STA_160FP_part1

Johnson Tian

2024-06-06

```
base_path <- "/Users/johnson/Library/Mobile Documents/com~apple~CloudDocs/UCDavis/2024 Spring/STA 160/"

player <- read.csv(paste0(base_path, "player.csv"))
player_17 <- read.csv(paste0(base_path, "player_17.csv"))
player_18 <- read.csv(paste0(base_path, "player_18.csv"))
player_19 <- read.csv(paste0(base_path, "player_19.csv"))
player_20 <- read.csv(paste0(base_path, "player_20.csv"))
player_21 <- read.csv(paste0(base_path, "player_21.csv"))
daily_avg_speed <- read.csv(paste0(base_path, "daily_avg_speed.csv"))
```

```
player_pfx <- player %>%
  group_by(game_date) %>%
  summarise(
    `Average horizontal movement` = mean(pfx_x, na.rm = TRUE),
    `Average vertical movement` = mean(pfx_z, na.rm = TRUE)) %>%
  ungroup() %>%
  melt(id=c("game_date")) %>%
  mutate(Year = as.factor(substr(game_date,1,4)))

player_avg_yr <- player %>%
  group_by(Year) %>%
  summarise(
    pfx_x = round(mean(pfx_x, na.rm = TRUE),2),
    pfx_z = round(mean(pfx_z, na.rm = TRUE),2))


player_avg_yr
```

```
## # A tibble: 5 x 3
##    Year pfx_x pfx_z
##   <int> <dbl> <dbl>
## 1  2017  0.3   0.61
## 2  2018  0.35  0.5
## 3  2019  0.48  0.59
## 4  2020  0.42  0.66
## 5  2021  0.44  0.71
```

```
# updated functions in the package `baseballr`

csv_from_url <- function(...){
  data.table::fread(...)
}
#-------------------------------------------------------------------------------
```

```r
make_baseballr_data <- function(df, type, timestamp){
  out <- df %>%
    tidyr::as_tibble()

  class(out) <- c("baseballr_data","tbl_df","tbl","data.table","data.frame")
  attr(out,"baseballr_timestamp") <- timestamp
  attr(out,"baseballr_type") <- type
  return(out)
}
# @title
statcast_search <- function(start_date = Sys.Date() - 1, end_date = Sys.Date(),
                                  playerid = NULL,
                                  player_type = "batter", ...) {
  # Check for other user errors.
  if (start_date <= "2015-03-01") { # March 1, 2015 was the first date of Spring Training.
    message("Some metrics such as Exit Velocity and Batted Ball Events have only been compiled since 20
  }
  if (start_date < "2008-03-25") { # March 25, 2008 was the first date of the 2008 season.
    stop("The data are limited to the 2008 MLB season and after.")
    return(NULL)
  }
  if (start_date == Sys.Date()) {
    message("The data are collected daily at 3 a.m. Some of today's games may not be included.")
  }
  if (start_date > as.Date(end_date)) {
    stop("The start date is later than the end date.")
    return(NULL)
  }

  playerid_var <- ifelse(player_type == "pitcher",
                         "pitchers_lookup%5B%5D", "batters_lookup%5B%5D")

  vars <- tibble::tribble(
    ~var, ~value,
    "all", "true",
    "hfPT", "",
    "hfAB", "",
    "hfBBT", "",
    "hfPR", "",
    "hfZ", "",
    "stadium", "",
    "hfBBL", "",
    "hfNewZones", "",
    "hfGT", "R%7CPO%7CS%7C&hfC",
    "hfSea", paste0(lubridate::year(start_date), "%7C"),
    "hfSit", "",
    "hfOuts", "",
    "opponent", "",
    "pitcher_throws", "",
    "batter_stands", "",
    "hfSA", "",
    "player_type", player_type,
    "hfInfield", "",
```

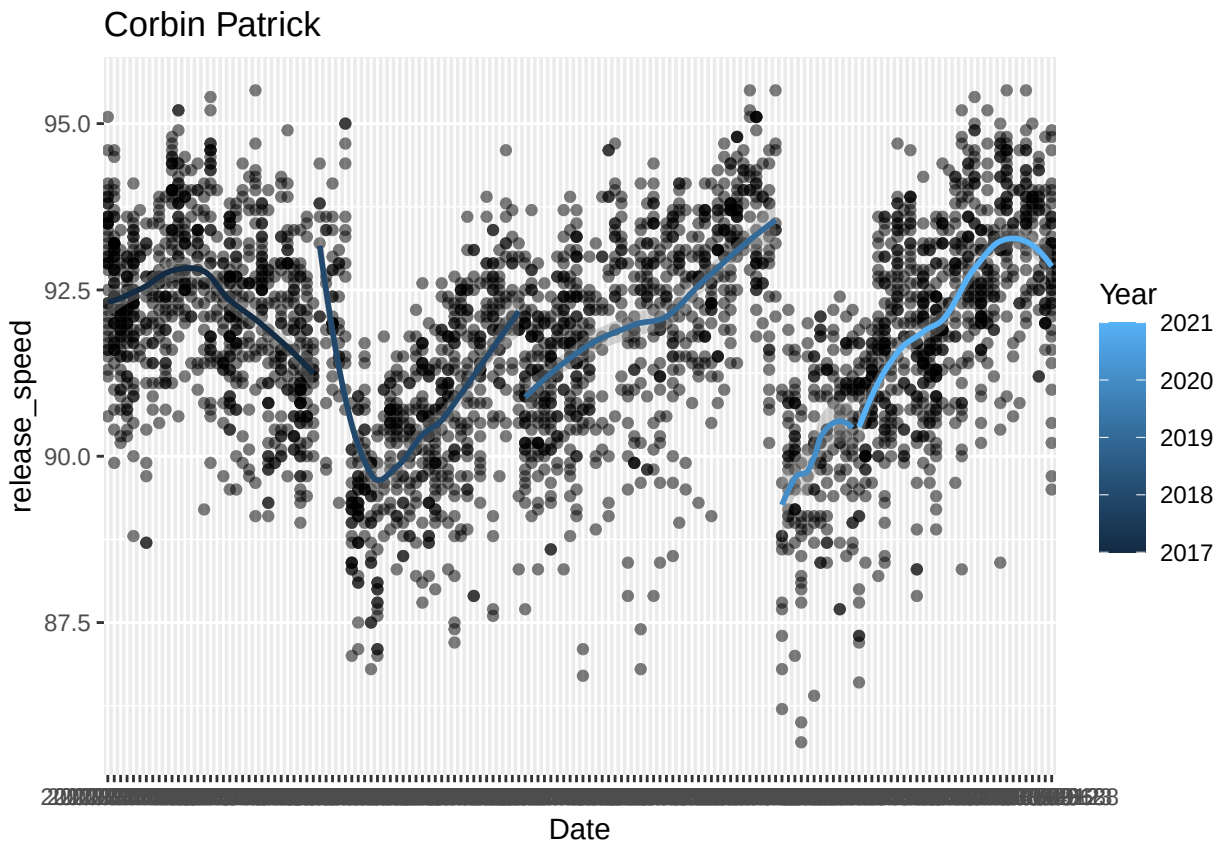
```
  "team", "",
  "position", "",
  "hfOutfield", "",
  "hfRO", "",
  "home_road", "",
  playerid_var, ifelse(is.null(playerid), "", as.character(playerid)),
  "game_date_gt", as.character(start_date),
  "game_date_lt", as.character(end_date),
  "hfFlag", "",
  "hfPull", "",
  "metric_1", "",
  "hfInn", "",
  "min_pitches", "0",
  "min_results", "0",
  "group_by", "name",
  "sort_col", "pitches",
  "player_event_sort", "h_launch_speed",
  "sort_order", "desc",
  "min_abs", "0",
  "type", "details") %>%
  dplyr::mutate(pairs = paste0(.data$var, "=", .data$value))

if (is.null(playerid)) {
  # message("No playerid specified. Collecting data for all batters/pitchers.")
  vars <- vars %>%
    dplyr::filter(!grepl("lookup", .data$var))
}

url_vars <- paste0(vars$pairs, collapse = "&")
url <- paste0("https://baseballsavant.mlb.com/statcast_search/csv?", url_vars)
# message(url)

# Do a try/catch to show errors that the user may encounter while downloading.
tryCatch(
  {
    suppressMessages(
      suppressWarnings(
        payload <- csv_from_url(url, encoding ="UTF-8")
      )
    )
  },
  error = function(cond) {
    message(cond)
    stop("No payload acquired")
  },
  # this will never run??
  warning = function(cond) {
    message(cond)
  }
)
# returns 0 rows on failure but > 1 columns
if (nrow(payload) > 1) {
```

```
    names(payload) <- c("pitch_type", "game_date", "release_speed", "release_pos_x",
                        "release_pos_z", "player_name", "batter", "pitcher", "events",
                        "description", "spin_dir", "spin_rate_deprecated", "break_angle_deprecated",
                        "break_length_deprecated", "zone", "des", "game_type", "stand",
                        "p_throws", "home_team", "away_team", "type", "hit_location",
                        "bb_type", "balls", "strikes", "game_year", "pfx_x", "pfx_z",
                        "plate_x", "plate_z", "on_3b", "on_2b", "on_1b", "outs_when_up",
                        "inning", "inning_topbot", "hc_x", "hc_y", "tfs_deprecated",
                        "tfs_zulu_deprecated", "fielder_2", "umpire", "sv_id", "vx0",
                        "vy0", "vz0", "ax", "ay", "az", "sz_top", "sz_bot", "hit_distance_sc",
                        "launch_speed", "launch_angle", "effective_speed", "release_spin_rate",
                        "release_extension", "game_pk", "pitcher_1", "fielder_2_1",
                        "fielder_3", "fielder_4", "fielder_5", "fielder_6", "fielder_7",
                        "fielder_8", "fielder_9", "release_pos_y", "estimated_ba_using_speedangle",
                        "estimated_woba_using_speedangle", "woba_value", "woba_denom",
                        "babip_value", "iso_value", "launch_speed_angle", "at_bat_number",
                        "pitch_number", "pitch_name", "home_score", "away_score", "bat_score",
                        "fld_score", "post_away_score", "post_home_score", "post_bat_score",
                        "post_fld_score", "if_fielding_alignment", "of_fielding_alignment",
                        "spin_axis", "delta_home_win_exp", "delta_run_exp", "bat_speed", "swing_length"]
    payload <- process_statcast_payload(payload) %>%
      make_baseballr_data("MLB Baseball Savant Statcast Search data from baseballsavant.mlb.com",Sys.ti
    return(payload)
  } else {
    warning("No valid data found")


# (somewhere within the statcast_search function before the payload is searched for)
colos <- c("pitch_type", "game_date",
           "release_speed", "release_pos_x", "release_pos_z",
           "player_name", "batter", "pitcher",
           "events", "description", "spin_dir",
           "spin_rate_deprecated", "break_angle_deprecated",
           "break_length_deprecated", "zone", "des",
           "game_type", "stand", "p_throws",
           "home_team", "away_team", "type",
           "hit_location", "bb_type", "balls",
           "strikes", "game_year", "pfx_x",
           "pfx_z", "plate_x", "plate_z",
           "on_3b", "on_2b", "on_1b", "outs_when_up",
           "inning", "inning_topbot", "hc_x",
           "hc_y", "tfs_deprecated", "tfs_zulu_deprecated",
           "fielder_2", "umpire", "sv_id",
           "vx0", "vy0", "vz0", "ax",
           "ay", "az", "sz_top", "sz_bot",
           "hit_distance_sc", "launch_speed", "launch_angle",
           "effective_speed", "release_spin_rate",
           "release_extension", "game_pk", "pitcher_1",
           "fielder_2_1", "fielder_3", "fielder_4",
           "fielder_5", "fielder_6", "fielder_7",
           "fielder_8", "fielder_9", "release_pos_y",
           "estimated_ba_using_speedangle", "estimated_woba_using_speedangle",
           "woba_value", "woba_denom", "babip_value",
```

```
            "iso_value", "launch_speed_angle", "at_bat_number",
            "pitch_number", "pitch_name", "home_score",
            "away_score", "bat_score", "fld_score",
            "post_away_score", "post_home_score",
            "post_bat_score", "post_fld_score", "if_fielding_alignment",
            "of_fielding_alignment", "spin_axis",
            "delta_home_win_exp", "delta_run_exp")
colNumber <- ncol(payload)
if(length(colos) != colNumber){
  newCols <- paste("newStat", 1:(length(colos) - colNumber))
  colos <- c(colos, newCols)
  message("New stats detected! baseballr will be updated soon to properly identify these stats")
}
# payload is acquired somewhere in here
# when the payload columns need to be named:
names(payload) <- colos



payload <- payload %>%
  make_baseballr_data("MLB Baseball Savant Statcast Search data from baseballsavant.mlb.com",Sys.time()
    return(payload)
  }
}

statcast_search.default <- function(start_date = Sys.Date() - 1, end_date = Sys.Date(),
                                    playerid = NULL, player_type = "batter", ...) {

  message(paste0(start_date, " is not a date. Attempting to coerce..."))
  start_Date <- as.Date(start_date)

  tryCatch(
    {
      end_Date <- as.Date(end_date)
    },
    warning = function(cond) {
      message(paste0(end_date, " was not coercible into a date. Using today."))
      end_Date <- Sys.Date()
      message("Original warning message:")
      message(cond)
    }
  )

  statcast_search(start_Date, end_Date,
                          playerid, player_type, ...)

}

statcast_search_batters <- function(start_date, end_date, batterid = NULL, ...) {
  statcast_search(start_date, end_date, playerid = batterid,
                          player_type = "batter", ...)
}
```

```
statcast_search_pitchers <- function(start_date, end_date, pitcherid = NULL, ...) {
  statcast_search(start_date, end_date, playerid = pitcherid,
                  player_type = "pitcher", ...)
}
```

```
player %>%
  filter(pitch_type=="FF") %>%
  ggplot(aes(game_date, release_speed)) +
  geom_point(alpha = 0.5) +
  stat_smooth(aes(group = Year, color = Year),formula = y~x, method='loess')+
  ggtitle("Corbin Patrick") +
  ylab("release_speed") +
  xlab("Date")
```



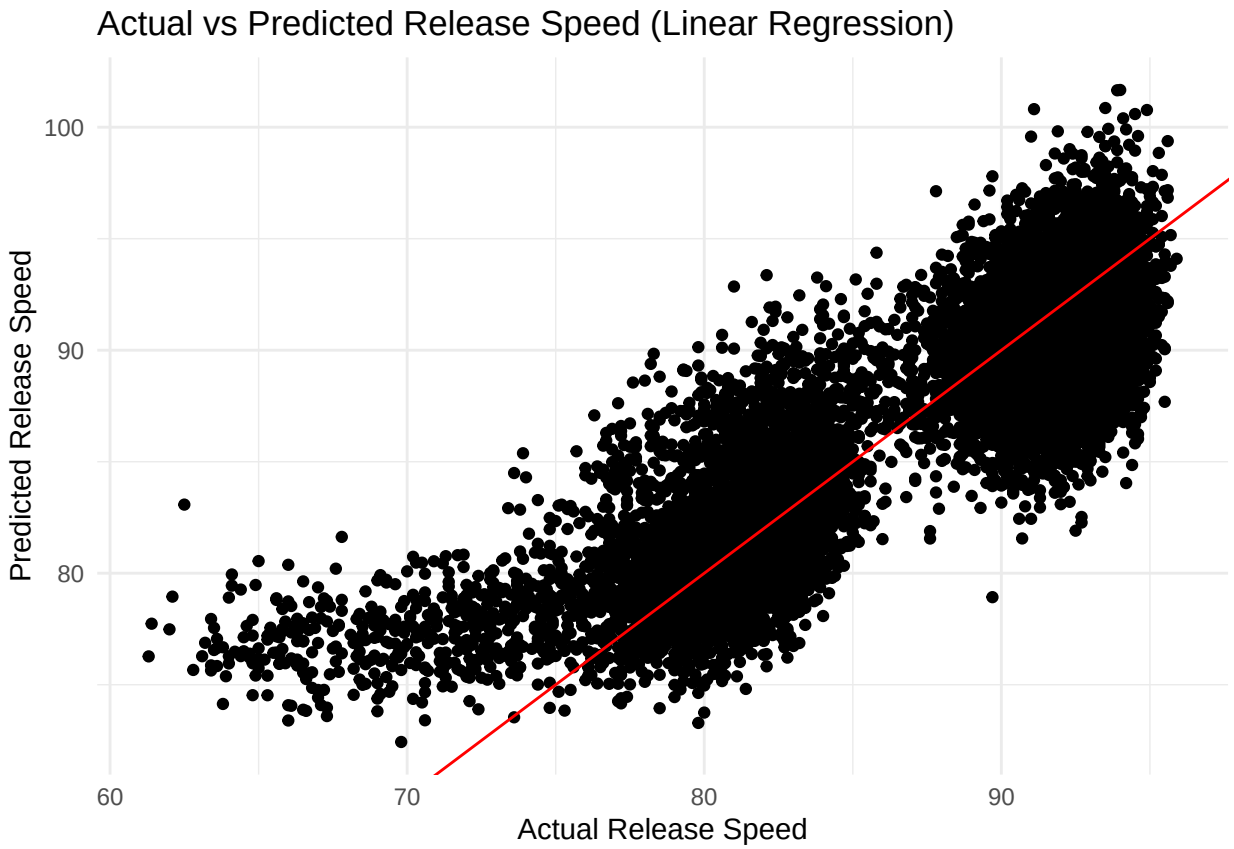

```
player <- player %>%
  drop_na(release_speed, release_pos_x, release_pos_z, pfx_x, pfx_z, vx0, vy0, vz0, ax, ay, az)

player_linear_model_combined <- player %>%
  mutate(
    release_pos = release_pos_x * release_pos_z,
    pfx_interact = pfx_x * pfx_z,
    vx_vy_interact = vx0 * vy0,
    ax_ay_az_interact = ax * ay * az
  )
```

```
#linear regression
```

```r
release_speed_linear_model <- lm(release_speed ~ release_pos + pfx_interact + vx_vy_interact + vz0 + ax

summary(release_speed_linear_model)
```

```
##
## Call:
## lm(formula = release_speed ~ release_pos + pfx_interact + vx_vy_interact +
##     vz0 + ax_ay_az_interact, data = player_linear_model_combined)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -20.5766  -1.5878   0.2149   1.9341  10.7736
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         9.219e+01  3.998e-01  230.62   <2e-16 ***
## release_pos        -1.081e+00  2.534e-02  -42.66   <2e-16 ***
## pfx_interact        2.698e+00  9.244e-02   29.19   <2e-16 ***
## vx_vy_interact      6.650e-03  1.127e-04   59.01   <2e-16 ***
## vz0                -2.627e-01  1.378e-02  -19.07   <2e-16 ***
## ax_ay_az_interact -4.308e-04  1.112e-05  -38.74   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.083 on 13718 degrees of freedom
## Multiple R-squared:  0.7568, Adjusted R-squared:  0.7567
## F-statistic:  8539 on 5 and 13718 DF,  p-value: < 2.2e-16
```

```r
player_linear_model_combined$predicted_release_speed <- predict(release_speed_linear_model, player_linea

ggplot(player_linear_model_combined, aes(x = release_speed, y = predicted_release_speed)) +
  geom_point() +
  geom_abline(slope = 1, intercept = 0, color = "red") +
  labs(title = "Actual vs Predicted Release Speed (Linear Regression)",
       x = "Actual Release Speed",
       y = "Predicted Release Speed") +
  theme_minimal()
```



lm(formula = release_speed ~ release_pos + pfx_interact + vx_vy_interact + ax_ay_az_interact + vz0, data = player_linear_model_pure_combined)

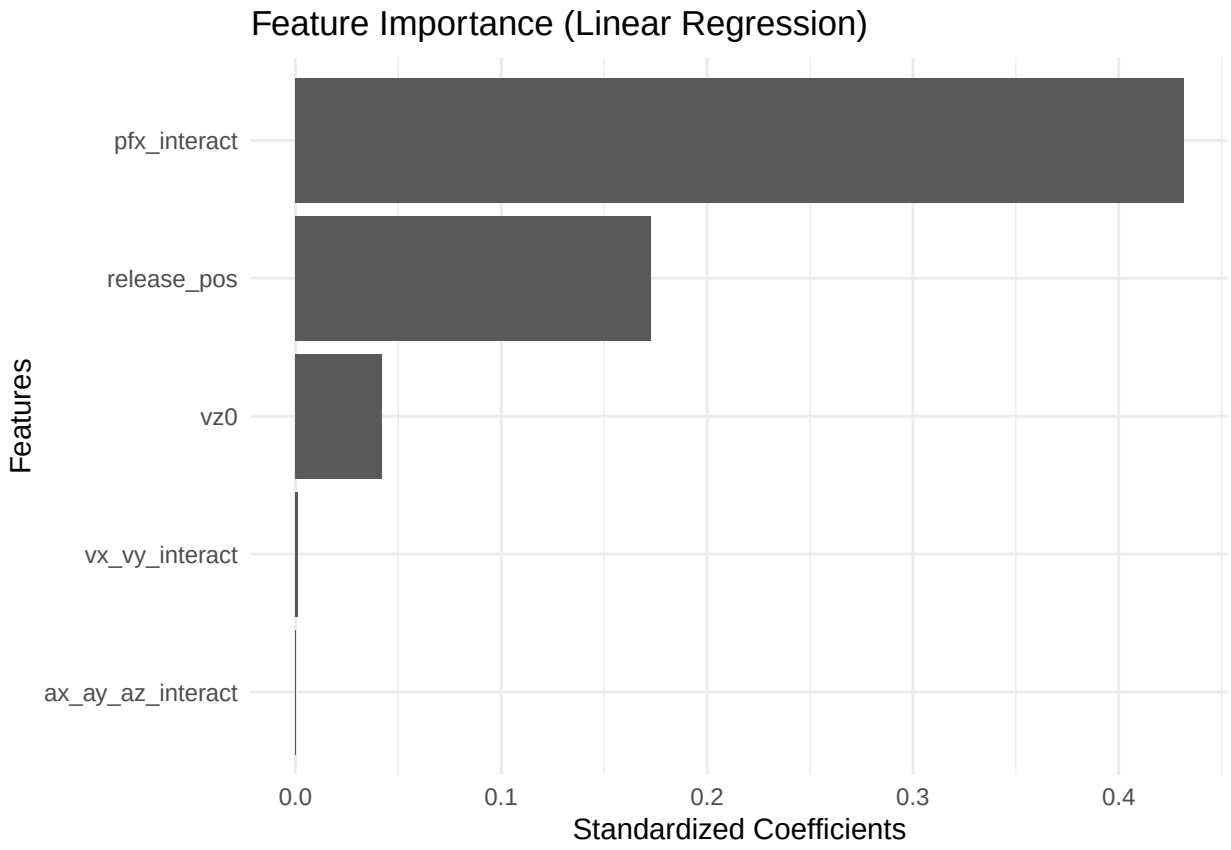
```
player <- player %>%
  drop_na(release_speed, release_pos_x, release_pos_z, pfx_x, pfx_z, vx0, vy0, vz0, ax, ay, az)

player <- player %>%
  mutate(
    release_pos = release_pos_x * release_pos_z,
    pfx_interact = pfx_x * pfx_z,
    vx_vy_interact = vx0 * vy0,
    ax_ay_az_interact = ax * ay * az
  )

player_linear_model_pure_combined <- player %>%
  select(release_speed, release_pos, pfx_interact, vx_vy_interact, ax_ay_az_interact, vz0)
```

```
summary(player_linear_model_pure_combined)
```

```
##  release_speed    release_pos      pfx_interact     vx_vy_interact
##  Min.   :61.30   Min.   : 9.906   Min.   :-1.0925   Min.   :-152.9
##  1st Qu.:81.50   1st Qu.:13.781   1st Qu.: 0.0156   1st Qu.: 640.8
##  Median :88.90   Median :14.520   Median : 0.6161   Median : 833.2
##  Mean   :86.41   Mean   :14.490   Mean   : 0.5734   Mean   : 842.9
##  3rd Qu.:92.00   3rd Qu.:15.236   3rd Qu.: 1.0353   3rd Qu.:1037.3
##  Max.   :95.90   Max.   :18.404   Max.   : 2.8036   Max.   :1978.6
##  ax_ay_az_interact       vz0
##  Min.   :-18487.5   Min.   :-14.504
```

```
##  1st Qu.: -6698.0   1st Qu.: -7.057
##  Median : -3397.3   Median : -5.494
##  Mean   : -3064.4   Mean   : -5.373
##  3rd Qu.:   772.6   3rd Qu.: -3.807
##  Max.   : 12837.3   Max.   :  5.143
```

```r
standardized_coefficients <- coef(summary(release_speed_linear_model))[, "Estimate"] / sd(player_linear
names(standardized_coefficients) <- rownames(coef(summary(release_speed_linear_model)))

importance_df <- data.frame(
  Feature = names(standardized_coefficients)[-1],
  Importance = abs(standardized_coefficients[-1])
)

importance_df <- importance_df %>%
  arrange(desc(Importance))

ggplot(importance_df, aes(x = reorder(Feature, Importance), y = Importance)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  labs(title = "Feature Importance (Linear Regression)",
       x = "Features",
       y = "Standardized Coefficients") +
  theme_minimal()
```

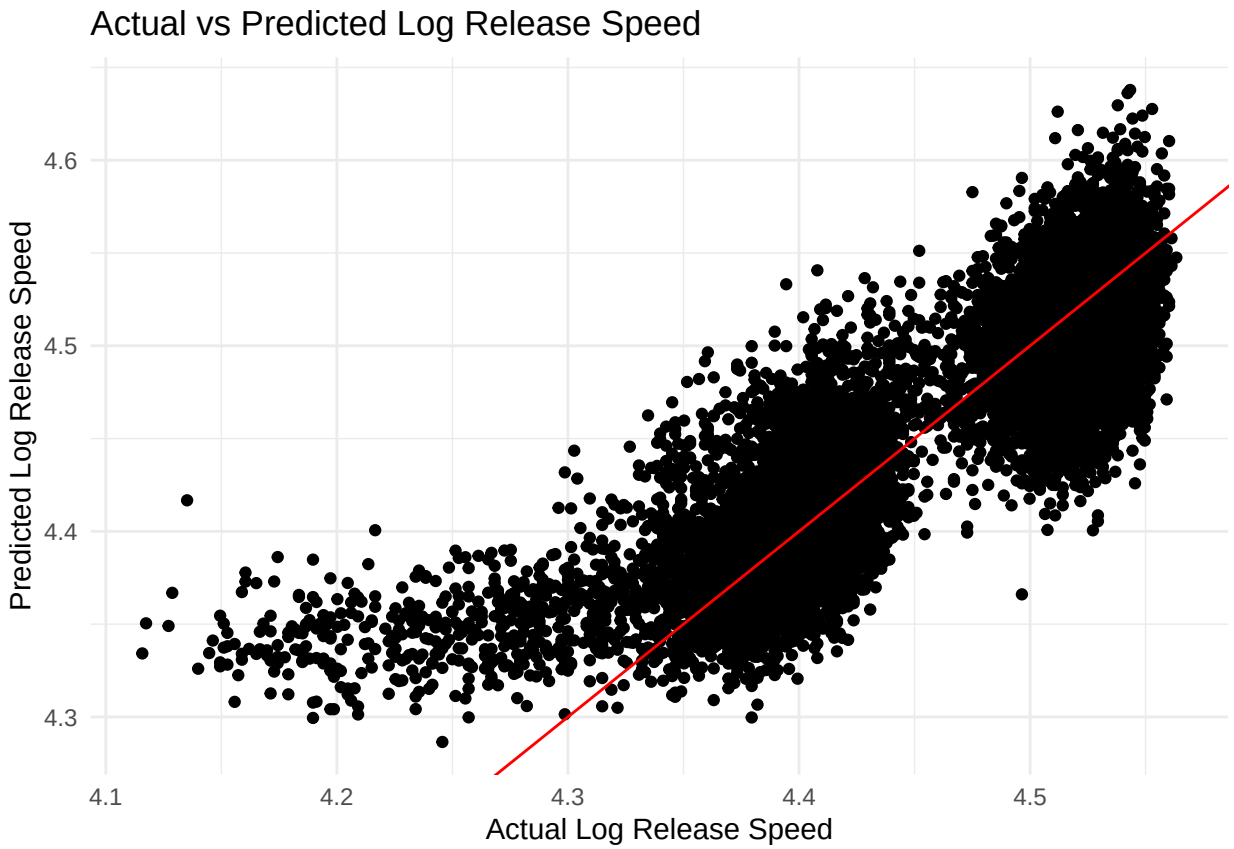



```r
#log regression
log_release_speed_linear_model <- lm(log(release_speed) ~ release_pos + pfx_interact + vx_vy_interact +
```

```r
summary(log_release_speed_linear_model)
```

```
##
## Call:
## lm(formula = log(release_speed) ~ release_pos + pfx_interact +
##     vx_vy_interact + ax_ay_az_interact + vz0, data = player_linear_model_pure_combined)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.281568 -0.018383  0.003081  0.023232  0.130421
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        4.526e+00  4.892e-03  925.09   <2e-16 ***
## release_pos       -1.299e-02  3.101e-04  -41.87   <2e-16 ***
## pfx_interact       3.144e-02  1.131e-03   27.79   <2e-16 ***
## vx_vy_interact     7.928e-05  1.379e-06   57.49   <2e-16 ***
## ax_ay_az_interact -4.859e-06  1.361e-07  -35.70   <2e-16 ***
## vz0               -3.571e-03  1.686e-04  -21.18   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03773 on 13718 degrees of freedom
## Multiple R-squared:  0.7434, Adjusted R-squared:  0.7433
## F-statistic:  7949 on 5 and 13718 DF,  p-value: < 2.2e-16
```

```r
player_linear_model_pure_combined$predicted_log_release_speed <- predict(log_release_speed_linear_model

ggplot(player_linear_model_pure_combined, aes(x = log(release_speed), y = predicted_log_release_speed))
  geom_point() +
  geom_abline(slope = 1, intercept = 0, color = "red") +
  labs(title = "Actual vs Predicted Log Release Speed",
       x = "Actual Log Release Speed",
       y = "Predicted Log Release Speed") +
  theme_minimal()
```




```
#random forest
player <- player %>%
  drop_na(release_speed, release_pos_x, release_pos_z, pfx_x, pfx_z, vx0, vy0, vz0, ax, ay, az)

player <- player %>%
  mutate(
    release_pos = release_pos_x * release_pos_z,
    pfx_interact = pfx_x * pfx_z,
    vx_vy_interact = vx0 * vy0,
    ax_ay_az_interact = ax * ay * az
  )

player_linear_model_pure_combined <- player %>%
  select(release_speed, release_pos, pfx_interact, vx_vy_interact, ax_ay_az_interact, vz0)
set.seed(123)
rf_model_player_linear_model_pure_combined <- randomForest(release_speed ~ ., data = player_linear_model

print(rf_model_player_linear_model_pure_combined)
```

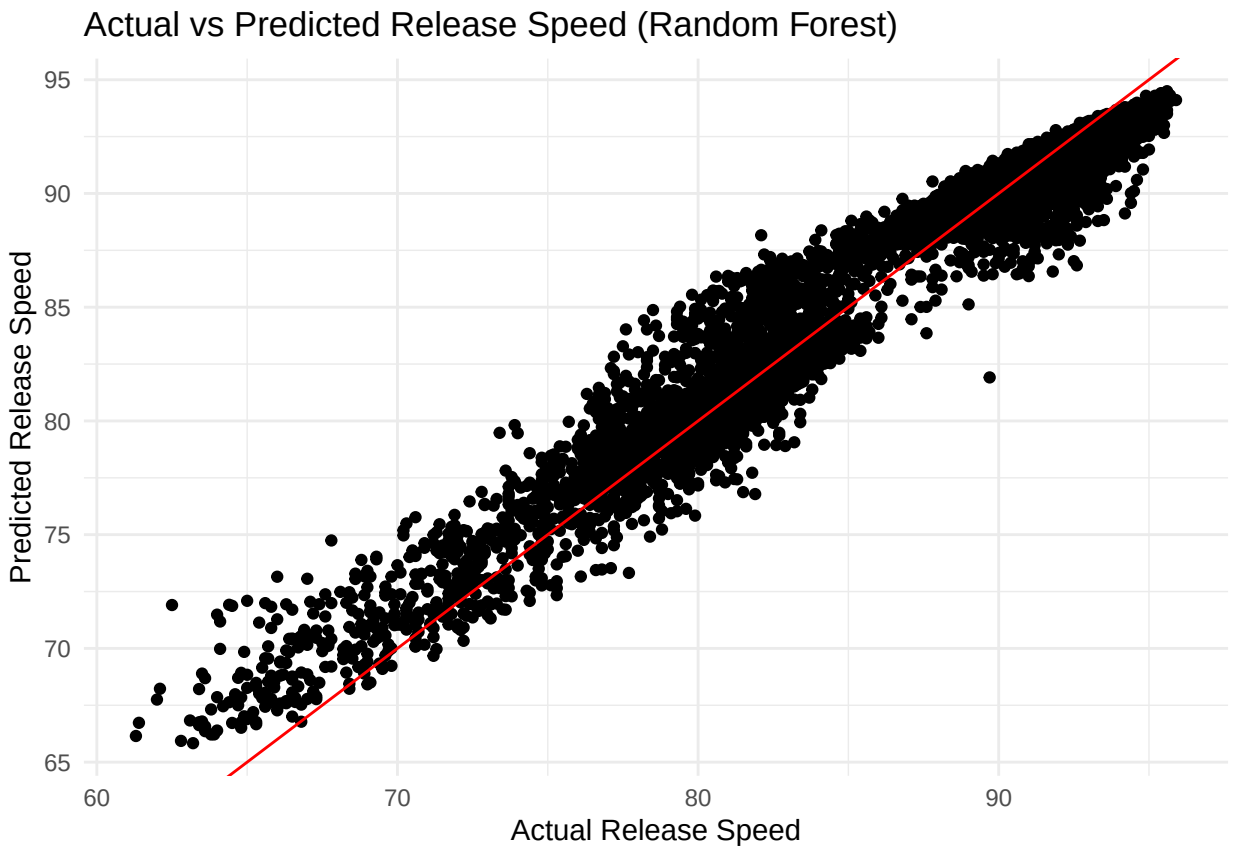
```
##
## Call:
##  randomForest(formula = release_speed ~ ., data = player_linear_model_pure_combined,      ntree = 50
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 1
##
##           Mean of squared residuals: 6.19884
```

```
##                       % Var explained: 84.14
```

```
importance(rf_model_player_linear_model_pure_combined)
```

```
##                     %IncMSE IncNodePurity
## release_pos        88.31804      51541.91
## pfx_interact       58.06925     132176.52
## vx_vy_interact     94.91077      96447.94
## ax_ay_az_interact 59.18214     177586.28
## vz0                57.73520      66248.31
```

```
player_linear_model_pure_combined$predicted_release_speed_rf <- predict(rf_model_player_linear_model_pu

ggplot(player_linear_model_pure_combined, aes(x = release_speed, y = predicted_release_speed_rf)) +
  geom_point() +
  geom_abline(slope = 1, intercept = 0, color = "red") +
  labs(title = "Actual vs Predicted Release Speed (Random Forest)",
       x = "Actual Release Speed",
       y = "Predicted Release Speed") +
  theme_minimal()
```



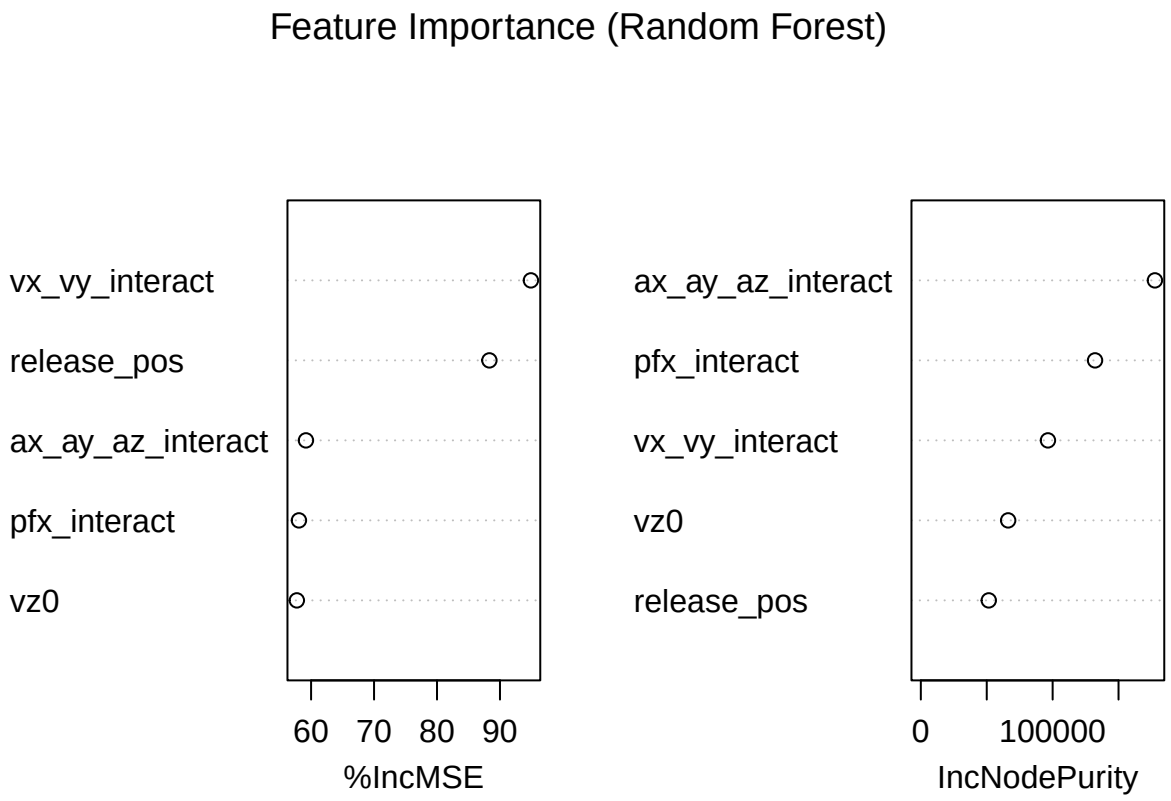

```
mse_rf <- mean((player_linear_model_pure_combined$release_speed - player_linear_model_pure_combined$pred
rmse_rf <- sqrt(mse_rf)

cat("Random Forest Model - MSE:", mse_rf, "\n")
```

```
## Random Forest Model - MSE: 1.433921
```

```r
cat("Random Forest Model - RMSE:", rmse_rf, "\n")
```

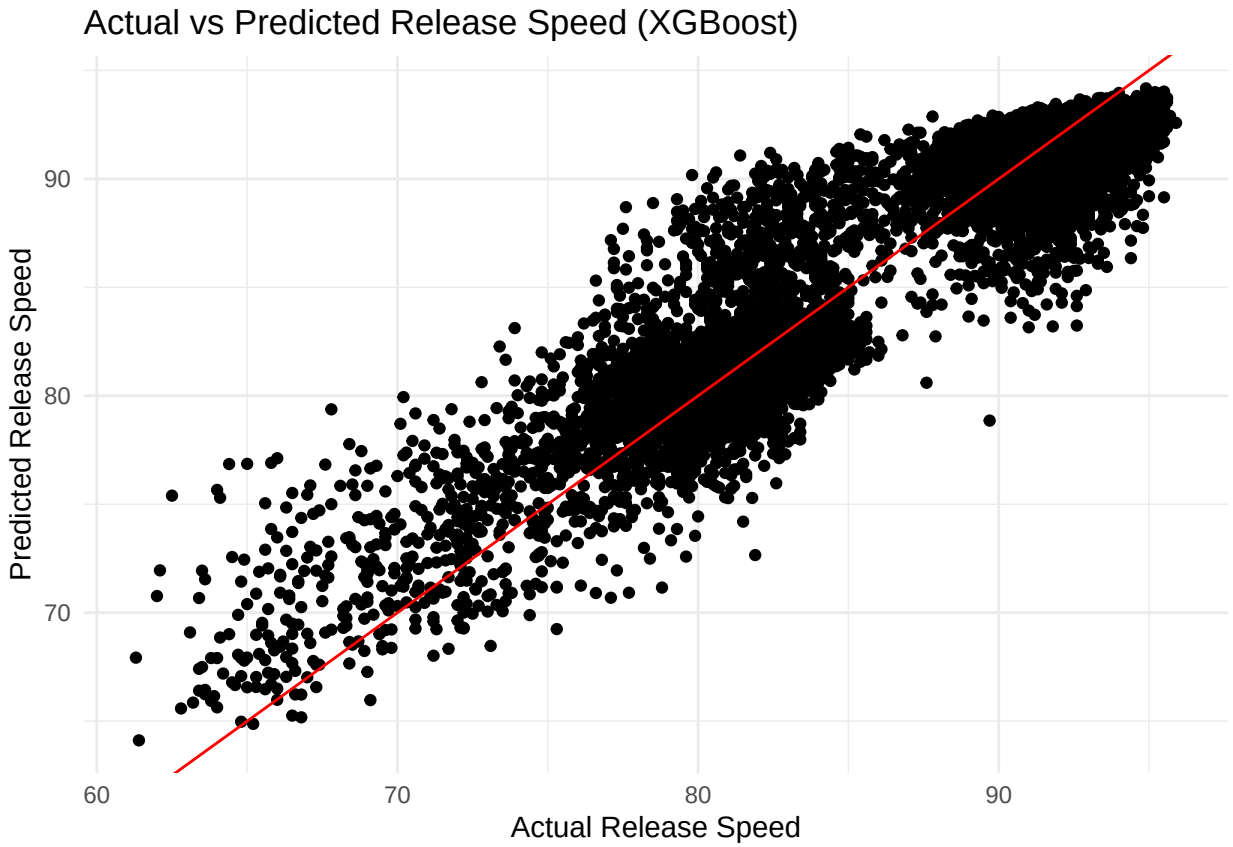
```
## Random Forest Model - RMSE: 1.197464
```

```r
importance_rf <- importance(rf_model_player_linear_model_pure_combined)
varImpPlot(rf_model_player_linear_model_pure_combined, main = "Feature Importance (Random Forest)")
```




```r
#xgboost
player <- player %>%
  drop_na(release_speed, release_pos_x, release_pos_z, pfx_x, pfx_z, vx0, vy0, vz0, ax, ay, az)

player <- player %>%
  mutate(
    release_pos = release_pos_x * release_pos_z,
    pfx_interact = pfx_x * pfx_z,
    vx_vy_interact = vx0 * vy0,
    ax_ay_az_interact = ax * ay * az
  )

player_linear_model_pure_combined <- player %>%
  select(release_speed, release_pos, pfx_interact, vx_vy_interact, ax_ay_az_interact, vz0)
train_matrix <- as.matrix(player_linear_model_pure_combined %>% select(-release_speed))
train_label <- player_linear_model_pure_combined$release_speed

dtrain <- xgb.DMatrix(data = train_matrix, label = train_label)

params <- list(
  objective = "reg:squarederror",
  eval_metric = "rmse",
```

```
  max_depth = 6,
  eta = 0.1,
  subsample = 0.7,
  colsample_bytree = 0.7
)
set.seed(123)
xgb_model <- xgb.train(params = params, data = dtrain, nrounds = 100)

player_linear_model_pure_combined$predicted_release_speed_xgb <- predict(xgb_model, train_matrix)

ggplot(player_linear_model_pure_combined, aes(x = release_speed, y = predicted_release_speed_xgb)) +
  geom_point() +
  geom_abline(slope = 1, intercept = 0, color = "red") +
  labs(title = "Actual vs Predicted Release Speed (XGBoost)",
       x = "Actual Release Speed",
       y = "Predicted Release Speed") +
  theme_minimal()
```



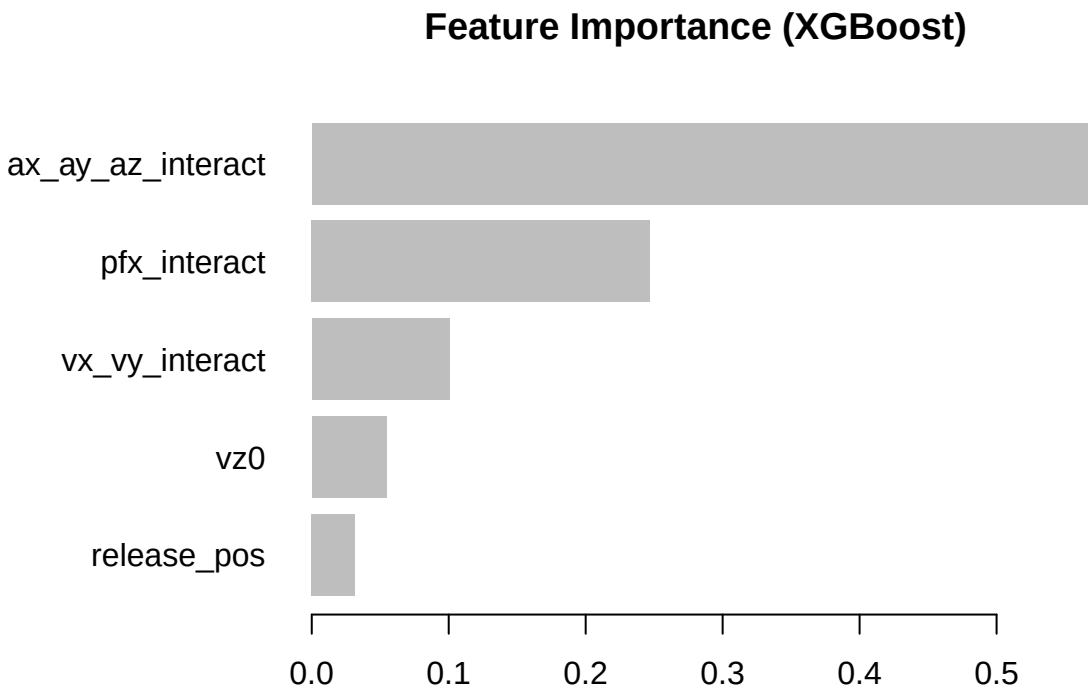

```
mse_xgb <- mean((player_linear_model_pure_combined$release_speed - player_linear_model_pure_combined$pre
rmse_xgb <- sqrt(mse_xgb)

cat("XGBoost Model - MSE:", mse_xgb, "\n")
```

```
## XGBoost Model - MSE: 4.568579
```

```
cat("XGBoost Model - RMSE:", rmse_xgb, "\n")
```

```
## XGBoost Model - RMSE: 2.137424
```

```r
importance_xgb <- xgb.importance(model = xgb_model)
xgb.plot.importance(importance_xgb, main = "Feature Importance (XGBoost)")
```




```r
print(xgb_model)
```

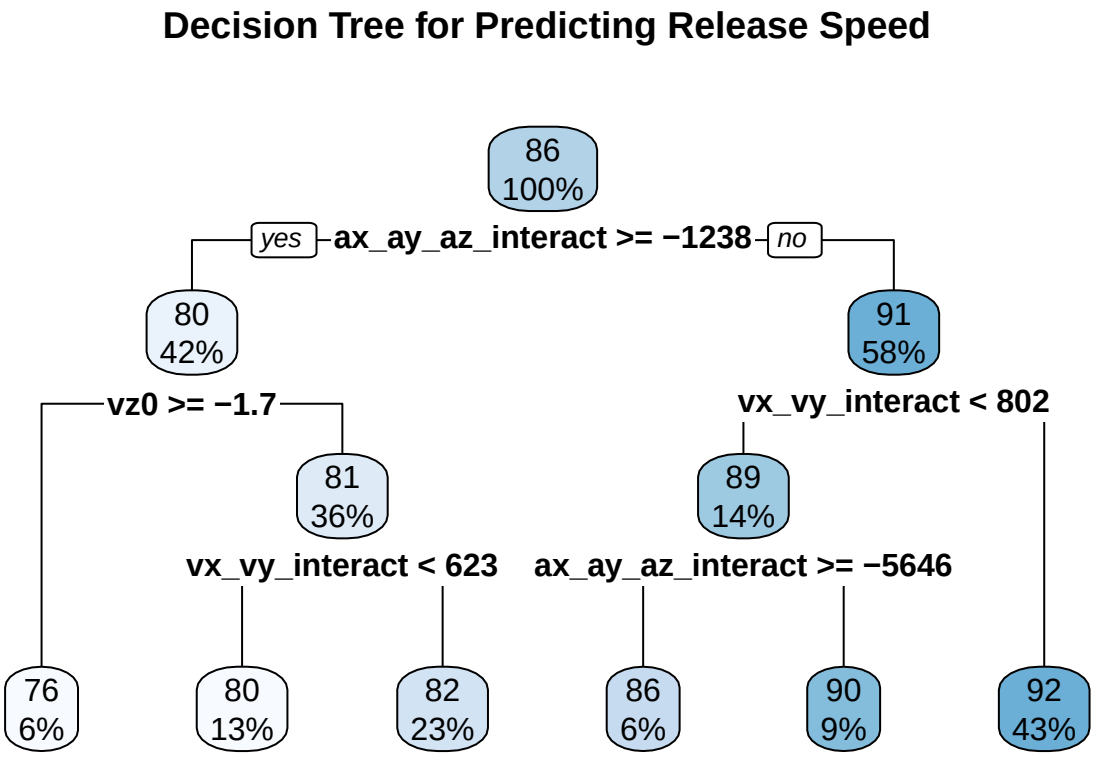
```
## ##### xgb.Booster
## raw: 330.2 Kb
## call:
##   xgb.train(params = params, data = dtrain, nrounds = 100)
## params (as set within xgb.train):
##   objective = "reg:squarederror", eval_metric = "rmse", max_depth = "6", eta = "0.1", subsample = "0
## xgb.attributes:
##   niter
## callbacks:
##   cb.print.evaluation(period = print_every_n)
## # of features: 5
## niter: 100
## nfeatures : 5
```

```r
#decision Tree
player_linear_model_pure_combined <- player %>%
  select(release_speed, release_pos, pfx_interact, vx_vy_interact, ax_ay_az_interact, vz0)
set.seed(123)
dt_model <- rpart(release_speed ~ ., data = player_linear_model_pure_combined, method = "anova")

print(dt_model)
```

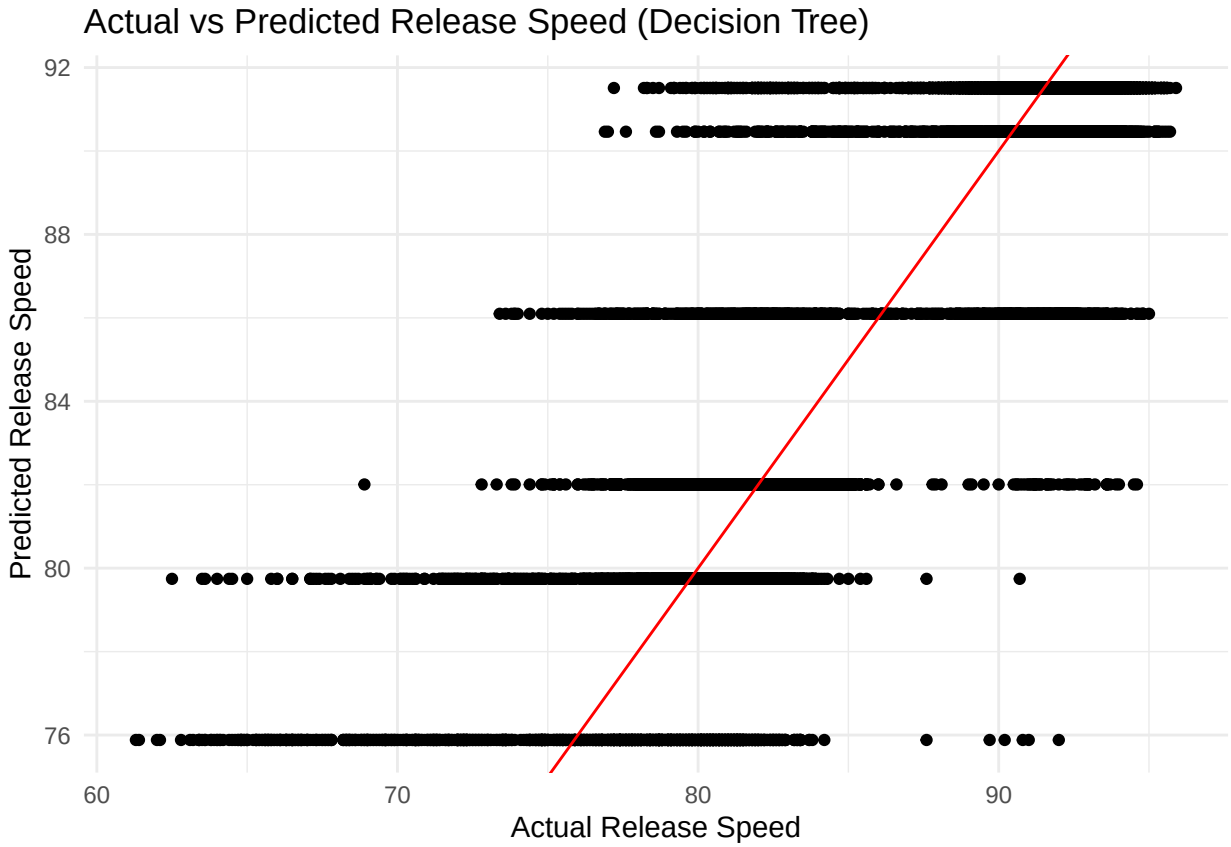
```
## n= 13724
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 13724 536309.50 86.41330
```

```
##    2) ax_ay_az_interact>=-1238.113 5800  84427.91 80.40181
##      4) vz0>=-1.724912 883  26524.46 75.88233 *
##      5) vz0< -1.724912 4917  36628.67 81.21342
##       10) vx_vy_interact< 622.5903 1725  16909.26 79.74452 *
##       11) vx_vy_interact>=622.5903 3192  13986.02 82.00724 *
##    3) ax_ay_az_interact< -1238.113 7924  88863.61 90.81343
##      6) vx_vy_interact< 802.3471 1962  44766.32 88.69791
##       12) ax_ay_az_interact>=-5646.338 793  24339.29 86.09748 *
##       13) ax_ay_az_interact< -5646.338 1169  11426.90 90.46193 *
##      7) vx_vy_interact>=802.3471 5962  32426.92 91.50961 *
```

```
rpart.plot(dt_model, main = "Decision Tree for Predicting Release Speed")
```




```
predicted_release_speed_dt <- predict(dt_model, player_linear_model_pure_combined)

ggplot(player_linear_model_pure_combined, aes(x = release_speed, y = predicted_release_speed_dt)) +
  geom_point() +
  geom_abline(slope = 1, intercept = 0, color = "red") +
  labs(title = "Actual vs Predicted Release Speed (Decision Tree)",
       x = "Actual Release Speed",
       y = "Predicted Release Speed") +
  theme_minimal()
```


Actual vs Predicted Release Speed (Decision Tree)
Predicted Release Speed
92
88
84
80
76
60
70
80
90
Actual Release Speed