Ray_Methods

2025-11-30

Data Loading

```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.1     v stringr   1.5.2
## v ggplot2   4.0.0     v tibble    3.3.0
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.1.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

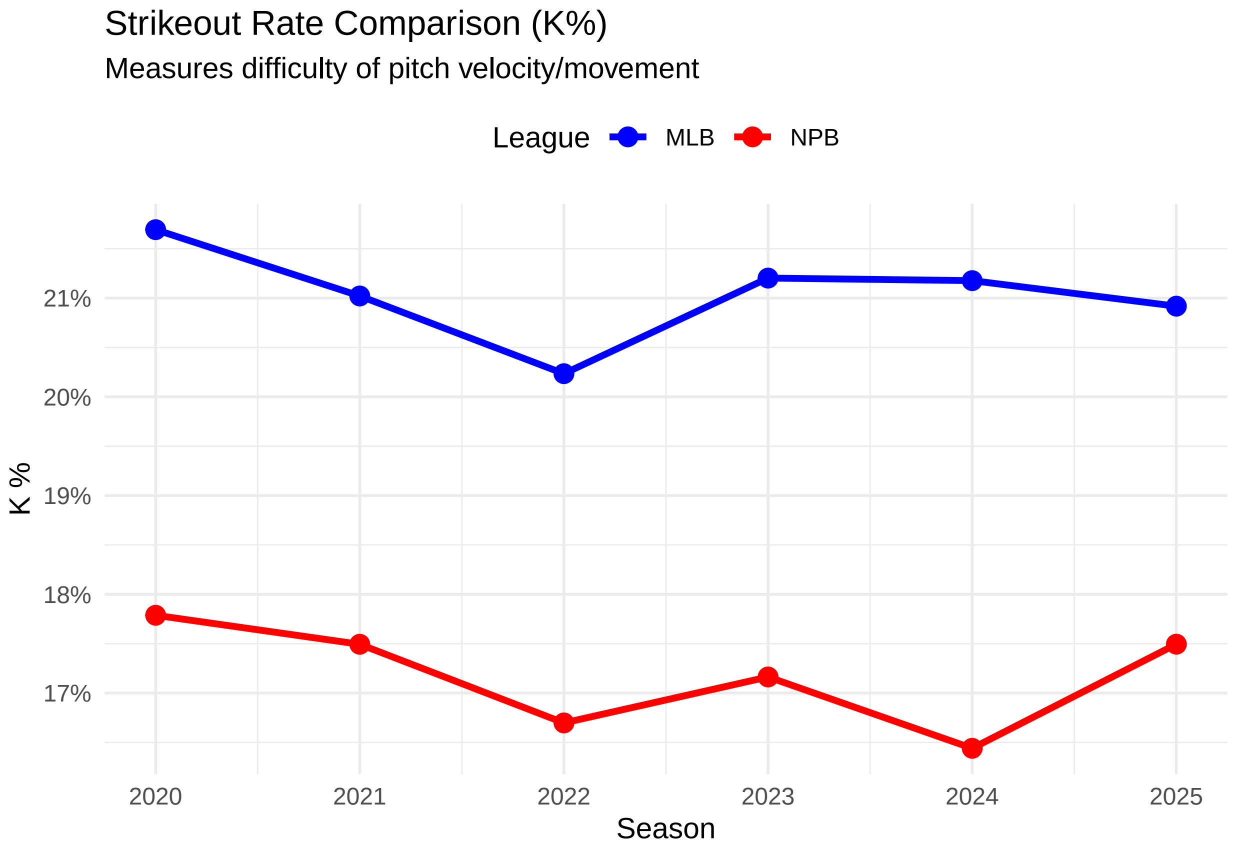
```r
mlb <- read_csv("mlb_stats.csv", show_col_types = FALSE) %>%
  rename(Name = G...2)
```

```
## New names:
## * ‘G‘ -> ‘G...2‘
## * ‘G‘ -> ‘G...6‘
```

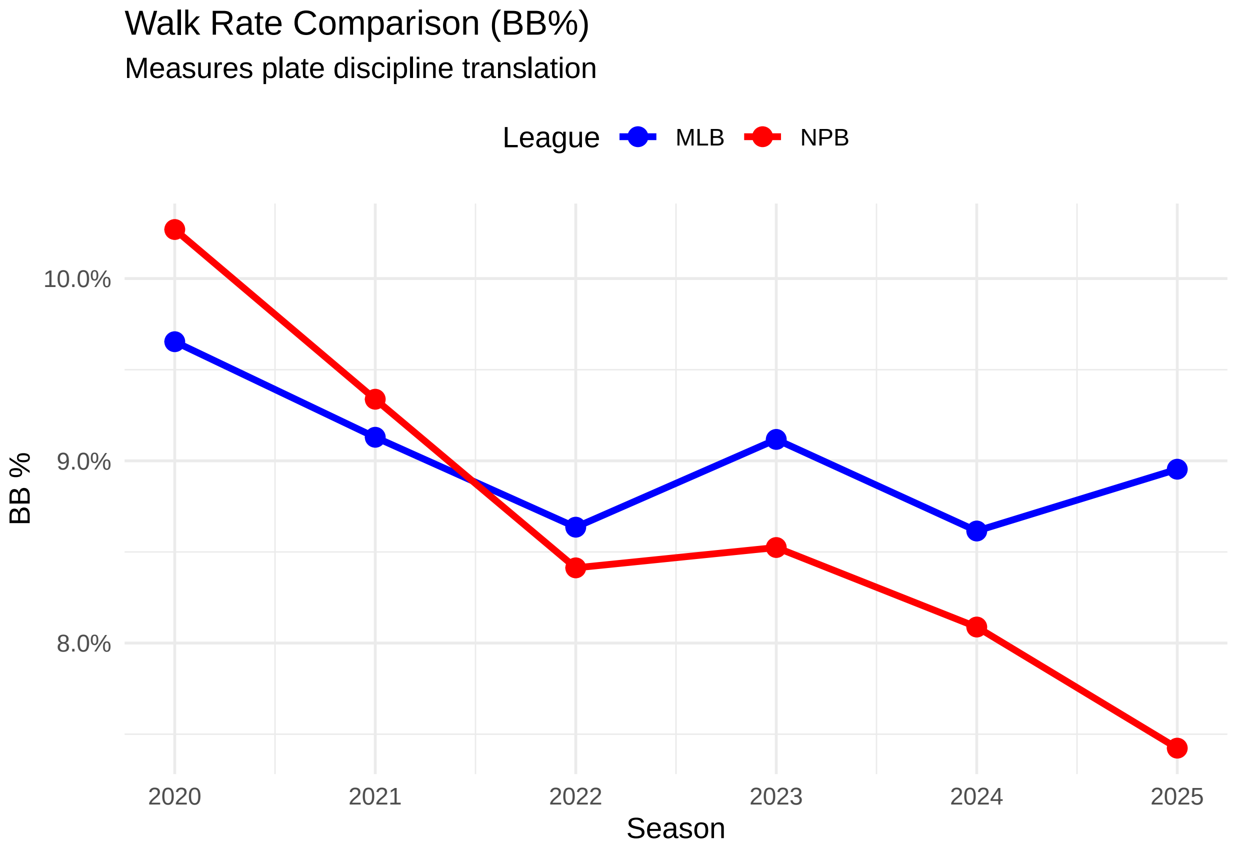
```r
npb <- read.csv("npb_stats.csv")

mlb_prep <- mlb %>% select(SEASON, PA, SO, BB, HR) %>% mutate(League = "MLB")
npb_prep <- npb %>% mutate(SEASON = Season) %>% select(SEASON, PA, SO, BB, HR) %>% mutate(League = "NPB

combined_data <- bind_rows(mlb_prep, npb_prep) %>%
  filter(SEASON >= 2020, SEASON <= 2025) %>%
  mutate(
    K_Rate = SO / PA,
    BB_Rate = BB / PA,
    HR_Rate = HR / PA
  ) %>%
  group_by(SEASON, League) %>%
  summarise(
    Strikeout_Pct = mean(K_Rate, na.rm = TRUE),
    Walk_Pct = mean(BB_Rate, na.rm = TRUE),
    Homerun_Pct = mean(HR_Rate, na.rm = TRUE),
    .groups = "drop"
  )
```

MLB vs. NPB

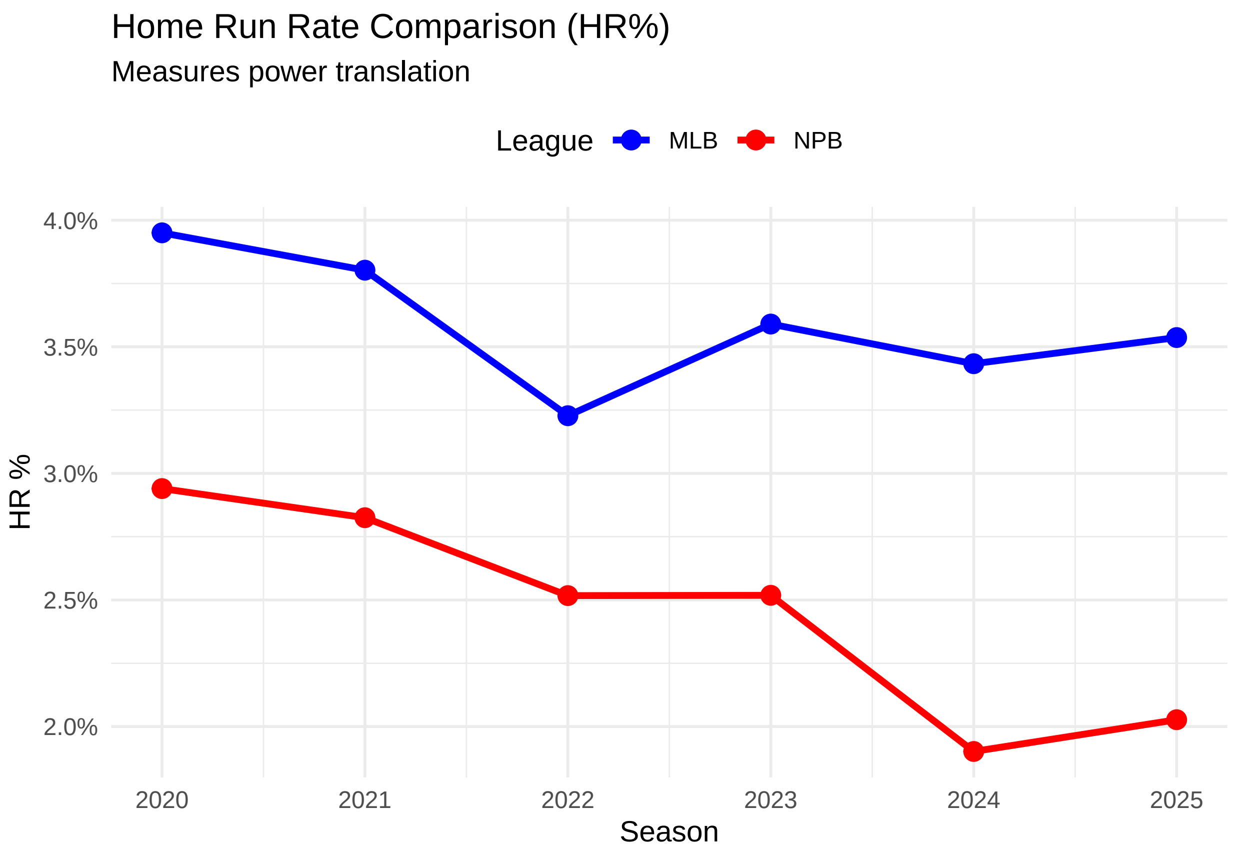
```
p1 <- ggplot(combined_data, aes(x = SEASON, y = Strikeout_Pct, color = League)) +
  geom_line(linewidth = 1.2) +
  geom_point(size = 3) +
  scale_color_manual(values = c("NPB" = "red", "MLB" = "blue")) +
  scale_y_continuous(labels = scales::percent) +
  scale_x_continuous(breaks = 2020:2025) +
  labs(
    title = "Strikeout Rate Comparison (K%)",
    subtitle = "Measures difficulty of pitch velocity/movement",
    y = "K %", x = "Season"
  ) +
  theme_minimal() +
  theme(legend.position = "top")

print(p1)
```



```
p2 <- ggplot(combined_data, aes(x = SEASON, y = Walk_Pct, color = League)) +
  geom_line(linewidth = 1.2) +
  geom_point(size = 3) +
  scale_color_manual(values = c("NPB" = "red", "MLB" = "blue")) +
  scale_y_continuous(labels = scales::percent) +
  scale_x_continuous(breaks = 2020:2025) +
  labs(
    title = "Walk Rate Comparison (BB%)",
    subtitle = "Measures plate discipline translation",
    y = "BB %", x = "Season"
  ) +
  theme_minimal() +
  theme(legend.position = "top")

print(p2)
```



```
p3 <- ggplot(combined_data, aes(x = SEASON, y = Homerun_Pct, color = League)) +
  geom_line(linewidth = 1.2) +
  geom_point(size = 3) +
  scale_color_manual(values = c("NPB" = "red", "MLB" = "blue")) +
  scale_y_continuous(labels = scales::percent) +
  scale_x_continuous(breaks = 2020:2025) +
  labs(
    title = "Home Run Rate Comparison (HR%)",
    subtitle = "Measures power translation",
    y = "HR %", x = "Season"
  ) +
  theme_minimal() +
  theme(legend.position = "top")

print(p3)
```



```
library(lubridate)

posting_raw <- read.csv("posting_system.csv", stringsAsFactors = FALSE)

valid_npb_teams <- c(
  "Yomiuri", "Hanshin", "Seibu", "Rakuten", "Marines",
  "Swallows", "BayStars", "Dragons", "Carp",
  "Fighters", "Buffaloes", "Hawks", "BlueWave", "Kintetsu"
)

manifest <- posting_raw %>%
  mutate(
    Post_Date_Clean = mdy(Posting.Date),
    Post_Year = year(Post_Date_Clean),
    Post_Month = month(Post_Date_Clean),
    MLB_First_Season = if_else(Post_Month >= 11, Post_Year + 1, Post_Year),
    NPB_Final_Season = MLB_First_Season - 1,
  ) %>%
  filter(
    MLB_First_Season >= 2015,
    str_detect(NPB.Team, paste(valid_npb_teams, collapse = "|")),
    !str_detect(tolower(MLB.Team), "none")
  ) %>%
  select(Player, NPB.Team, MLB.Team, NPB_Final_Season, MLB_First_Season)
```

Retrieve Batter Manifest

```
library(fs)

file_list <- dir_ls("npb_data", glob = "*.txt")

npb_full <- file_list %>%
  map_dfr(function(file_path) {
    read_csv(file_path, show_col_types = FALSE) %>%
      mutate(Season = as.numeric(str_extract(path_file(file_path), "\\d{4}")))
  })

npb_transition <- npb_full %>%
  mutate(Name = str_trim(Name)) %>%
  inner_join(manifest, by = c("Name" = "Player", "Season" = "NPB_Final_Season")) %>%
  mutate(League = "NPB") %>%
  select(Name, Season, League, PA, SO, BB, HR)

mlb_transition <- mlb %>%
  mutate(Name = str_trim(Name)) %>%
  inner_join(manifest, by = c("Name" = "Player")) %>%
  filter(SEASON >= MLB_First_Season) %>%
  mutate(League = "MLB") %>%
  rename(Season = SEASON) %>%
  select(Name, Season, League, PA, SO, BB, HR)

main_transition <- bind_rows(npb_transition, mlb_transition)


tsutsugo_raw <- read.csv("yoshi_tsutsugo.txt")

tsutsugo_clean <- tsutsugo_raw %>%
  filter(!Team %in% c("3TM", "2TM")) %>%
  mutate(
    Name = "Yoshi Tsutsugo",
    League = "MLB"
  ) %>%
  select(Name, Season, League, PA, SO, BB, HR)


batter_transition_data <- bind_rows(main_transition, tsutsugo_clean) %>%
  arrange(Name, Season) %>%
  mutate(
    K_Rate = SO / PA,
    BB_Rate = BB / PA,
    HR_Rate = HR / PA
  )
batter_transition_data_final <- batter_transition_data %>%
  group_by(Name, Season, League) %>%
  summarise(
    PA = sum(PA, na.rm = TRUE),
    SO = sum(SO, na.rm = TRUE),
    BB = sum(BB, na.rm = TRUE),
```

```r
    HR = sum(HR, na.rm = TRUE),
    .groups = "drop"
  ) %>%
  mutate(
    K_Rate = SO / PA,
    BB_Rate = BB / PA,
    HR_Rate = HR / PA
  )
```

Retrieve Pitcher Manifest

```r
pitcher_manifest <- manifest %>%
  filter(!Player %in% unique(batter_transition_data_final$Name))

pitcher_list <- dir_ls("pitchers", glob = "*.txt")

raw_pitcher_data <- pitcher_list %>%
  map_dfr(function(file_path) {
    player_name <- path_file(file_path) %>%
      str_remove("\\.txt$") %>%
      str_replace_all("_", " ")

    read_csv(file_path, show_col_types = FALSE) %>%
      mutate(
        Year = as.numeric(Year),
        Name = player_name
      )
  })

pitcher_transition_data <- raw_pitcher_data %>%
  inner_join(pitcher_manifest, by = c("Name" = "Player")) %>%
  filter(
    (Year == NPB_Final_Season & Lev == "Fgn" & !Lg %in% c("JPWL", "JPEL")) |
    (Year >= MLB_First_Season & Lev == "Maj")
  ) %>%
  filter(!Tm %in% c("2 Teams", "3 Teams")) %>%
  mutate(
    League = if_else(Lev == "Fgn", "NPB", "MLB"),
    Season = Year
  ) %>%
  select(Name, Season, League, ERA, IP, SO, BB, WHIP) %>%
  arrange(Name, Season)


pitcher_transition_final <- pitcher_transition_data %>%
  mutate(
    Outs = floor(IP) * 3 + round((IP %% 1) * 10)
  ) %>%
  group_by(Name, Season, League) %>%
  summarise(
    Total_Outs = sum(Outs, na.rm = TRUE),
    SO = sum(SO, na.rm = TRUE),
    BB = sum(BB, na.rm = TRUE),
    Estimated_ER = sum((ERA * IP) / 9 * IP, na.rm = TRUE) / sum(IP, na.rm = TRUE) * 9,
    ERA = weighted.mean(ERA, Outs, na.rm = TRUE),
    WHIP = weighted.mean(WHIP, Outs, na.rm = TRUE),
    .groups = "drop"
  ) %>%
  mutate(
    IP = floor(Total_Outs / 3) + (Total_Outs %% 3) / 10,
    K_9 = (SO * 9) / (Total_Outs / 3),
    BB_9 = (BB * 9) / (Total_Outs / 3)
  ) %>%
  select(Name, Season, League, IP, ERA, WHIP, SO, BB, K_9, BB_9)
```

wOBA

```
woba_weights <- tibble(
  Season = 2015:2025,
  wBB = c(0.687, 0.691, 0.693, 0.690, 0.690, 0.699, 0.692, 0.689, 0.696, 0.689, 0.691),
  wHBP = c(0.718, 0.721, 0.723, 0.720, 0.719, 0.728, 0.722, 0.720, 0.726, 0.720, 0.722),
  w1B = c(0.881, 0.878, 0.877, 0.880, 0.870, 0.883, 0.879, 0.884, 0.883, 0.882, 0.882),
  w2B = c(1.256, 1.242, 1.232, 1.247, 1.217, 1.238, 1.242, 1.261, 1.244, 1.254, 1.252),
  w3B = c(1.594, 1.569, 1.552, 1.578, 1.529, 1.558, 1.568, 1.601, 1.569, 1.590, 1.584),
  wHR = c(2.065, 2.015, 1.980, 2.031, 1.940, 1.979, 2.007, 2.072, 2.004, 2.050, 2.037)
)
tsutsugo_for_woba <- tsutsugo_raw %>%
  filter(!Team %in% c("3TM", "2TM")) %>%
  mutate(
    Name = "Yoshi Tsutsugo",
    Season = Season,
    League = "MLB"
  ) %>%
  select(
    Name, Season, PA, H, `X2B`, `X3B`, HR, BB, HBP, IBB
  ) %>%
  rename("2B"=X2B) %>%
  rename("3B"=X3B)

mlb_raw <- read_csv("mlb_stats.csv", show_col_types = FALSE) %>%
  rename(Name = `G...2`) %>%
  rename(Season = SEASON) %>%
  filter(Season >= 2015) %>%
  select(Name, Season, PA, H, `2B`, `3B`, HR, BB, HBP, IBB) %>%
  bind_rows(tsutsugo_for_woba)
```

```
## New names:
## * ‘G‘ -> ‘G...2‘
## * ‘G‘ -> ‘G...6‘
```

```r
mlb_births <- read_csv("MLB_BirthYears.csv", show_col_types = FALSE)



mlb_raw_with_age <- mlb_raw %>%
  mutate(Name = str_trim(Name)) %>%
  left_join(mlb_births, by = "Name") %>%
  mutate(
    Age_Offset = if_else(birthMonth > 6, 1, 0),
    Age = Season - birthYear - Age_Offset
  ) %>%
  filter(!is.na(Age), !is.na(PA)) %>%

  select(-playerID, -birthYear, -birthMonth, -birthDay, -Age_Offset)
```

```
## Warning in left_join(., mlb_births, by = "Name"): Detected an unexpected many-to-many relationship be
## i Row 20 of ‘x‘ matches multiple rows in ‘y‘.
## i Row 2968 of ‘y‘ matches multiple rows in ‘x‘.
## i If a many-to-many relationship is expected, set ‘relationship =
##   "many-to-many"‘ to silence this warning.
```

```r
file_list <- dir_ls("npb_data", glob = "*.txt")
npb_raw <- file_list %>%
  map_dfr(function(file_path) {
    read_csv(file_path, show_col_types = FALSE) %>%
      mutate(Season = as.numeric(str_extract(path_file(file_path), "\\d{4}")))
  }) %>%
  filter(Season >= 2015) # Apply 2015 filter


process_woba <- function(df, league_name) {
  df %>%
    mutate(
      Name = str_trim(Name),
      H1B = H - `2B` - `3B` - HR,
      League = league_name
    ) %>%
    left_join(woba_weights, by = "Season") %>%
    mutate(
      wOBA_Num = (wBB * BB) + (wHBP * HBP) + (w1B * H1B) + (w2B * `2B`) + (w3B * `3B`) + (wHR * HR),
      wOBA_Denom = PA - IBB
    ) %>%
    drop_na(wOBA_Num, wOBA_Denom, Age) %>%
    select(Name, Season, Age, League, PA, wOBA_Num, wOBA_Denom)
}

mlb_woba <- process_woba(mlb_raw_with_age, "MLB")
npb_woba <- process_woba(npb_raw, "NPB")
```

League-Adjusted Comparison (z-scores):

```
library(ggrepel)
```

```
## Warning: package 'ggrepel' was built under R version 4.5.2
```

```
process_woba_fixed <- function(df, league_name) {
  df %>%
    mutate(
      Name = str_trim(Name),
      H1B = H - `2B` - `3B` - HR,
      League = league_name
    ) %>%
    left_join(woba_weights, by = "Season") %>%
    mutate(
      wOBA_Num = (wBB * BB) + (wHBP * HBP) + (w1B * H1B) + (w2B * `2B`) + (w3B * `3B`) + (wHR * HR),
      wOBA_Denom = PA - IBB
    ) %>%
    # FIX: Aggregate Split Seasons (Summing stats by Player/Year)
    group_by(Name, Season, League, Age) %>%
    summarise(
      wOBA_Num = sum(wOBA_Num, na.rm = TRUE),
      wOBA_Denom = sum(wOBA_Denom, na.rm = TRUE),
      .groups = "drop"
    )
}

mlb_woba_clean <- process_woba_fixed(mlb_raw_with_age, "MLB")
npb_woba_clean <- process_woba_fixed(npb_raw, "NPB")

manifest_fixed <- manifest %>%
  mutate(
    Player = case_when(
      Player == "Yoshitomo Tsutsugo" ~ "Yoshi Tsutsugo",
      TRUE ~ Player
    ))

all_player_stats <- bind_rows(mlb_woba_clean, npb_woba_clean) %>%
  inner_join(manifest_fixed, by = c("Name" = "Player")) %>%
  filter(
    (League == "NPB" & Season == NPB_Final_Season) |
    (League == "MLB" & Season >= MLB_First_Season)
  ) %>%
  group_by(Name, Season, League, Age) %>%
  summarise(
    wOBA_Num = sum(wOBA_Num, na.rm = TRUE),
    wOBA_Denom = sum(wOBA_Denom, na.rm = TRUE),
    .groups = "drop"
  ) %>%
  mutate(Player_wOBA = wOBA_Num / wOBA_Denom) %>%
  filter(is.finite(Player_wOBA))
```

```r
regression_data <- all_player_stats %>%
  inner_join(manifest_fixed, by = c("Name" = "Player")) %>%

  filter(
    (League == "NPB" & Season == NPB_Final_Season) |
    (League == "MLB" & Season == MLB_First_Season)
  ) %>%
    group_by(Name, League) %>%
  summarise(
    wOBA = sum(wOBA_Num) / sum(wOBA_Denom),
    Age = max(Age),
    .groups = "drop"
  ) %>%
  pivot_wider(
    names_from = League,
    values_from = c(wOBA, Age)
  ) %>%

  select(
    Name,
    NPB_Final_wOBA = wOBA_NPB,
    MLB_Year1_wOBA = wOBA_MLB,
    Age = Age_MLB
  ) %>%
  filter(!is.na(NPB_Final_wOBA) & !is.na(MLB_Year1_wOBA))

print(regression_data)
```

```
## # A tibble: 4 x 4
##   Name             NPB_Final_wOBA MLB_Year1_wOBA   Age
##   <chr>                     <dbl>          <dbl> <dbl>
## 1 Masataka Yoshida          0.449          0.339    29
## 2 Seiya Suzuki              0.456          0.338    27
## 3 Shohei Ohtani             0.398          0.402    23
## 4 Yoshi Tsutsugo            0.382          0.312    28
```

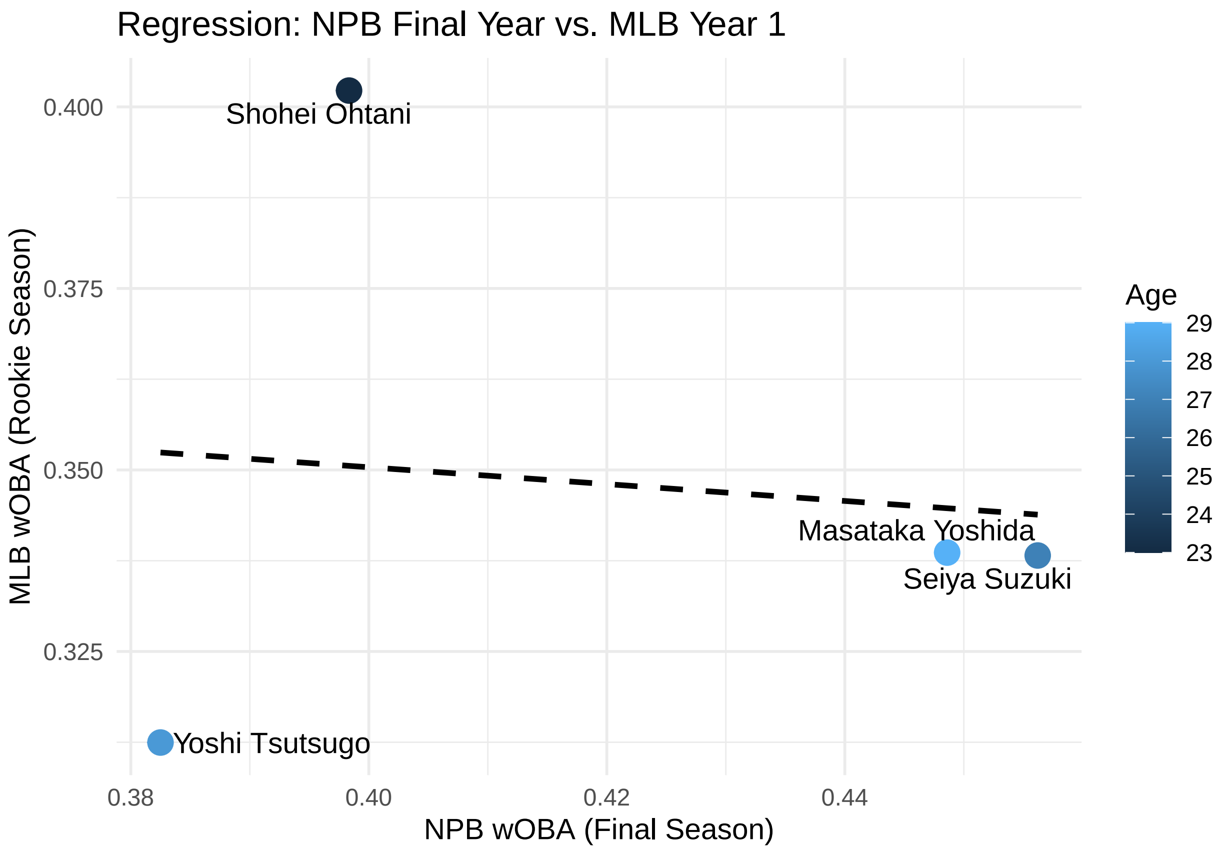
```r
model <- lm(MLB_Year1_wOBA ~ NPB_Final_wOBA + Age, data = regression_data)
summary(model)
```

```
##
## Call:
## lm(formula = MLB_Year1_wOBA ~ NPB_Final_wOBA + Age, data = regression_data)
##
## Residuals:
##         1         2         3         4
##  0.015896 -0.015610  0.006301 -0.006588
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.620098   0.183251   3.384    0.183
## NPB_Final_wOBA  0.274987   0.410576   0.670    0.624
## Age            -0.014508   0.005698  -2.546    0.238
```

```
##
## Residual standard error: 0.02407 on 1 degrees of freedom
## Multiple R-squared:  0.868,  Adjusted R-squared:  0.604
## F-statistic: 3.288 on 2 and 1 DF,  p-value: 0.3633
```

```
# --- Plot ---
ggplot(regression_data, aes(x = NPB_Final_wOBA, y = MLB_Year1_wOBA)) +
  geom_point(aes(color = Age), size = 4) +
  geom_smooth(method = "lm", se = FALSE, linetype = "dashed", color = "black") +
  geom_text_repel(aes(label = Name)) +
  labs(
    title = "Regression: NPB Final Year vs. MLB Year 1",
    x = "NPB wOBA (Final Season)",
    y = "MLB wOBA (Rookie Season)"
  ) +
  theme_minimal()
```

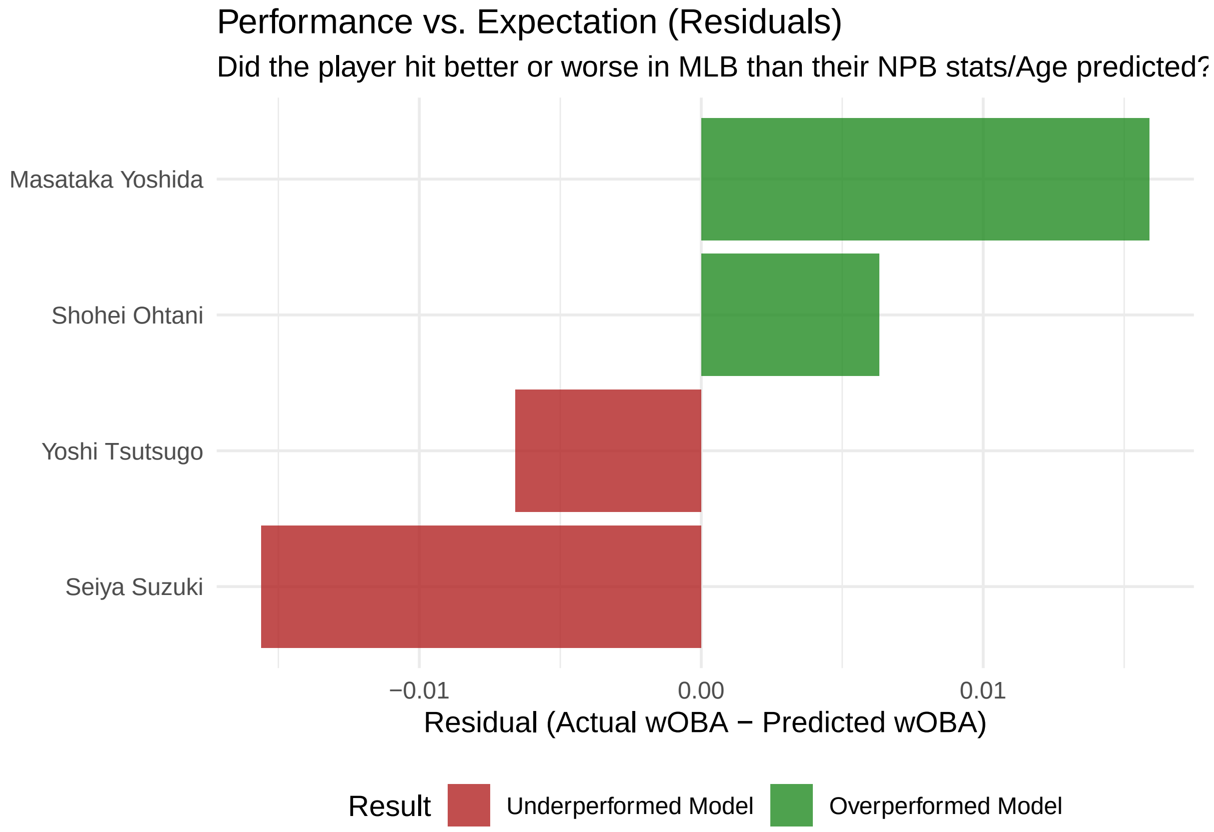
```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```



## Residuals

```
residuals_df <- regression_data %>%
  mutate(
    Predicted_wOBA = predict(model, newdata = .),
    Residual = MLB_Year1_wOBA - Predicted_wOBA
  )
```

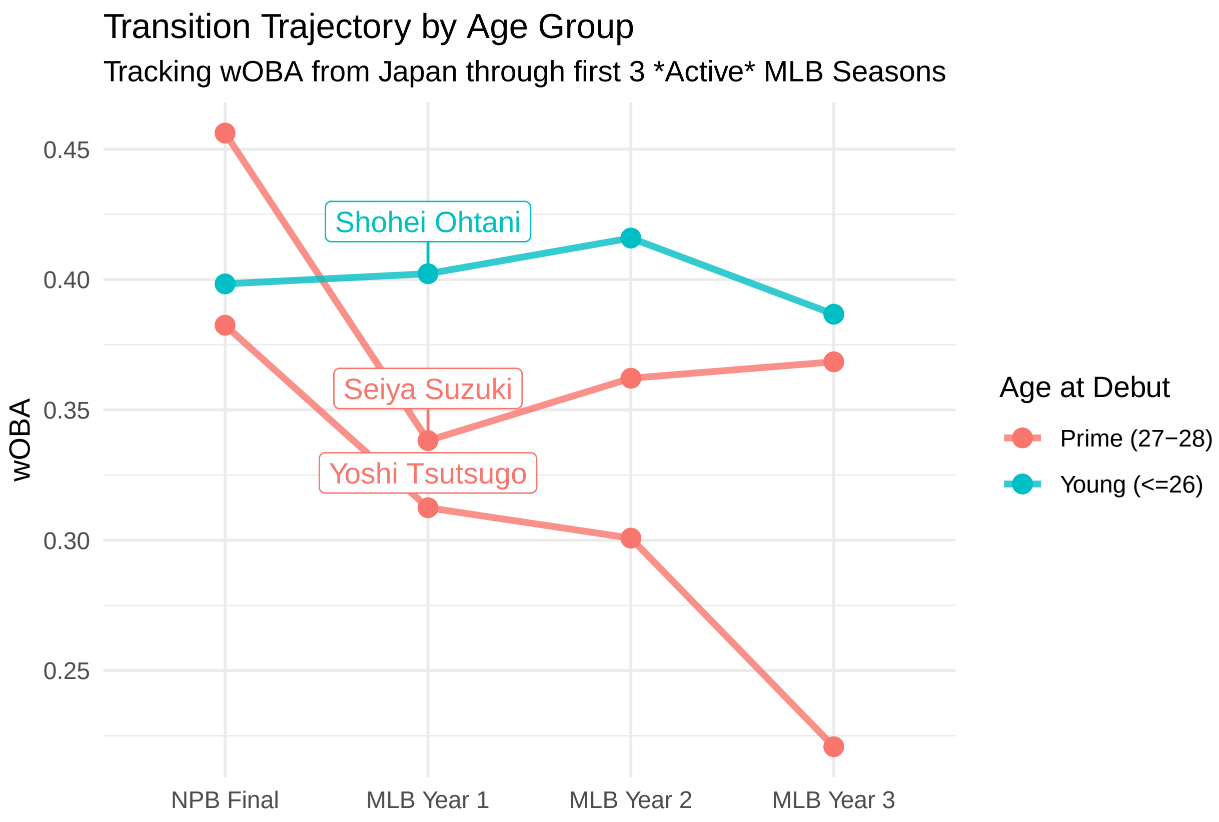
```r
ggplot(residuals_df, aes(x = reorder(Name, Residual), y = Residual, fill = Residual > 0)) +
  geom_col(alpha = 0.8) +
  coord_flip() +
  scale_fill_manual(
    values = c("firebrick", "forestgreen"),
    labels = c("Underperformed Model", "Overperformed Model")
  ) +
  labs(
    title = "Performance vs. Expectation (Residuals)",
    subtitle = "Did the player hit better or worse in MLB than their NPB stats/Age predicted?",
    y = "Residual (Actual wOBA - Predicted wOBA)",
    x = "",
    fill = "Result"
  ) +
  theme_minimal() +
  theme(legend.position = "bottom")
```



## Time Adjustment

```r
long_term_names <- all_player_stats %>%
  filter(League == "MLB") %>%
  group_by(Name) %>%
  filter(n() >= 3) %>%
  pull(Name) %>%
  unique()
```

```r
time_plot_df <- all_player_stats %>%
  filter(Name %in% long_term_names) %>%
  group_by(Name, League) %>%
  arrange(Season) %>%
  mutate(Season_Rank = row_number()) %>%
  group_by(Name) %>%

  mutate(
    Stage = case_when(
      League == "NPB" ~ "NPB Final",  # <--- Fixed missing "~" here
      League == "MLB" & Season_Rank == 1 ~ "MLB Year 1",
      League == "MLB" & Season_Rank == 2 ~ "MLB Year 2",
      League == "MLB" & Season_Rank == 3 ~ "MLB Year 3",
      TRUE ~ NA_character_
    ),
    Rookie_Age = min(Age[League == "MLB"], na.rm = TRUE)
  ) %>%

  filter(!is.na(Stage)) %>%
  ungroup() %>%

  mutate(
    Age_Group = case_when(
      Rookie_Age <= 26 ~ "Young (<=26)",
      Rookie_Age >= 29 ~ "Veteran (29+)",
      TRUE ~ "Prime (27-28)"
    ),
    Stage = factor(Stage, levels = c("NPB Final", "MLB Year 1", "MLB Year 2", "MLB Year 3"))
  )


ggplot(time_plot_df, aes(x = Stage, y = Player_wOBA, group = Name, color = Age_Group)) +
  geom_line(linewidth = 1.2, alpha = 0.8) +
  geom_point(size = 3) +
  geom_label_repel(
    data = time_plot_df %>% filter(Stage == "MLB Year 1"),
    aes(label = Name),
    nudge_y = 0.02,
    show.legend = FALSE
  ) +
  labs(
    title = "Transition Trajectory by Age Group",
    subtitle = "Tracking wOBA from Japan through first 3 *Active* MLB Seasons",
    y = "wOBA",
    x = "",
    color = "Age at Debut"
  ) +
  theme_minimal()
```


Transition Trajectory by Age Group
Tracking wOBA from Japan through first 3 *Active* MLB Seasons
wOBA
0.45
0.40
0.35
0.30
0.25
NPB Final
MLB Year 1
MLB Year 2
MLB Year 3
Shohei Ohtani
Seiya Suzuki
Yoshi Tsutsugo
Age at Debut
Prime (27−28)
Young (<=26)