project_hanyiw

2024-04-19

```
library(nflverse)
```

```
## Warning: package 'nflverse' was built under R version 4.3.3
```

```
## -- Attaching packages --------------------------------------- nflverse 1.0.3 --
```

```
## v nflfastR 4.6.1     v nflreadr 1.4.0
## v nflseedR 1.2.0     v nflplotR 1.3.1
## v nfl4th   1.0.4
```

```
## Warning: package 'nflfastR' was built under R version 4.3.3
```

```
## Warning: package 'nflseedR' was built under R version 4.3.3
```

```
## Warning: package 'nfl4th' was built under R version 4.3.3
```

```
## Warning: package 'nflreadr' was built under R version 4.3.3
```

```
## Warning: package 'nflplotR' was built under R version 4.3.3
```

```
## ------------------------------------------------------------------ Ready to go! --
```

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
```

```
## -- 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 errors
```

```
library(broom)
library(glmnet)
```

```
## Warning: package 'glmnet' was built under R version 4.3.3
```

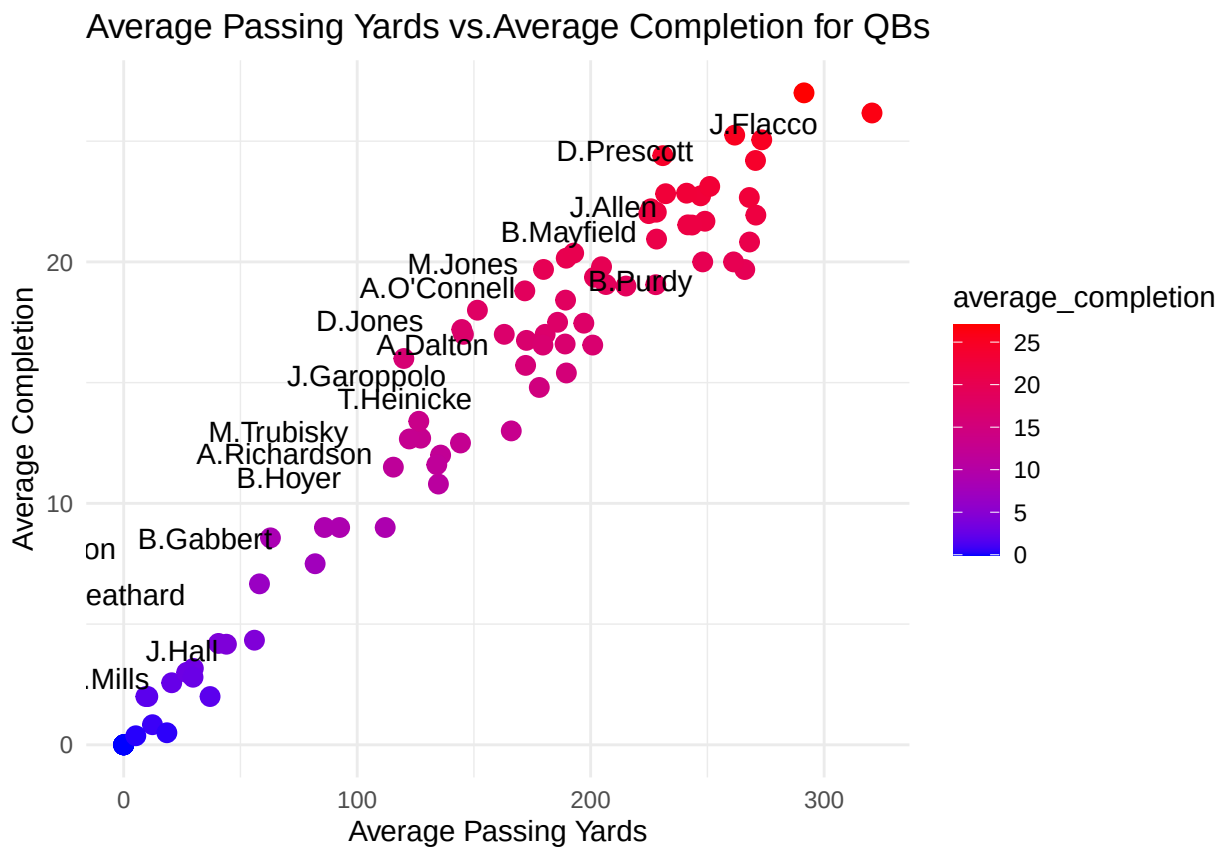
```
## Loading required package: Matrix
##
## Attaching package: 'Matrix'
##
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
##
## Loaded glmnet 4.1-8
```

```
library(ggplot2)


player_stats <- nflreadr::load_player_stats()
player_stats
```

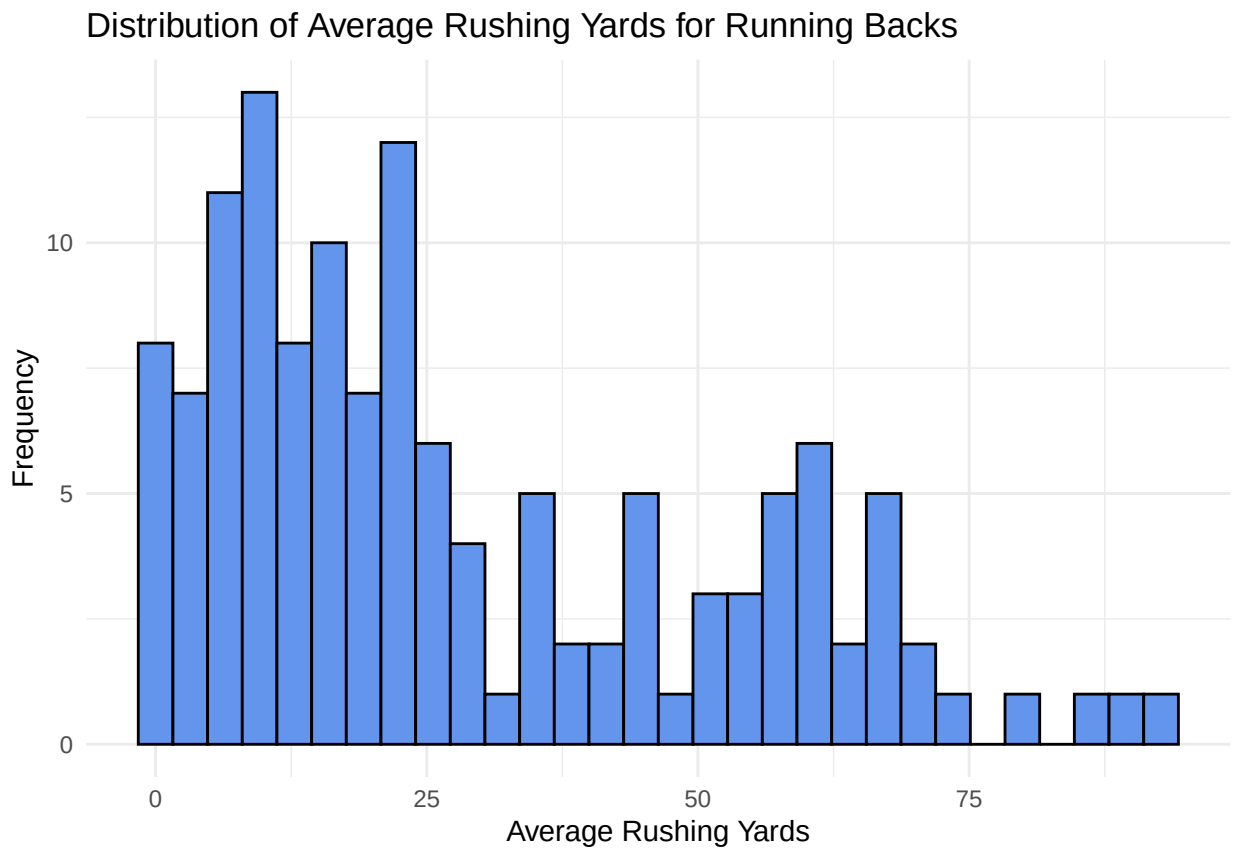
```
## -- nflverse player stats: offense ----------------------------------------------
## i Data updated: 2024-03-11 08:09:54 EDT

## # A tibble: 5,653 x 53
##    player_id  player_name player_display_name position position_group
##    <chr>      <chr>       <chr>               <chr>    <chr>
##  1 00-0023459 A.Rodgers   Aaron Rodgers       QB       QB
##  2 00-0024243 M.Lewis     Marcedes Lewis      TE       TE
##  3 00-0024243 M.Lewis     Marcedes Lewis      TE       TE
##  4 00-0024243 M.Lewis     Marcedes Lewis      TE       TE
##  5 00-0024243 M.Lewis     Marcedes Lewis      TE       TE
##  6 00-0024243 M.Lewis     Marcedes Lewis      TE       TE
##  7 00-0026158 J.Flacco    Joe Flacco          QB       QB
##  8 00-0026158 J.Flacco    Joe Flacco          QB       QB
##  9 00-0026158 J.Flacco    Joe Flacco          QB       QB
## 10 00-0026158 J.Flacco    Joe Flacco          QB       QB
## # i 5,643 more rows
## # i 48 more variables: headshot_url <chr>, recent_team <chr>, season <int>,
## #   week <int>, season_type <chr>, opponent_team <chr>, completions <int>,
## #   attempts <int>, passing_yards <dbl>, passing_tds <int>,
## #   interceptions <dbl>, sacks <dbl>, sack_yards <dbl>, sack_fumbles <int>,
## #   sack_fumbles_lost <int>, passing_air_yards <dbl>,
## #   passing_yards_after_catch <dbl>, passing_first_downs <dbl>, ...
```

```
qb_stats <- player_stats %>%
  filter(position == "QB") %>%
  group_by(player_name) %>%
  summarize(average_completion = mean(completions, na.rm = TRUE),
            average_passing_yards = mean(passing_yards, na.rm = TRUE))


# Scatter plot of average touchdowns vs. passing yards
ggplot(qb_stats, aes(x = average_passing_yards, y = average_completion, label = player_name)) +
  geom_point(aes(color = average_completion), size = 3) +
  geom_text(check_overlap = TRUE, hjust = 1.5, vjust = 1) +
  labs(title = "Average Passing Yards vs.Average Completion for QBs",
       x = "Average Passing Yards",
```

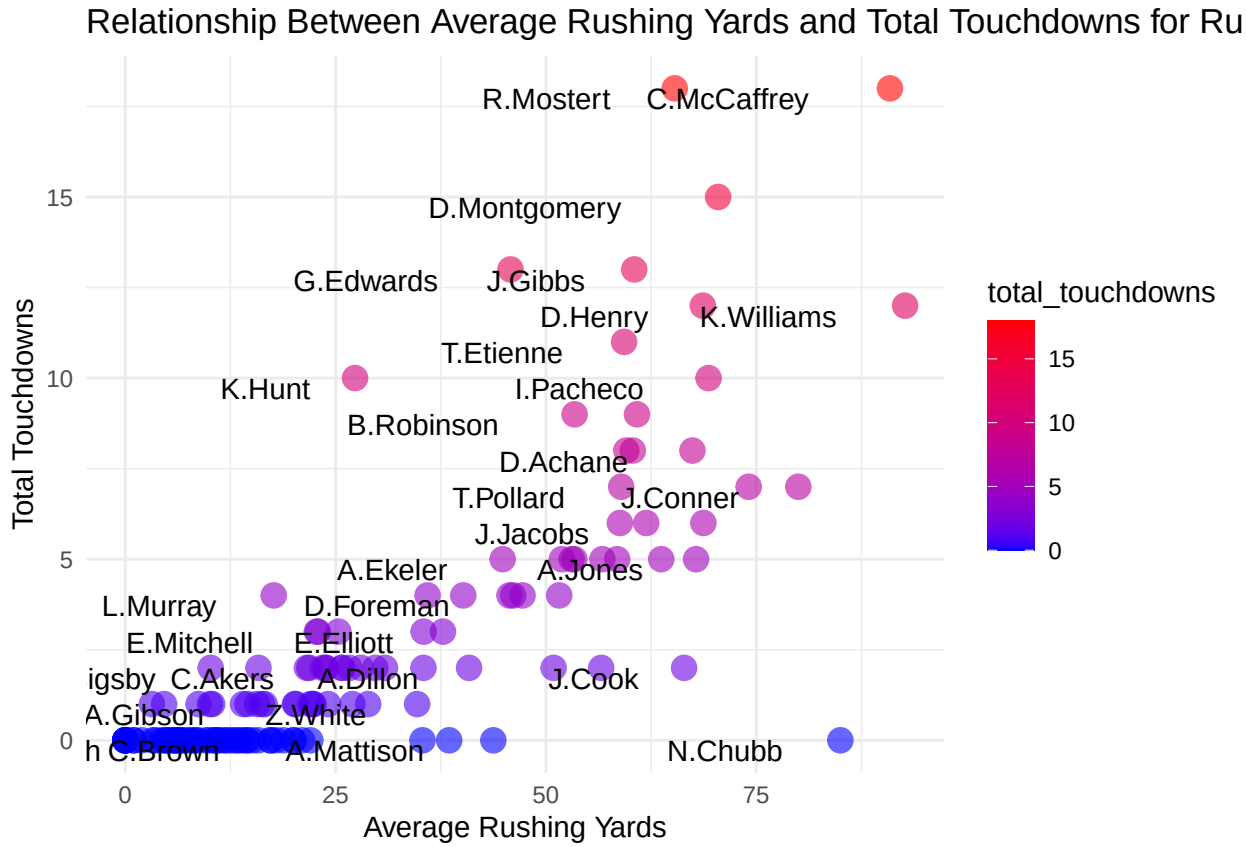
```
       y = "Average Completion") +
  theme_minimal() +
  scale_color_gradient(low = "blue", high = "red")
```




```
rb_stats <- player_stats %>%
  filter(position == "RB") %>%
  group_by(player_name) %>%
  summarize(average_rushing_yards = mean(rushing_yards, na.rm = TRUE),
            total_touchdowns = sum(rushing_tds, na.rm = TRUE))

# Create a histogram of average rushing yards
ggplot(rb_stats, aes(x = average_rushing_yards)) +
  geom_histogram(bins = 30, fill = "cornflowerblue", color = "black") +
  labs(title = "Distribution of Average Rushing Yards for Running Backs",
       x = "Average Rushing Yards",
       y = "Frequency") +
  theme_minimal()
```




```
ggplot(rb_stats, aes(x = average_rushing_yards, y = total_touchdowns,label = player_name)) +
  geom_point(aes(color = total_touchdowns), size = 4, alpha = 0.6) +
  geom_text(check_overlap = TRUE, hjust = 1.5, vjust = 1) +
  scale_color_gradient(low = "blue", high = "red") +
  labs(title = "Relationship Between Average Rushing Yards and Total Touchdowns for Running Backs",
       x = "Average Rushing Yards",
       y = "Total Touchdowns") +
  theme_minimal()
```




```
player_salary <- read.csv("data/salary_Proj.csv")
player_stats <- nflreadr::load_player_stats()

stats_across_weeks <- player_stats %>%
  group_by(player_display_name) %>%
  summarise(across(where(is.numeric), mean),
            position = first(position),
            team = first(recent_team),
            Player = first(player_display_name),
            .groups = 'drop')

inner_join_result <- inner_join(stats_across_weeks, player_salary, by = "Player")
# Using table to see the frequency of each team


pos_counts <- table(inner_join_result$Position)
print(pos_counts)
```

```
##
##  CB  DE  DT  FB   G ILB OLB   P  QB  RB  RT  SS   T  TE  WR
##   2   1   1   8   1   1   2   9  52  47   1   1   1  52  87
```

```
# filter data for each position
QB_data <- subset(inner_join_result, position == 'QB' )
RB_data <- subset(inner_join_result, position == 'RB' )
```

```r
TE_data <- subset(inner_join_result, position == 'TE' )
WR_data <- subset(inner_join_result, position == 'WR' )
```

```r
# WR Preprocessing

# Drop non-related columns(Season,Position,Week,....), columns that is nan
WR_data <- WR_data %>%
  select(-position,-season, -week, -Position, - Player)

WR_data <- WR_data %>%
  select_if(~ !all(is.na(.)))

WR_response <- WR_data  %>%
  select(Base.Salary)

# Standardize each columns
library(scales)
```

```
##
## Attaching package: 'scales'

## The following object is masked from 'package:purrr':
##
##     discard

## The following object is masked from 'package:readr':
##
##     col_factor
```

```r
#WR_data <- WR_data %>%
#  mutate_if(is.numeric, scales::rescale)

# Select data with enough variability to model
WR_data <- WR_data %>%
  select(carries, rushing_yards, rushing_tds, rushing_first_downs, receptions, targets,
         receiving_yards, receiving_tds, receiving_air_yards, receiving_yards_after_catch,
         receiving_first_downs, receiving_epa, racr, target_share, air_yards_share, wopr,
         fantasy_points, fantasy_points_ppr, Base.Salary) %>%
  mutate(across(-Base.Salary, scale))


tibble(WR_data)
```

```
## # A tibble: 87 x 19
##    carries[,1] rushing_yards[,1] rushing_tds[,1] rushing_first_downs[,1]
##          <dbl>             <dbl>           <dbl>                   <dbl>
##  1      -0.541           -0.519          -0.263                  -0.503
##  2      -0.374           -0.358          -0.263                  -0.503
##  3      -0.541           -0.519          -0.263                  -0.503
##  4      -0.541           -0.519          -0.263                  -0.503
##  5      -0.541           -0.519          -0.263                  -0.503
##  6       0.207            0.108          -0.263                   0.421
```

```
##  7       0.463           0.586          -0.263                 0.0131
##  8      -0.541          -0.519          -0.263                -0.503
##  9       0.553           0.0798         -0.263                 0.172
## 10      -0.338          -0.159          -0.263                 0.124
## # i 77 more rows
## # i 15 more variables: receptions <dbl[,1]>, targets <dbl[,1]>,
## #   receiving_yards <dbl[,1]>, receiving_tds <dbl[,1]>,
## #   receiving_air_yards <dbl[,1]>, receiving_yards_after_catch <dbl[,1]>,
## #   receiving_first_downs <dbl[,1]>, receiving_epa <dbl[,1]>, racr <dbl[,1]>,
## #   target_share <dbl[,1]>, air_yards_share <dbl[,1]>, wopr <dbl[,1]>,
## #   fantasy_points <dbl[,1]>, fantasy_points_ppr <dbl[,1]>, ...
```

```
# tibble(WR_response)
```

```
# Linear regression model to see the variation
library(glmnet)

#lm_model <- lm(Base.Salary ~ carries + receptions +targets+receiving_yards+receiving_tds+receiving_air


WR_data <- WR_data %>%
  mutate(across(everything(), ~replace_na(., 0)))

x_matrix <- as.matrix(WR_data[, c("carries", "targets", "receiving_yards", "receiving_tds",
                        "receiving_air_yards", "receiving_first_downs", "receiving_epa", "racr",
                        "target_share", "air_yards_share", "fantasy_points")
])  # Extract predictors and convert to matrix

y_vector <- WR_data$Base.Salary

cv_lasso <- cv.glmnet(x_matrix, y_vector, alpha = 1)

coefficients <- coef(cv_lasso, s = "lambda.min")

print(coefficients)
```

```
## 12 x 1 sparse Matrix of class "dgCMatrix"
##                               s1
## (Intercept)           5011011.5
## carries                796546.9
## targets                       .
## receiving_yards               .
## receiving_tds                 .
## receiving_air_yards           .
## receiving_first_downs 2745132.2
## receiving_epa                 .
## racr                  -317002.3
## target_share                  .
## air_yards_share       1255734.4
## fantasy_points         143492.1
```

```
optimal_lambda <- cv_lasso$lambda.min
optimal_lambda_1se <- cv_lasso$lambda.1se

cat("Optimal lambda (minimizes mean cross-validated error):", optimal_lambda, "\n")
```

```
## Optimal lambda (minimizes mean cross-validated error): 220543.4
```

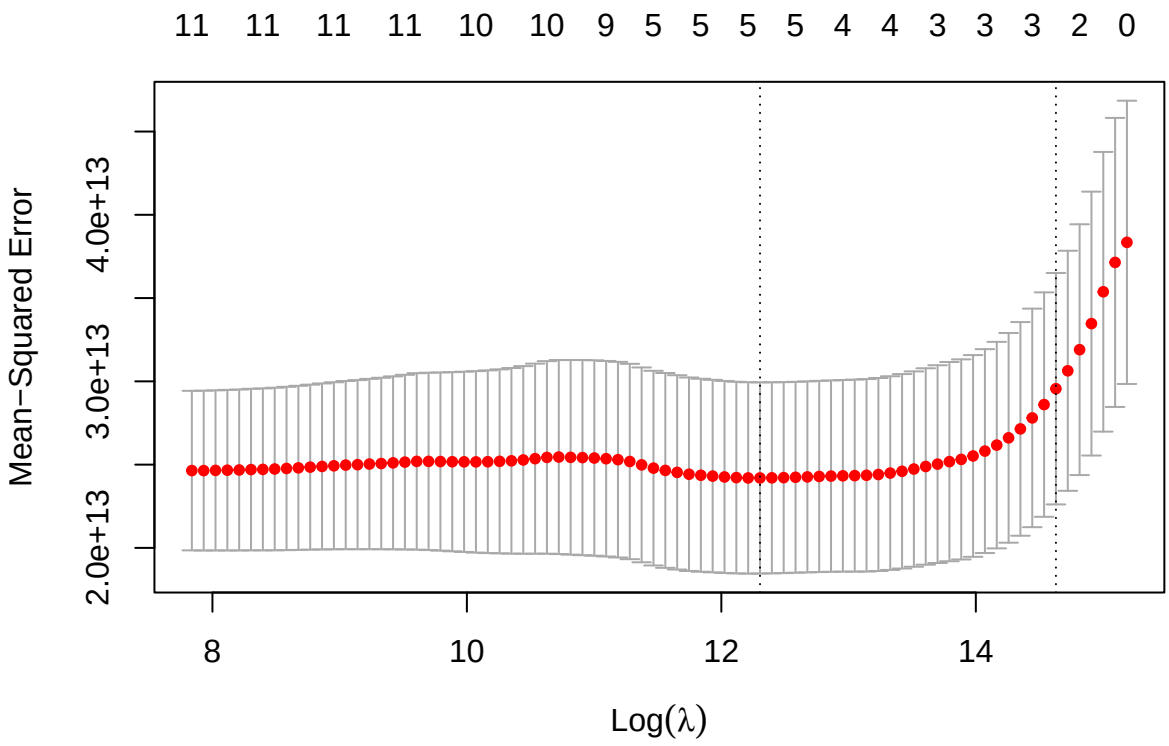
```
cat("Optimal lambda (one standard error rule):", optimal_lambda_1se, "\n")
```

```
## Optimal lambda (one standard error rule): 2257330
```

```
deviance_explained <- cv_lasso$cvm[cv_lasso$lambda == optimal_lambda]
cat("Deviance explained by the model:", deviance_explained, "\n")
```

```
## Deviance explained by the model: 2.420057e+13
```

```
plot(cv_lasso)
```




```
deviance_explained <- cv_lasso$cvm[cv_lasso$lambda == optimal_lambda]
cat("Deviance explained by the model:", deviance_explained, "\n")
```

```
## Deviance explained by the model: 2.420057e+13
```

```r
predicted_salaries <- predict(cv_lasso, newx = x_matrix, s = "lambda.min")
predicted_sd <- sd(predicted_salaries)
```

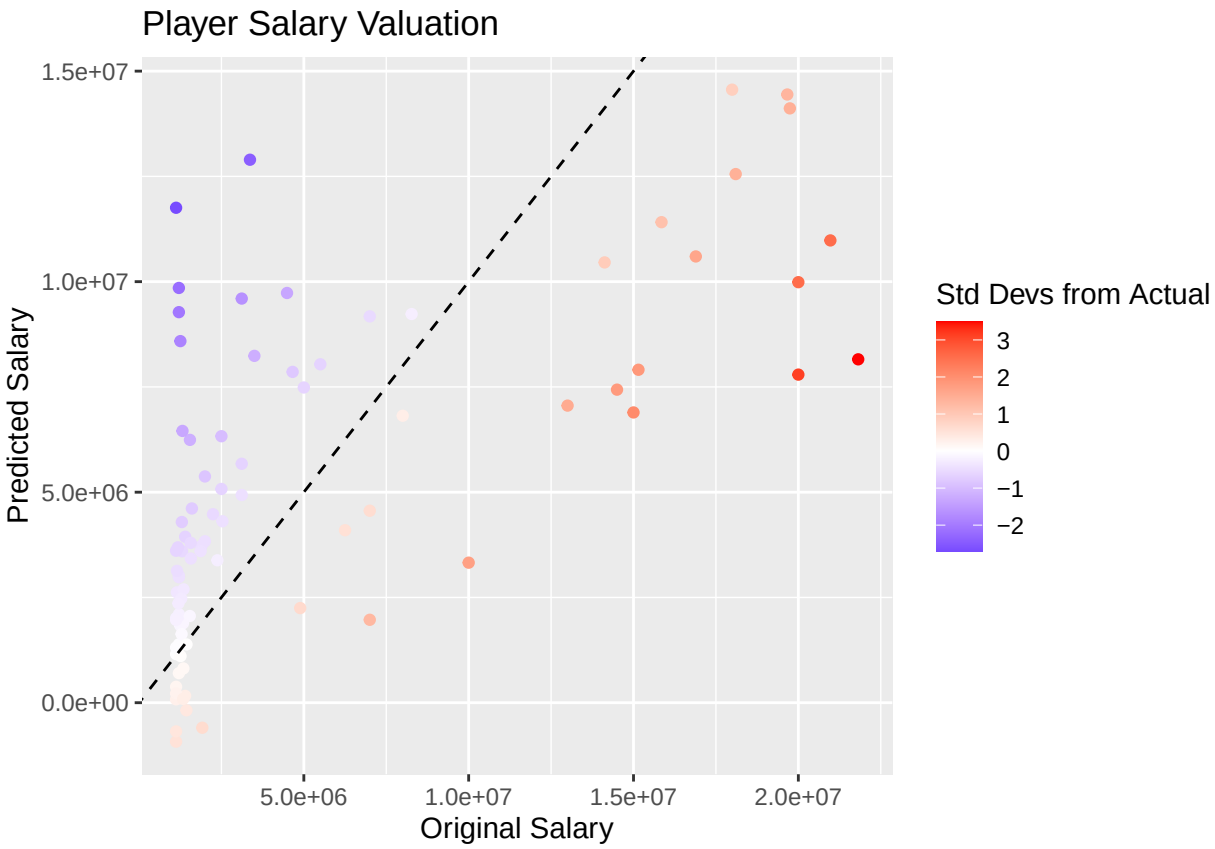
```r
# Assuming original_salaries is correctly ordered and aligned with x_matrix
original_salaries <- WR_data$Base.Salary  # Adjust column name as necessary
# Calculate differences and standard deviations away from actual
salary_diff <- original_salaries - predicted_salaries
std_devs_away <- salary_diff / predicted_sd
salary_diff
```

```
##         lambda.min
##  [1,] -10630259.67
##  [2,]  -2537411.08
##  [3,]  -2292043.44
##  [4,]   6673615.16
##  [5,]  10012590.45
##  [6,]  -9529554.37
##  [7,]   1184468.15
##  [8,]   3667016.25
##  [9,]  -1471833.16
## [10,]   1236513.83
## [11,]  -5240882.54
## [12,]  -1342647.88
## [13,]   3431595.66
## [14,]   -891969.93
## [15,]  12207509.45
## [16,]  -1193415.48
## [17,]   7064694.59
## [18,]  -4697729.72
## [19,]   8106261.83
## [20,]   5943817.43
## [21,]  -1809443.84
## [22,]   4437199.86
## [23,]  -2576012.32
## [24,]  -1833930.73
## [25,]   6291394.51
## [26,]   2049698.98
## [27,]   -961047.74
## [28,]  -1770758.12
## [29,]   9992807.14
## [30,]  -3829295.97
## [31,]   -583960.10
## [32,]  -2175867.00
## [33,]   1044427.65
## [34,]   -552962.75
## [35,]  -1816290.69
## [36,]  -5135941.25
## [37,]  -2484037.91
## [38,]    746027.64
## [39,]   -858425.40
## [40,]  -1851193.83
## [41,]  -2486386.30
## [42,]    -29012.28
```

```
## [43,]    -79546.84
## [44,]    906245.58
## [45,]  -1828698.48
## [46,]    152044.20
## [47,]   2642621.95
## [48,]   1619428.74
## [49,]  -1979465.00
## [50,]  -2558322.03
## [51,]  -2232164.48
## [52,]   5030141.96
## [53,]   5627110.47
## [54,]  -1731717.81
## [55,]   1805665.40
## [56,]  -2509824.85
## [57,]  -1778232.70
## [58,]  -2541233.11
## [59,]   5544831.32
## [60,]  -2991576.17
## [61,]   -519334.69
## [62,]  -4737452.65
## [63,]  -8640745.26
## [64,]  -7332859.58
## [65,]   -176547.08
## [66,]  -6482369.41
## [67,]  -3373577.07
## [68,]    508040.77
## [69,]   -885720.51
## [70,]  -1004670.95
## [71,]   -186401.23
## [72,]   2153684.77
## [73,]  -3009059.09
## [74,]   -339258.72
## [75,]  -8065819.88
## [76,]  13660636.11
## [77,]     57326.15
## [78,]   7242291.39
## [79,]   1249809.71
## [80,]  -2227943.96
## [81,]  -3197026.55
## [82,]    532341.95
## [83,]   5219881.74
## [84,]   -840497.70
## [85,]  -1163633.89
## [86,]   2513996.49
## [87,]   2440275.92
```

```r
# Adding this info back to the original dataframe
WR_data <- subset(inner_join_result, position == 'WR' )

WR_data$predicted_salary <- predicted_salaries
WR_data$salary_diff <- salary_diff
WR_data$std_devs_away <- std_devs_away

# Plotting
```

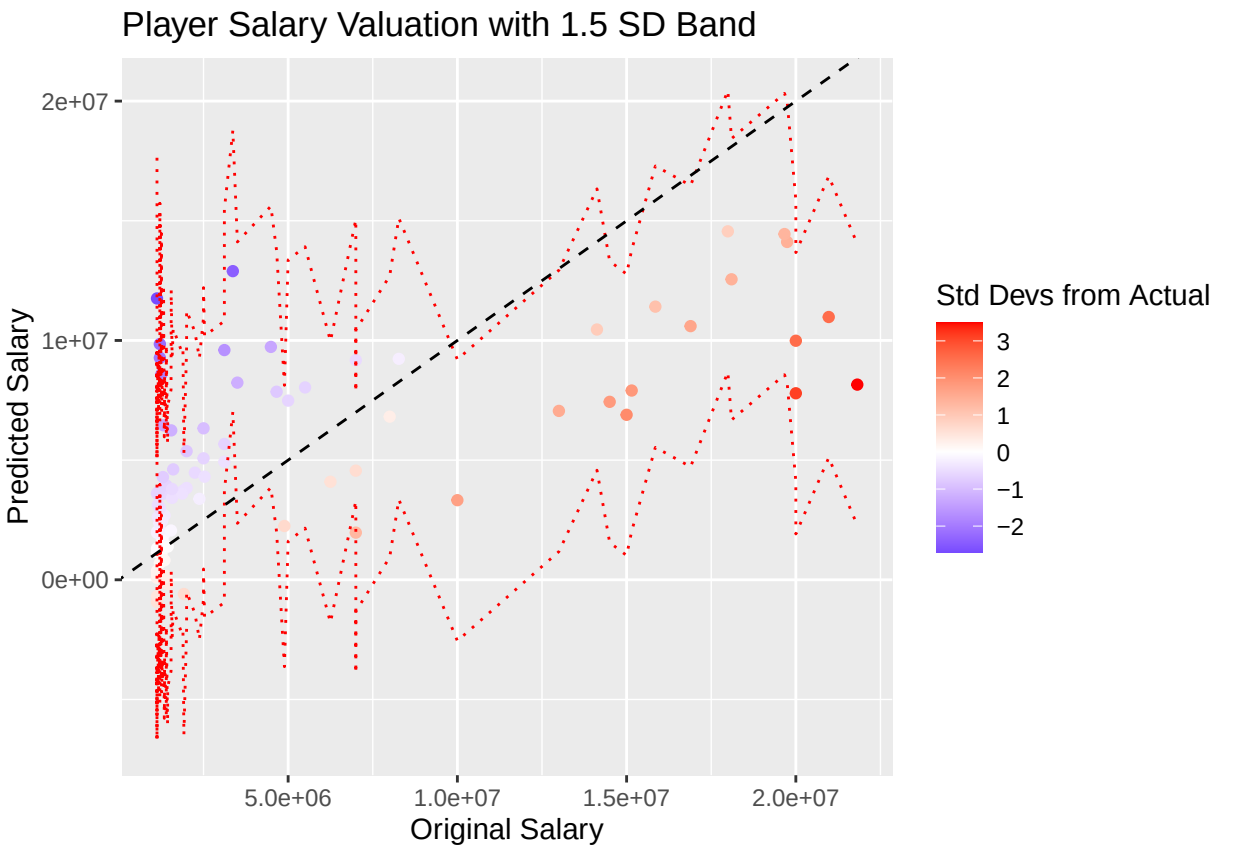
```
library(ggplot2)
ggplot(WR_data, aes(x = original_salaries, y = predicted_salary, color = std_devs_away,label = player_d
  geom_point() +
  scale_color_gradient2(low = "blue", mid = "white", high = "red", midpoint = 0) +
  labs(title = "Player Salary Valuation",
       x = "Original Salary",
       y = "Predicted Salary",
       color = "Std Devs from Actual") +
  geom_abline(intercept = 0, slope = 1, linetype = "dashed", color = "black")  # Adds a y=x reference l
```




```
# Calculate the upper and lower bounds of the 1.5 SD band
lower_bound <- predicted_salaries - 1.5 * predicted_sd
upper_bound <- predicted_salaries + 1.5 * predicted_sd

# Add these to the dataframe for plotting and analysis
WR_data$lower_bound <- lower_bound
WR_data$upper_bound <- upper_bound
```

```
ggplot(WR_data, aes(x = original_salaries, y = predicted_salary)) +
  geom_point(aes(color = std_devs_away)) +
  geom_line(aes(y = lower_bound), linetype = "dotted", color = "red") +
  geom_line(aes(y = upper_bound), linetype = "dotted", color = "red") +
  geom_abline(intercept = 0, slope = 1, linetype = "dashed", color = "black") +
  scale_color_gradient2(low = "blue", mid = "white", high = "red", midpoint = 0) +
  labs(title = "Player Salary Valuation with 1.5 SD Band",
```

```
       x = "Original Salary",
       y = "Predicted Salary",
       color = "Std Devs from Actual")
```




```
library(ggplot2)
library(ggrepel)  # Ensure this package is installed
```

```
## Warning: package 'ggrepel' was built under R version 4.3.3
```

```
ggplot(WR_data, aes(x = Base.Salary, y = predicted_salaries, label = player_display_name)) +
  geom_point(aes(color = std_devs_away), size = 4, alpha = 0.6) +
  geom_text_repel(
    aes(color = std_devs_away),
    size = 3.5,
    box.padding = 0.35,
    point.padding = 0.5,
    segment.color = 'grey50'
  ) +
  scale_color_gradient(low = "blue", high = "red") +
  labs(title = "Salary vs. Theoretical Salary",
       x = "Actual Salary",
       y = "Theoretical Estimated Salary") +
  theme_minimal() +
  theme(legend.position = "right")  # Adjust legend position if needed
```

```
## Warning: ggrepel: 54 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```