%
  mutate(
    Pitcher = case_when(
      pitcher == 666142 ~ "Cole Ragans",
      pitcher == 607625 ~ "Seth Lugo",
      pitcher == 663903 ~ "Brady Singer",
      pitcher == 608379 ~ "Michael Wacha",
      pitcher == 679525 ~ "Alec Marsh",
      TRUE ~ as.character(pitcher)
    )
  ) %>%
  filter(Pitcher %in% c("Cole Ragans", "Seth Lugo", "Brady Singer", "Michael Wacha", "Alec Marsh")) %>%
  mutate(
    pitcher_outcome = case_when(
      events %in% c("strikeout", "field_out", "double_play", "strikeout_double_play",
                    "force_out", "fielders_choice_out", "field_error",
                    "catcher_interf", "truncated_pa") ~ "success",
      events %in% c("single", "double", "triple", "home_run", "walk", "hit_by_pitch",
                    "fielders_choice", "sac_bunt", "sac_fly", "sac_fly_double_play") ~ "fail",
      TRUE ~ NA_character_
    )
  ) %>%
  filter(!is.na(pitcher_outcome))

# Calculate summary statistics
summary_stats <- pp_data %>%
  group_by(Pitcher) %>%
  summarise(
    Total_Pitches = n(),
    Total_Success = sum(pitcher_outcome == "success"),
    Total_Fail = sum(pitcher_outcome == "fail"),
    Avg_Hit_Distance = round(mean(hit_distance_sc, na.rm = TRUE), 2),
    Avg_Release_Speed = round(mean(release_speed, na.rm = TRUE), 2),
    Avg_Effective_Speed = round(mean(effective_speed, na.rm = TRUE), 2),
    Avg_Normalized_Accel = round(mean(combined_normalized_acceleration, na.rm = TRUE), 2),
    Avg_Normalized_Velocity = round(mean(combined_normalized_velocity, na.rm = TRUE), 2),
    .groups = "drop"
  ) %>%
  mutate(Success_Percentage = round((Total_Success / Total_Pitches) * 100, 2)) %>%
  arrange(desc(Success_Percentage))

# Display the table
summary_stats %>%
  kable(format = "latex", booktabs = TRUE, caption = "Pitcher Performance Summary") %>%
  kable_styling(latex_options = c("striped", "hold_position", "scale_down")) %>%
  column_spec(1, bold = TRUE) %>%
  row_spec(0, bold = TRUE, color = "white", background = "#007bff") %>%
  add_header_above(c(" " = 1, "Basic Stats" = 2, "Performance Metrics" = 5, "Normalized Metrics" = 2))
  scroll_box(width = "100%", height = "300px")
```

Table 1: Pitcher Performance Summary

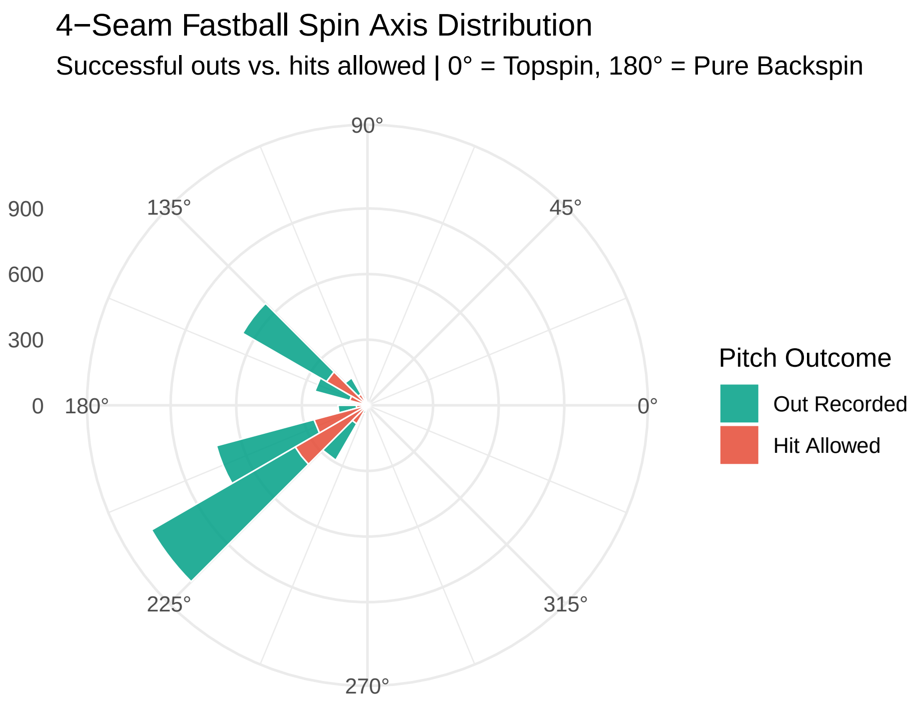
	Basic Stats		Performance Metrics					Normalized Metrics	
Pitcher	**Total_Pitches**	**Total_Success**	**Total_Fail**	**Avg_Hit_Distance**	**Avg_Release_Speed**	**Avg_Effective_Speed**	**Avg_Normalized_Accel**	**Avg_Normalized_Velocity**	**Success_Percentage**
Cole Ragans	766	538	228	168.47	90.13	89.93	0.65	-0.74	70.23
Seth Lugo	839	586	253	170.08	87.68	87.29	-0.11	0.30	69.85
Michael Wacha	684	461	223	172.71	87.64	88.47	-0.15	0.02	67.40
Alec Marsh	539	361	178	176.38	89.85	89.68	-0.15	0.22	66.98
Brady Singer	721	481	240	162.11	87.50	88.91	-0.33	0.35	66.71

```
strikeouts_per_pitcher <- data_baseball %>%
  filter(home_team == "KC" | away_team == "KC") %>%
  filter(events == "strikeout") %>%
  group_by(pitcher) %>%
  summarise(strikeouts = n()) %>%
  arrange(desc(strikeouts))
```

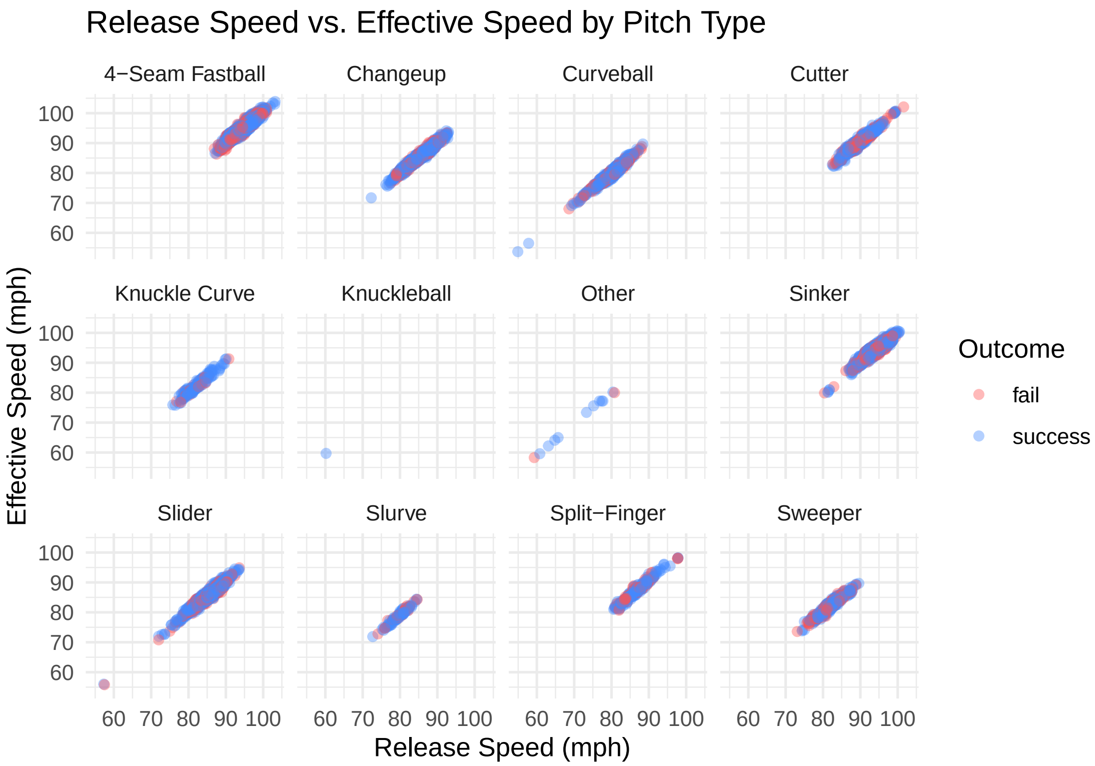
```r
ggplot(kc_data, aes(x = pfx_x, y = pfx_z, color = pitcher_outcome)) +
  geom_point(alpha = 0.4) +
  facet_wrap(~pitch_name) +
  geom_vline(xintercept = 0, linetype = "dashed") +
  geom_hline(yintercept = 0, linetype = "dashed") +
  scale_color_manual(values = c("fail" = "#FF5252", "success" = "#448AFF")) +
  labs(
    title = "Horizontal vs. Vertical Movement by Pitch Type",
    x = "Horizontal Movement (pfx_x, inches)",
    y = "Vertical Movement (pfx_z, inches)",
    color = "Outcome"
  ) +
  theme_minimal()
```



```r
fastball_data <- kc_data %>%
  filter(pitch_name == "4-Seam Fastball",
         !is.na(spin_axis),
         !is.na(pitcher_outcome)) %>%
  mutate(
    spin_axis = as.numeric(spin_axis),
    pitcher_outcome = factor(pitcher_outcome,
                             levels = c("success", "fail"),
                             labels = c("Out Recorded", "Hit Allowed"))
  )

ggplot(fastball_data, aes(x = spin_axis, fill = pitcher_outcome)) +
  geom_histogram(
    binwidth = 15,
    boundary = 0,
    color = "white",
    linewidth = 0.3,
    position = "stack",
    alpha = 0.85
  ) +
  coord_polar(start = -pi/2, direction = -1) +
  scale_x_continuous(
    limits = c(0, 360),
    breaks = seq(0, 315, by = 45),
    labels = c("0°", "45°", "90°", "135°", "180°", "225°", "270°", "315°")
  ) +
  scale_fill_manual(
    values = c("Out Recorded" = "#00A087", "Hit Allowed" = "#E64B35"),
    guide = guide_legend(title = "Pitch Outcome")
  ) +
  labs(
    title = "4-Seam Fastball Spin Axis Distribution",
    subtitle = "Successful outs vs. hits allowed | 0° = Topspin, 180° = Pure Backspin",
    x = "",
    y = ""
  ) +
  theme_minimal()
```



```r
ggplot(kc_data, aes(x = release_speed, y = effective_speed, color = pitcher_outcome)) +
  geom_point(alpha = 0.4) +
  facet_wrap(~pitch_name) +
  scale_color_manual(values = c("fail" = "#FF5252", "success" = "#448AFF")) +
  labs(
    title = "Release Speed vs. Effective Speed by Pitch Type",
    x = "Release Speed (mph)",
    y = "Effective Speed (mph)",
    color = "Outcome"
  ) +
  theme_minimal()
```


Release Speed vs. Effective Speed by Pitch Type
4−Seam Fastball
Changeup
Curveball
Cutter
Knuckle Curve
Knuckleball
Other
Sinker
Slider
Slurve
Split−Finger
Sweeper
Effective Speed (mph)
Release Speed (mph)
Outcome
fail
success