Boxing EDA

David Aponte

July 19, 2018

Background:

I remember my first boxing match. I was 17 years old, 152 lbs, 5’9. At this stage, I was in the best shape of my life. I was waking up at 5 am to run and sprint; in the afternoons I was doing countless rounds of shadow boxing, bag work, pad work, and sparring. The training was brutal, but I loved what it did to me: it taught me discipline, resiliency, and the power of hard work. And since its genesis men and women all over the world have endured the bumps and bruises of boxing for those very same reasons: it made them a better person physically and mentally. Among fighters coming in from all sorts of socio-economic, political, cultural, and religious backgrounds - boxing is to them more than bunches of punches, but a multifaceted sport that requires technical skill and strategy. When it is just you and another person in that ring, you have no team or guise, to hide behind - the real you comes out and you see the fruit of all your rigorous training. And many boxers have tasted within those heart-testing moments the thrill, and pain, of pushing yourself beyond your own self-perceived limits.

But alongside the lovers of the sport are its critics. A common critique is that boxing is nothing more than “bunches of punches” being exchanged until someone gets knocked out; that its a mindless sport built for the barbaric. But for anyone who has trained for the sport, or closely watches it, knows that that is a gross caricature. Boxing is a very technical sport - hence why some people even call it a science - that requires strategy, problem-solving, and more mental energy than anything physical. Most boxing matches are won by points which are scored by how clean and hard the punches exchanged are, who was more aggressive, and who had more control of the ring. In this analysis, I am going to explore the physical features of boxers (BMI) and the decisions of the fights using a dataset that records boxing bouts. I would like to answer some questions such as: what is the most common type of weight group? what does the BMI of boxers look like across weight groups and stances? what does the decision of boxing bouts look like across weight groups? are knockouts the most common type of decision? I will primarily focus on exploring the dataset to the lay the groundwork for further analysis in the future. With that being said, let me begin.

Loading in the dataset

```
library(readr)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------------------------------------- tidy
verse 1.2.1 --
```

```
## v ggplot2 3.0.0     v purrr   0.2.5
## v tibble  1.4.2     v dplyr   0.7.6
## v tidyr   0.8.1     v stringr 1.3.1
## v ggplot2 3.0.0     v forcats 0.3.0
```

```
## -- Conflicts ------------------------------------------------------------------ tidyverse_
conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
boxing <- read_csv("C:/Users/apont/Downloads/bouts_out_new.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_integer(),
##   stance_A = col_character(),
##   stance_B = col_character(),
##   result = col_character(),
##   decision = col_character()
## )
```

```
## See spec(...) for full column specifications.
```

```
glimpse(boxing)
```

```
## Observations: 387,427
## Variables: 26
## $ age_A    <int> 35, 26, 28, 25, 25, 24, 23, 23, 36, 27, 22, 21, 21, 2...
## $ age_B    <int> 27, 31, 26, 29, 35, 31, 31, 31, 23, 22, 28, 40, 32, 3...
## $ height_A <int> 179, 175, 176, 175, 175, 175, 175, 175, 173, 177, 175...
## $ height_B <int> 175, 185, 175, 174, 170, 175, 175, 177, 175, 175, 177...
## $ reach_A  <int> 178, 179, NA, 179, 179, 179, 179, 179, 183, 183, 179,...
## $ reach_B  <int> 179, 185, 179, 180, 170, 178, 188, 175, 179, 179, 175...
## $ stance_A <chr> "orthodox", "orthodox", "orthodox", "orthodox", "orth...
## $ stance_B <chr> "orthodox", "orthodox", "orthodox", "orthodox", "orth...
## $ weight_A <int> 160, 164, 154, 155, 155, NA, 155, 155, 152, NA, 154, ...
## $ weight_B <int> 160, 164, 154, 155, NA, NA, 155, NA, NA, NA, 153, 154...
## $ won_A    <int> 37, 48, 23, 46, 45, 44, 43, 42, 44, 26, 40, 39, 38, 3...
## $ won_B    <int> 49, 50, 47, 31, 40, 32, 19, 22, 42, 41, 30, 46, 33, 2...
## $ lost_A   <int> 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ lost_B   <int> 1, 2, 1, 3, 4, 1, 1, 3, 0, 0, 4, 7, 4, 4, 4, 4, 11, 1...
## $ drawn_A  <int> 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1,...
## $ drawn_B  <int> 1, 1, 1, 0, 0, 0, 2, 0, 1, 1, 0, 1, 1, 2, 0, 2, 2, 6,...
## $ kos_A    <int> 33, 34, 13, 32, 32, 31, 31, 30, 26, 14, 29, 29, 28, 2...
## $ kos_B    <int> 34, 32, 33, 19, 33, 28, 12, 18, 30, 30, 18, 39, 28, 1...
## $ result   <chr> "draw", "win_A", "win_B", "win_A", "win_A", "win_A", ...
## $ decision <chr> "SD", "UD", "KO", "KO", "UD", "KO", "SD", "TKO", "MD"...
## $ judge1_A <int> 110, 120, NA, 47, 118, NA, 115, 89, 116, 112, NA, 119...
## $ judge1_B <int> 118, 108, NA, 48, 110, NA, 113, 82, 112, 115, NA, 109...
## $ judge2_A <int> 115, 120, NA, 49, 119, NA, 117, 88, 114, 109, NA, 119...
## $ judge2_B <int> 113, 108, NA, 46, 109, NA, 111, 83, 114, 118, NA, 109...
## $ judge3_A <int> 114, 120, NA, 48, 117, NA, 113, 89, 117, 111, NA, 118...
## $ judge3_B <int> 114, 108, NA, 47, 111, NA, 115, 82, 111, 116, NA, 110...
```

```
#I am setting this theme to have uniformity among all of my plots
theme_set(theme_classic())
```

Exploring the raw data

```
head(boxing)
```

```
## # A tibble: 6 x 26
##   age_A age_B height_A height_B reach_A reach_B stance_A stance_B weight_A
##   <int> <int>    <int>    <int>   <int>   <int> <chr>    <chr>       <int>
## 1    35    27      179      175     178     179 orthodox orthodox      160
## 2    26    31      175      185     179     185 orthodox orthodox      164
## 3    28    26      176      175      NA     179 orthodox orthodox      154
## 4    25    29      175      174     179     180 orthodox orthodox      155
## 5    25    35      175      170     179     170 orthodox orthodox      155
## 6    24    31      175      175     179     178 orthodox orthodox       NA
## # ... with 17 more variables: weight_B <int>, won_A <int>, won_B <int>,
## #   lost_A <int>, lost_B <int>, drawn_A <int>, drawn_B <int>, kos_A <int>,
## #   kos_B <int>, result <chr>, decision <chr>, judge1_A <int>,
## #   judge1_B <int>, judge2_A <int>, judge2_B <int>, judge3_A <int>,
## #   judge3_B <int>
```

```
tail(boxing)
```

```
## # A tibble: 6 x 26
##   age_A age_B height_A height_B reach_A reach_B stance_A stance_B weight_A
##   <int> <int>    <int>    <int>   <int>   <int> <chr>    <chr>       <int>
## 1    21    NA       NA       NA      NA      NA orthodox orthodox       NA
## 2    20    NA       NA       NA      NA      NA orthodox orthodox       NA
## 3    20    NA       NA       NA      NA      NA orthodox orthodox       NA
## 4    20    26       NA       NA      NA      NA orthodox orthodox       NA
## 5    20    NA       NA       NA      NA      NA orthodox orthodox       NA
## 6    20    NA       NA       NA      NA      NA orthodox orthodox       NA
## # ... with 17 more variables: weight_B <int>, won_A <int>, won_B <int>,
## #   lost_A <int>, lost_B <int>, drawn_A <int>, drawn_B <int>, kos_A <int>,
## #   kos_B <int>, result <chr>, decision <chr>, judge1_A <int>,
## #   judge1_B <int>, judge2_A <int>, judge2_B <int>, judge3_A <int>,
## #   judge3_B <int>
```

```
glimpse(boxing)
```

```
## Observations: 387,427
## Variables: 26
## $ age_A    <int> 35, 26, 28, 25, 25, 24, 23, 23, 36, 27, 22, 21, 21, 2...
## $ age_B    <int> 27, 31, 26, 29, 35, 31, 31, 31, 23, 22, 28, 40, 32, 3...
## $ height_A <int> 179, 175, 176, 175, 175, 175, 175, 175, 173, 177, 175...
## $ height_B <int> 175, 185, 175, 174, 170, 175, 175, 177, 175, 175, 177...
## $ reach_A  <int> 178, 179, NA, 179, 179, 179, 179, 179, 183, 183, 179,...
## $ reach_B  <int> 179, 185, 179, 180, 170, 178, 188, 175, 179, 179, 175...
## $ stance_A <chr> "orthodox", "orthodox", "orthodox", "orthodox", "orth...
## $ stance_B <chr> "orthodox", "orthodox", "orthodox", "orthodox", "orth...
## $ weight_A <int> 160, 164, 154, 155, 155, NA, 155, 155, 152, NA, 154, ...
## $ weight_B <int> 160, 164, 154, 155, NA, NA, 155, NA, NA, NA, 153, 154...
## $ won_A    <int> 37, 48, 23, 46, 45, 44, 43, 42, 44, 26, 40, 39, 38, 3...
## $ won_B    <int> 49, 50, 47, 31, 40, 32, 19, 22, 42, 41, 30, 46, 33, 2...
## $ lost_A   <int> 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ lost_B   <int> 1, 2, 1, 3, 4, 1, 1, 3, 0, 0, 4, 7, 4, 4, 4, 4, 11, 1...
## $ drawn_A  <int> 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1,...
## $ drawn_B  <int> 1, 1, 1, 0, 0, 0, 2, 0, 1, 1, 0, 1, 1, 2, 0, 2, 2, 6,...
## $ kos_A    <int> 33, 34, 13, 32, 32, 31, 31, 30, 26, 14, 29, 29, 28, 2...
## $ kos_B    <int> 34, 32, 33, 19, 33, 28, 12, 18, 30, 30, 18, 39, 28, 1...
## $ result   <chr> "draw", "win_A", "win_B", "win_A", "win_A", "win_A", ...
## $ decision <chr> "SD", "UD", "KO", "KO", "UD", "KO", "SD", "TKO", "MD"...
## $ judge1_A <int> 110, 120, NA, 47, 118, NA, 115, 89, 116, 112, NA, 119...
## $ judge1_B <int> 118, 108, NA, 48, 110, NA, 113, 82, 112, 115, NA, 109...
## $ judge2_A <int> 115, 120, NA, 49, 119, NA, 117, 88, 114, 109, NA, 119...
## $ judge2_B <int> 113, 108, NA, 46, 109, NA, 111, 83, 114, 118, NA, 109...
## $ judge3_A <int> 114, 120, NA, 48, 117, NA, 113, 89, 117, 111, NA, 118...
## $ judge3_B <int> 114, 108, NA, 47, 111, NA, 115, 82, 111, 116, NA, 110...
```

Findings:

There are a ton of missing values within this dataset. Without further information as to why these values are missing, I am going to assume they are recording errors. Lets find out how much is missing.

```
missing <- sum(is.na(boxing))
missing
```

```
## [1] 3959073
```

There are 3,959,073 missing values.

```
dim(boxing)
```

```
## [1] 387427     26
```

For the sake of ease in this analysis, I am going to omit all of the rows containing missing values, as it would still leave me with a sufficient subset of boxing bouts to analyze and then visualize. For the rest of this analysis I will be working with a copy of the original boxing dataset: boxing_copy.

```
#removing NA
boxing_copy <- na.omit(boxing)
dim(boxing_copy)
```

```
## [1] 2853   26
```

About the dataset I'll be working with:

The dataset "boxing_copy"" has 2,853 boxing bouts recorded. Each bout has 27 variables - ranging from the age of the fighter to the judges decision scores. The boxers for each bout are labeled as boxer_A and boxer_B (this dataset did not provide the names of the fighters). Each column is a variable and each row is an observation. Each observation is a boxing bout (or match) with 2 boxers competing against each other. Since no names are included, I decided to label each boxer in a bout as A-boxer and B-boxer. I did have the option of gathering the A and B boxers into one column for every variable, but that would have required me adding some sort of label for each value within the variable to allow me to later distinguish the fighters. That didnt seem like the best approach to me since each observation is a boxing match, which requires two seperate boxers; each with unique scores (e.g. judge1_A, judge1_B) and prior history (e.g. wins_A, wins_B). So I decided to keep the dataset as is, variables for each boxer - e.g. height_A, height_B, reach_A, reach_B - and to analyze the features for each boxer seperately. This option will allow me to analyze the differences between the boxers and how that affected the results of the boxing match (this is something for a follow up project).

```
glimpse(boxing_copy)
```

```
## Observations: 2,853
## Variables: 26
## $ age_A    <int> 35, 26, 25, 23, 21, 21, 19, 29, 28, 27, 23, 33, 31, 3...
## $ age_B    <int> 27, 31, 29, 31, 40, 32, 32, 33, 33, 33, 30, 28, 31, 2...
## $ height_A <int> 179, 175, 175, 175, 175, 175, 175, 173, 173, 173, 173...
## $ height_B <int> 175, 185, 174, 175, 174, 180, 167, 179, 169, 173, 165...
## $ reach_A  <int> 178, 179, 179, 179, 179, 179, 179, 178, 178, 178, 178...
## $ reach_B  <int> 179, 185, 180, 188, 180, 188, 173, 180, 183, 173, 168...
## $ stance_A <chr> "orthodox", "orthodox", "orthodox", "orthodox", "orth...
## $ stance_B <chr> "orthodox", "orthodox", "orthodox", "orthodox", "orth...
## $ weight_A <int> 160, 164, 155, 155, 154, 154, 150, 139, 140, 135, 139...
## $ weight_B <int> 160, 164, 155, 155, 154, 154, 149, 143, 140, 134, 139...
## $ won_A    <int> 37, 48, 46, 43, 39, 38, 31, 29, 26, 24, 13, 32, 26, 2...
## $ won_B    <int> 49, 50, 31, 19, 46, 33, 31, 29, 29, 29, 20, 19, 29, 2...
## $ lost_A   <int> 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0,...
## $ lost_B   <int> 1, 2, 3, 1, 7, 4, 1, 6, 1, 6, 11, 1, 4, 5, 1, 7, 3, 1...
## $ drawn_A  <int> 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,...
## $ drawn_B  <int> 1, 1, 0, 2, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1,...
## $ kos_A    <int> 33, 34, 32, 31, 29, 28, 23, 20, 18, 17, 10, 29, 23, 2...
## $ kos_B    <int> 34, 32, 19, 12, 39, 28, 23, 23, 20, 17, 11, 6, 20, 13...
## $ result   <chr> "draw", "win_A", "win_A", "win_A", "win_A", "win_A", ...
## $ decision <chr> "SD", "UD", "KO", "SD", "UD", "TKO", "TKO", "TKO", "T...
## $ judge1_A <int> 110, 120, 47, 115, 119, 40, 78, 70, 89, 119, 10, 50, ...
## $ judge1_B <int> 118, 108, 48, 113, 109, 35, 73, 60, 80, 109, 9, 43, 1...
## $ judge2_A <int> 115, 120, 49, 117, 119, 40, 78, 70, 90, 120, 10, 50, ...
## $ judge2_B <int> 113, 108, 46, 111, 109, 35, 73, 63, 79, 108, 9, 43, 1...
## $ judge3_A <int> 114, 120, 48, 113, 118, 40, 78, 70, 90, 119, 10, 49, ...
## $ judge3_B <int> 114, 108, 47, 115, 110, 35, 73, 63, 79, 109, 9, 44, 1...
```

Tidying and transforming the data:

I want to now make some new variables and coerce some of the categorical variables into factors.

```
#coercing them into factors
boxing_copy$result <- as.factor(boxing_copy$result)
boxing_copy$decision <- as.factor(boxing_copy$decision)
boxing_copy$stance_A <- as.factor(boxing_copy$stance_A)
boxing_copy$stance_B <- as.factor(boxing_copy$stance_B)

#making weight groups, difference variables, and BMI
boxing_copy<- boxing_copy %>%
  mutate(height_diff = height_A - height_B, reach_diff = reach_A - reach_B, won_diff = won_A - w
on_B, lost_diff = lost_A - lost_B, kos_diff = kos_A - kos_B, height_A_m = height_A / 100, weight
_A_kg = weight_A * 0.45, bmi_A = weight_A_kg / height_A_m**2, height_B_m = height_B / 100, weigh
t_B_kg = weight_B * 0.45, bmi_B = weight_B_kg / height_B_m**2, weight_group_A = case_when(
    weight_A > 200 ~ "heavyweight",
    weight_A == 200 ~ "crusierweight",
    weight_A < 200 & weight_A >= 175 ~
       "light heavyweight",
    weight_A < 175 & weight_A >= 168 ~
      "super middleweight",
    weight_A < 168 & weight_A >= 160 ~
      "middleweight",
    weight_A < 160 & weight_A >= 154 ~
      "super welterweight",
    weight_A < 154 & weight_A >= 147 ~
      "welterweight",
    weight_A < 147 & weight_A >= 140 ~
      "super lightweight",
    weight_A < 140 & weight_A >= 135 ~
      "lightweight",
    weight_A < 135 & weight_A >= 130 ~
      "super featherweight",
    weight_A < 126 & weight_A >= 126 ~
      "featherweight",
    weight_A < 122 & weight_A >= 122 ~
      "super bantamweight",
    weight_A < 118 & weight_A >= 118 ~
      "bantamweight",
    weight_A < 115 & weight_A >= 115 ~
      "super flyweight",
    weight_A < 112 & weight_A >= 112 ~
      "flyweight",
    weight_A < 108 & weight_A >= 108 ~
      "light flyweight",
    weight_A < 108 & weight_A >= 105 ~
      "minimumweight"), weight_group_B = case_when(
    weight_B > 200 ~ "heavyweight",
    weight_B == 200 ~ "crusierweight",
    weight_B < 200 & weight_B >= 175 ~
       "light heavyweight",
    weight_B < 175 & weight_B >= 168 ~
      "super middleweight",
    weight_B < 168 & weight_B >= 160 ~
      "middleweight",
    weight_B < 160 & weight_B >= 154 ~
```

```
      "super welterweight",
    weight_B < 154 & weight_B >= 147 ~
      "welterweight",
    weight_B < 147 & weight_B >= 140 ~
      "super lightweight",
    weight_B < 140 & weight_B >= 135 ~
      "lightweight",
    weight_B < 135 & weight_B >= 130 ~
      "super featherweight",
    weight_B < 126 & weight_B >= 126 ~
      "featherweight",
    weight_B < 122 & weight_B >= 122 ~
      "super bantamweight",
    weight_B < 118 & weight_B >= 118 ~
      "bantamweight",
    weight_B < 115 & weight_B >= 115 ~
      "super flyweight",
    weight_B < 112 & weight_B >= 112 ~
      "flyweight",
    weight_B < 108 & weight_B >= 108 ~
      "light flyweight",
    weight_B < 108 & weight_B >= 105 ~
      "minimumweight"))

#turning weight_groups into an ordinal variable
weight_groups = c(
  "minimumweight",
  "light flyweight",
  "flyweight",
  "super flyweight",
  "bantamweight",
  "super bantamweight",
  "featherweight",
  "super featherweight",
  "lightweight",
  "super lightweight",
  "welterweight",
  "super welterweight",
  "middleweight",
  "super middleweight",
  "light heavyweight",
  "cruiserweight",
  "heavyweight"
  )

boxing_copy$weight_group_B <- factor(boxing_copy$weight_group_B, ordered = TRUE, levels = weight
_groups)
boxing_copy$weight_group_A <- factor(boxing_copy$weight_group_A, ordered = TRUE, levels = weight
_groups)
glimpse(boxing_copy)
```

```
## Observations: 2,853
## Variables: 39
## $ age_A          <int> 35, 26, 25, 23, 21, 21, 19, 29, 28, 27, 23, 33,...
## $ age_B          <int> 27, 31, 29, 31, 40, 32, 32, 33, 33, 33, 30, 28,...
## $ height_A       <int> 179, 175, 175, 175, 175, 175, 175, 173, 173, 17...
## $ height_B       <int> 175, 185, 174, 175, 174, 180, 167, 179, 169, 17...
## $ reach_A        <int> 178, 179, 179, 179, 179, 179, 179, 178, 178, 17...
## $ reach_B        <int> 179, 185, 180, 188, 180, 188, 173, 180, 183, 17...
## $ stance_A       <fct> orthodox, orthodox, orthodox, orthodox, orthodo...
## $ stance_B       <fct> orthodox, orthodox, orthodox, orthodox, orthodo...
## $ weight_A       <int> 160, 164, 155, 155, 154, 154, 150, 139, 140, 13...
## $ weight_B       <int> 160, 164, 155, 155, 154, 154, 149, 143, 140, 13...
## $ won_A          <int> 37, 48, 46, 43, 39, 38, 31, 29, 26, 24, 13, 32,...
## $ won_B          <int> 49, 50, 31, 19, 46, 33, 31, 29, 29, 29, 20, 19,...
## $ lost_A         <int> 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ lost_B         <int> 1, 2, 3, 1, 7, 4, 1, 6, 1, 6, 11, 1, 4, 5, 1, 7...
## $ drawn_A        <int> 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ drawn_B        <int> 1, 1, 0, 2, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1,...
## $ kos_A          <int> 33, 34, 32, 31, 29, 28, 23, 20, 18, 17, 10, 29,...
## $ kos_B          <int> 34, 32, 19, 12, 39, 28, 23, 23, 20, 17, 11, 6, ...
## $ result         <fct> draw, win_A, win_A, win_A, win_A, win_A, win_A,...
## $ decision       <fct> SD, UD, KO, SD, UD, TKO, TKO, TKO, TKO, UD, TKO...
## $ judge1_A       <int> 110, 120, 47, 115, 119, 40, 78, 70, 89, 119, 10...
## $ judge1_B       <int> 118, 108, 48, 113, 109, 35, 73, 60, 80, 109, 9,...
## $ judge2_A       <int> 115, 120, 49, 117, 119, 40, 78, 70, 90, 120, 10...
## $ judge2_B       <int> 113, 108, 46, 111, 109, 35, 73, 63, 79, 108, 9,...
## $ judge3_A       <int> 114, 120, 48, 113, 118, 40, 78, 70, 90, 119, 10...
## $ judge3_B       <int> 114, 108, 47, 115, 110, 35, 73, 63, 79, 109, 9,...
## $ height_diff    <int> 4, -10, 1, 0, 1, -5, 8, -6, 4, 0, 8, 1, 1, -3, ...
## $ reach_diff     <int> -1, -6, -1, -9, -1, -9, 6, -2, -5, 5, 10, -10, ...
## $ won_diff       <int> -12, -2, 15, 24, -7, 5, 0, 0, -3, -5, -7, 13, -...
## $ lost_diff      <int> -1, -1, -2, 0, -7, -4, -1, -6, -1, -6, -11, -1,...
## $ kos_diff       <int> -1, 2, 13, 19, -10, 0, 0, -3, -2, 0, -1, 23, 3,...
## $ height_A_m     <dbl> 1.79, 1.75, 1.75, 1.75, 1.75, 1.75, 1.75, 1.73,...
## $ weight_A_kg    <dbl> 72.00, 73.80, 69.75, 69.75, 69.30, 69.30, 67.50...
## $ bmi_A          <dbl> 22.47121, 24.09796, 22.77551, 22.77551, 22.6285...
## $ height_B_m     <dbl> 1.75, 1.85, 1.74, 1.75, 1.74, 1.80, 1.67, 1.79,...
## $ weight_B_kg    <dbl> 72.00, 73.80, 69.75, 69.75, 69.30, 69.30, 67.05...
## $ bmi_B          <dbl> 23.51020, 21.56318, 23.03805, 22.77551, 22.8894...
## $ weight_group_A <ord> middleweight, middleweight, super welterweight,...
## $ weight_group_B <ord> middleweight, middleweight, super welterweight,...
```

Method:

For this analysis, I decided to make several new variables. One of them is weight_group_A and weight_group_B. These variables will help me categorize each boxer into a weight class (or weight group). With each boxer in a category, I can make comparisons across groups. I also wanted to take the height and weight information and use it to create a BMI variable (weight/height**2). Finally, to compare each boxer in a bout, I decided to make variables for the differences between each boxers height, reach, wins, losses, and knockouts. I didn't want to make a weight difference variable since boxers normally fight each other at similar weights - one exception to this is the heavyweight class which can have two boxers competing at very different weights, e.g. 220 lbs vs. 300 lbs. Since the heavyweight class is anything above 200, its possible to have boxers compete with large weight differences. This doesnt happen at the lower weights, since boxers compete at the required weight, for that particular weight division. The difference variables will allow me to compare boxers and see if the differences affect the fight results.

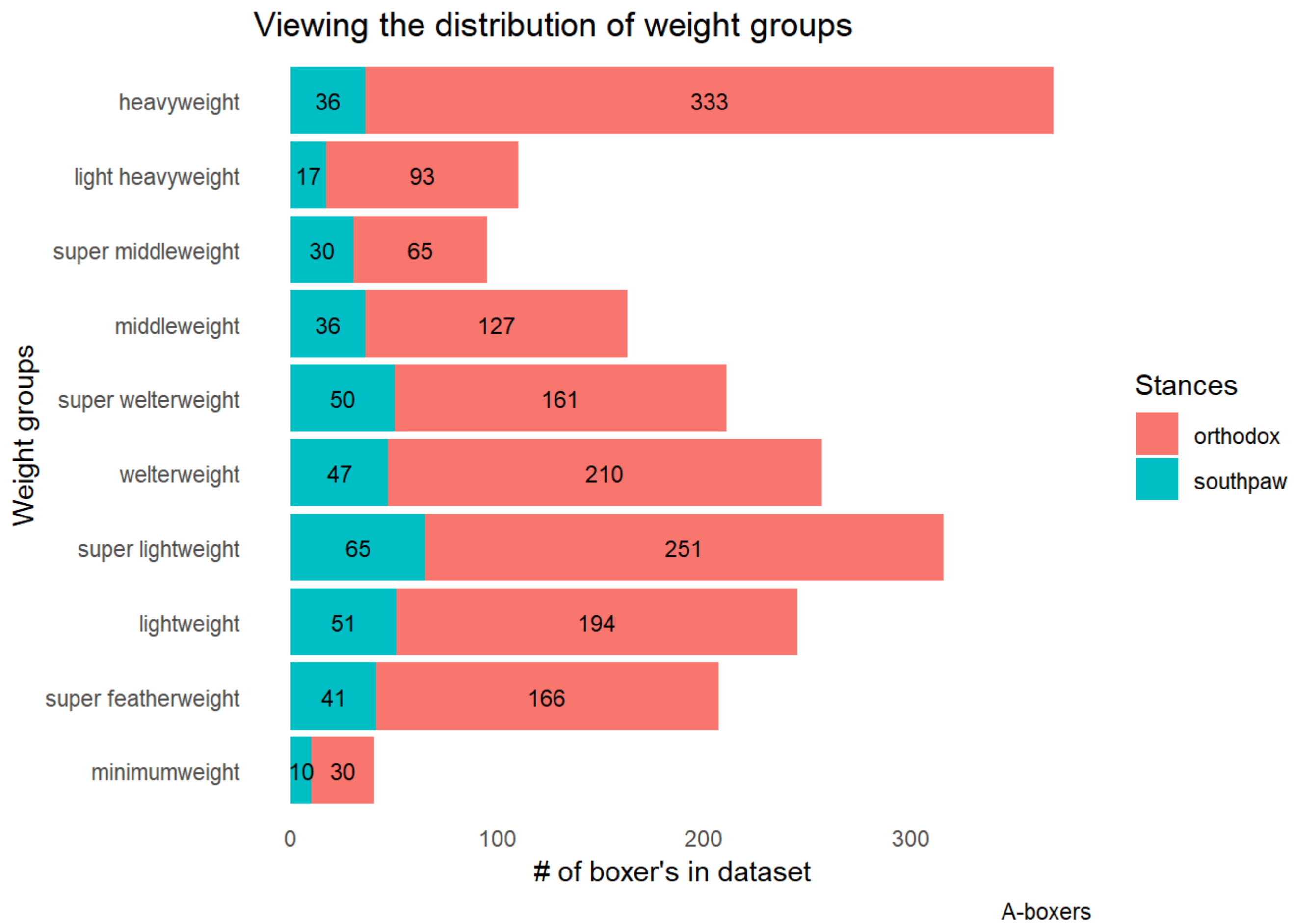
Note: a negative value for the difference variables mean that B-boxers had a greater value than A-boxers. E.g. if boxer A had a reach of 100, and boxer B had a reach of 150, the reach difference would be 100 - 150 = -50. This negative value means that the B-boxer had a greater reach.

As I begin, please keep in mind that the nature of these plots are primarily exploratory - i.e. Im looking to describe the data. This analysis will mainly lay the ground work for further projects.

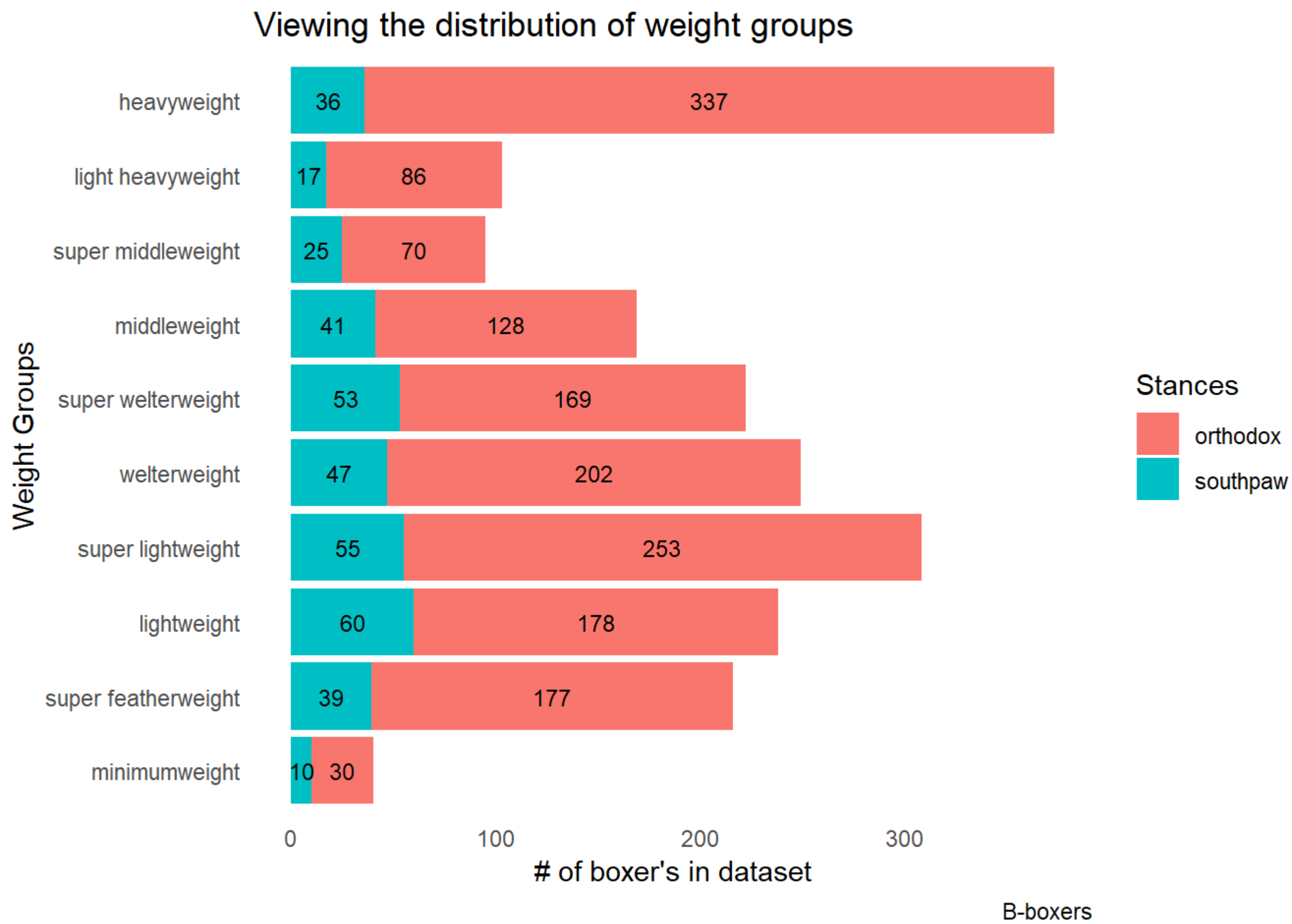
I would first like to view the distribution of boxers by weight group within the dataset - for both A-boxers and B-boxers.

Visualizing the distribution of boxers according to weight group and stance

```
#Plot 1
#distribution of weight_group_A
na.omit(boxing_copy) %>%
  ggplot(aes(x = weight_group_A, fill = stance_A)) +
  geom_bar() +
  labs(x = "Weight groups", y = "# of boxer's in dataset", title = "Viewing the distribution of
 weight groups",
       caption = "A-boxers") +
  coord_flip() +
  guides(fill=guide_legend(title="Stances")) +
  theme(axis.line = element_blank(), axis.ticks = element_blank()) +
   geom_text(aes(label=..count..),stat="count",position=position_stack(0.5), size = 3)
```



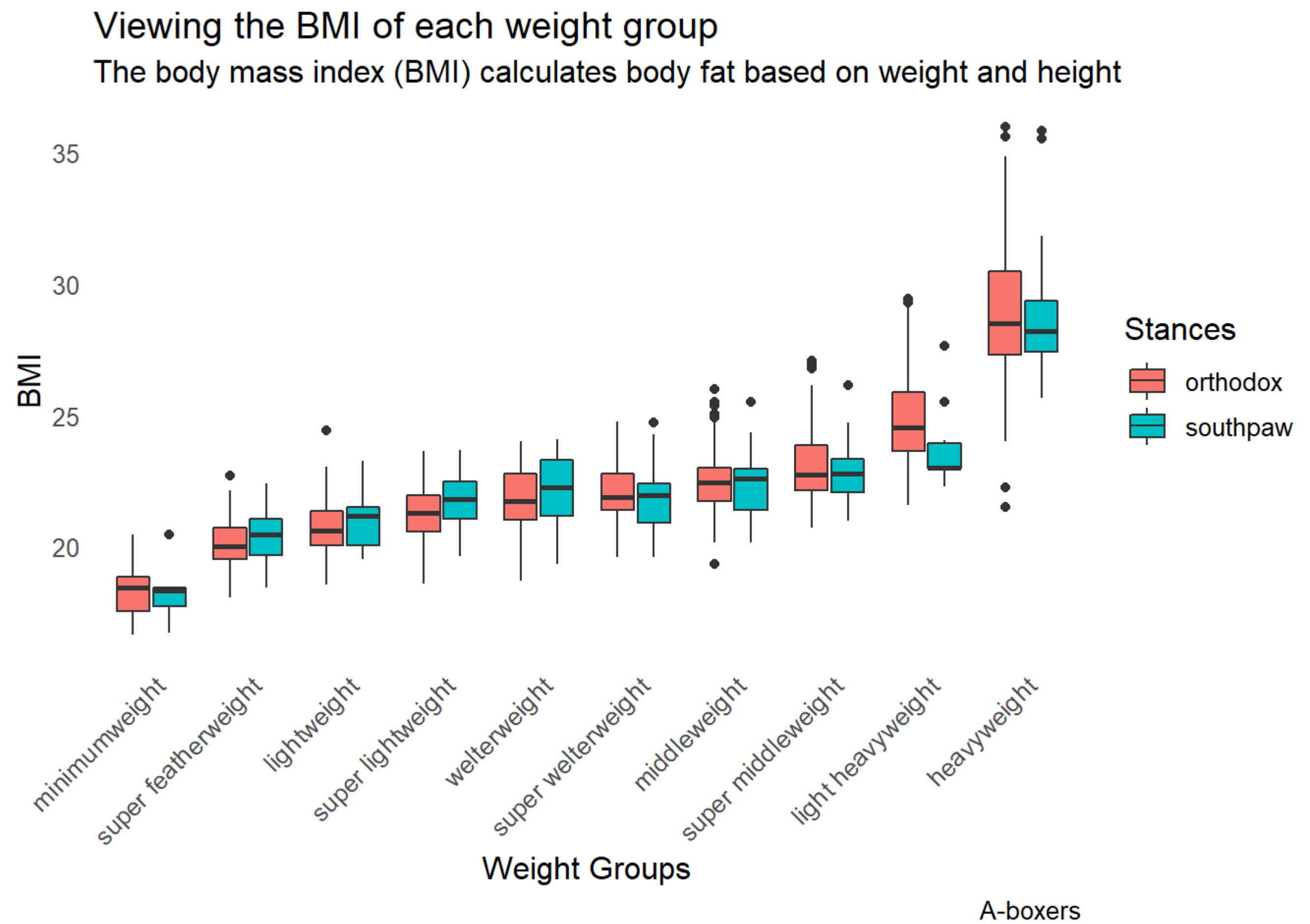

```
#distribution of weight_group_B
na.omit(boxing_copy) %>%
  ggplot(aes(x = weight_group_B, fill = stance_B)) +
  geom_bar() +
  labs(x = "Weight Groups", y = "# of boxer's in dataset", title = "Viewing the distribution of
 weight groups",
       caption = "B-boxers") +
  coord_flip() +
  guides(fill=guide_legend(title="Stances")) +
  theme(axis.line = element_blank(), axis.ticks = element_blank()) +
   geom_text(aes(label=..count..),stat="count",position=position_stack(0.5), size = 3)
```



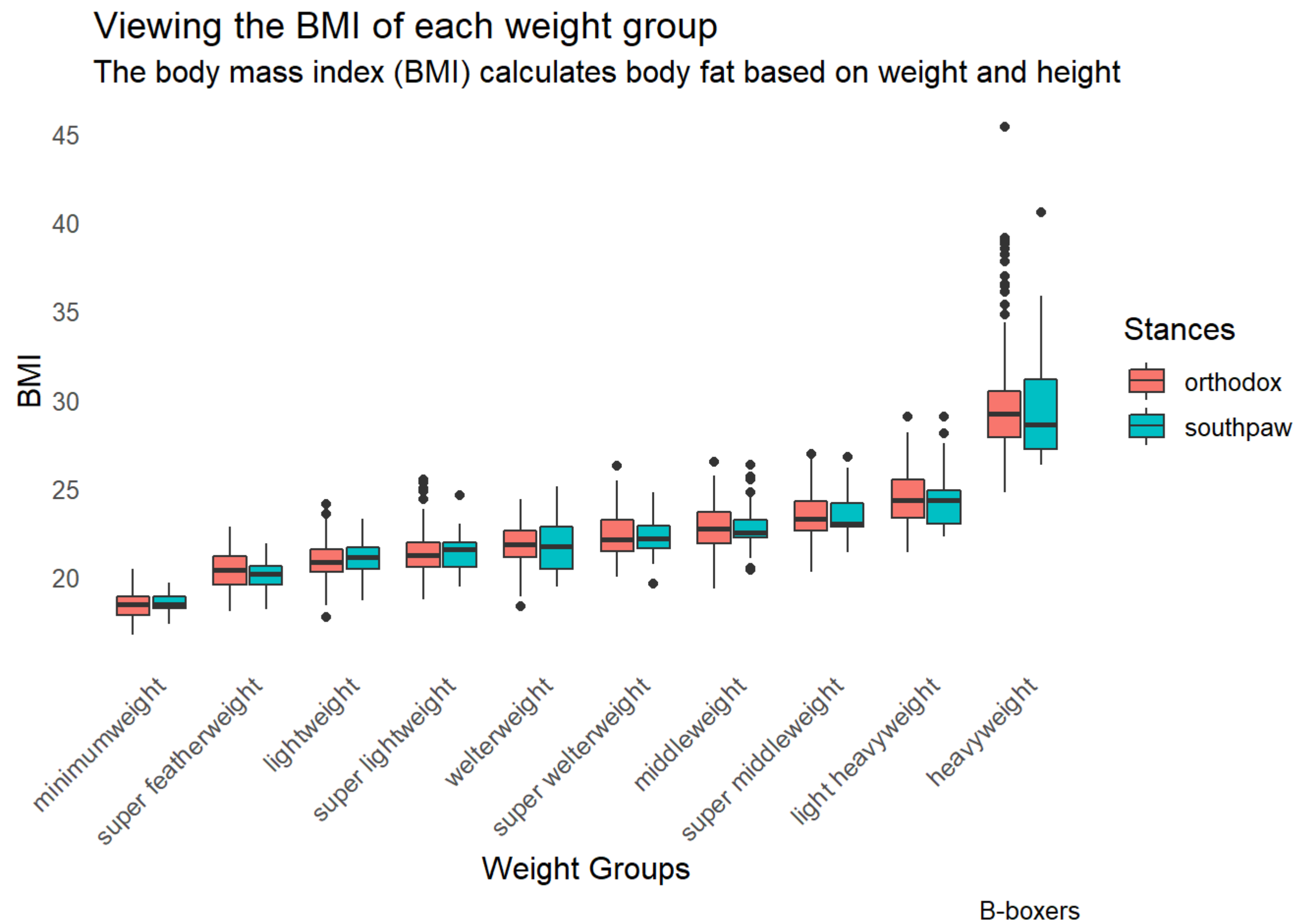

```
#removing the axis lines and ticks
```

Viewing the distribution of the BMI according to weight group and stance

```
#PLOT 2
#A-boxer
ggplot(na.omit(boxing_copy), aes(x = weight_group_A, y = bmi_A, fill = stance_A)) +
  geom_boxplot() +
  theme(text = element_text(size= 12), axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(x = "Weight Groups", y = "BMI", title = "Viewing the BMI of each weight group",
       subtitle = "The body mass index (BMI) calculates body fat based on weight and height",
       caption = "A-boxers") +
  guides(fill=guide_legend(title="Stances")) +
  theme(axis.line = element_blank(), axis.ticks = element_blank())
```



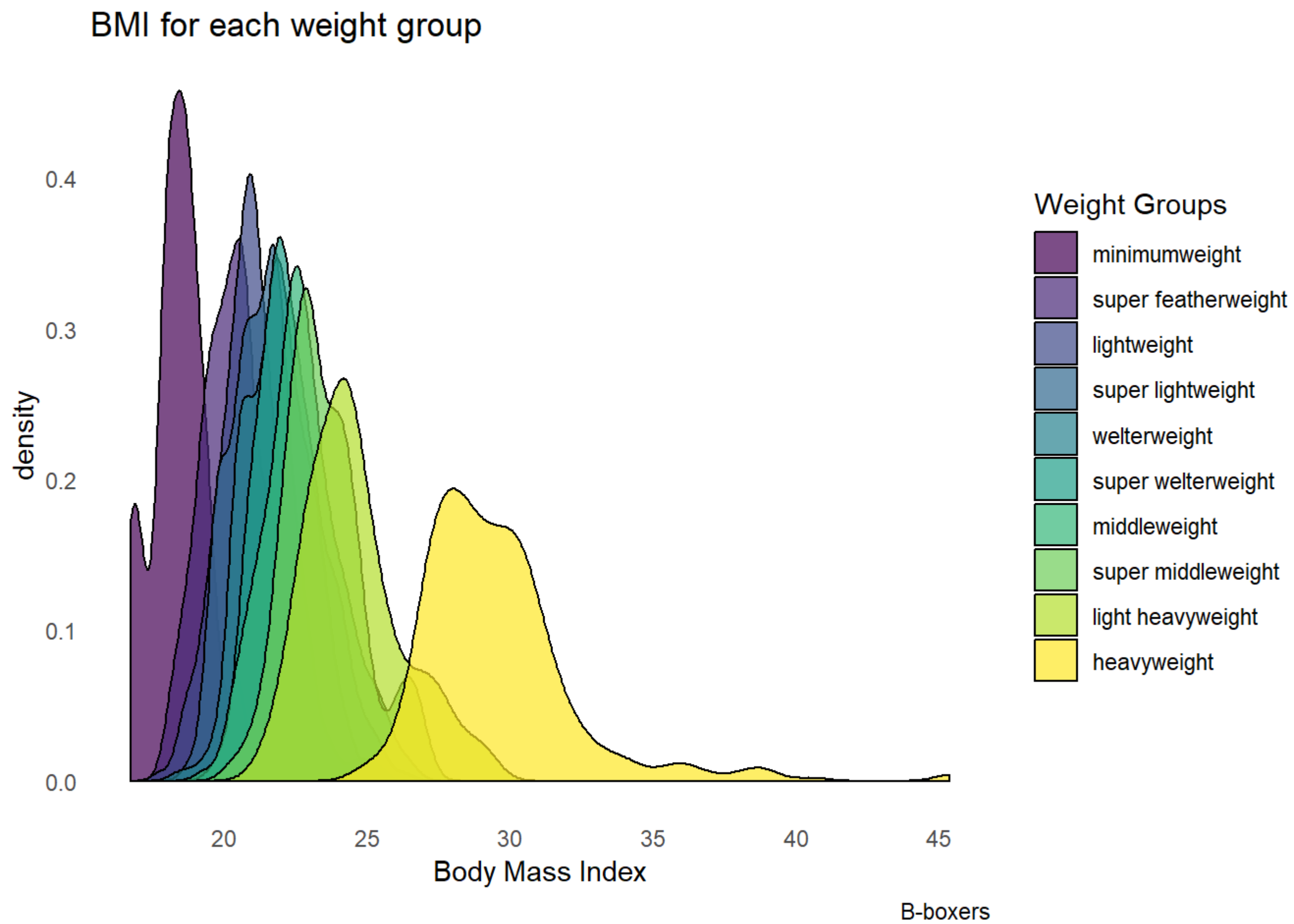

```
#B-boxer
ggplot(na.omit(boxing_copy), aes(x = weight_group_B, y = bmi_B, fill = stance_B)) +
  geom_boxplot() +
  theme(text = element_text(size= 12), axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(x = "Weight Groups", y = "BMI", title = "Viewing the BMI of each weight group",
       subtitle = "The body mass index (BMI) calculates body fat based on weight and height",
       caption = "B-boxers") +
  guides(fill=guide_legend(title="Stances")) +
  theme(axis.line = element_blank(), axis.ticks = element_blank())
```



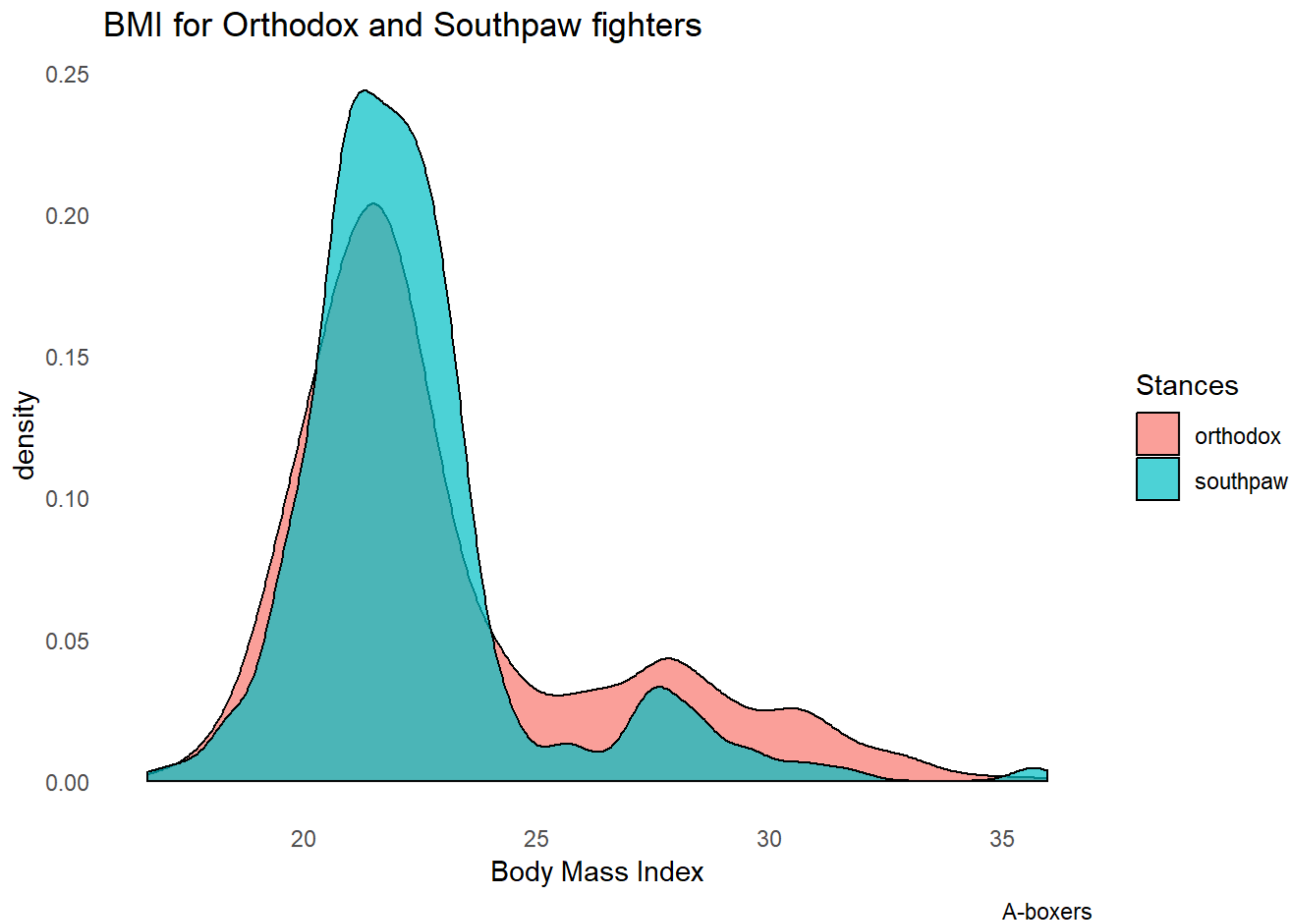
Viewing the shape of the distribution

```
#PLOT 3
#A-boxer
ggplot(na.omit(boxing_copy), aes(x = bmi_A)) +
  geom_density(aes(fill = weight_group_A), alpha = 0.7) +
  labs(x = "Body Mass Index", title = "BMI for each weight group",
       caption = "A-boxers") +
  guides(fill=guide_legend(title="Weight Groups")) +
  theme(axis.line = element_blank(), axis.ticks = element_blank())
```

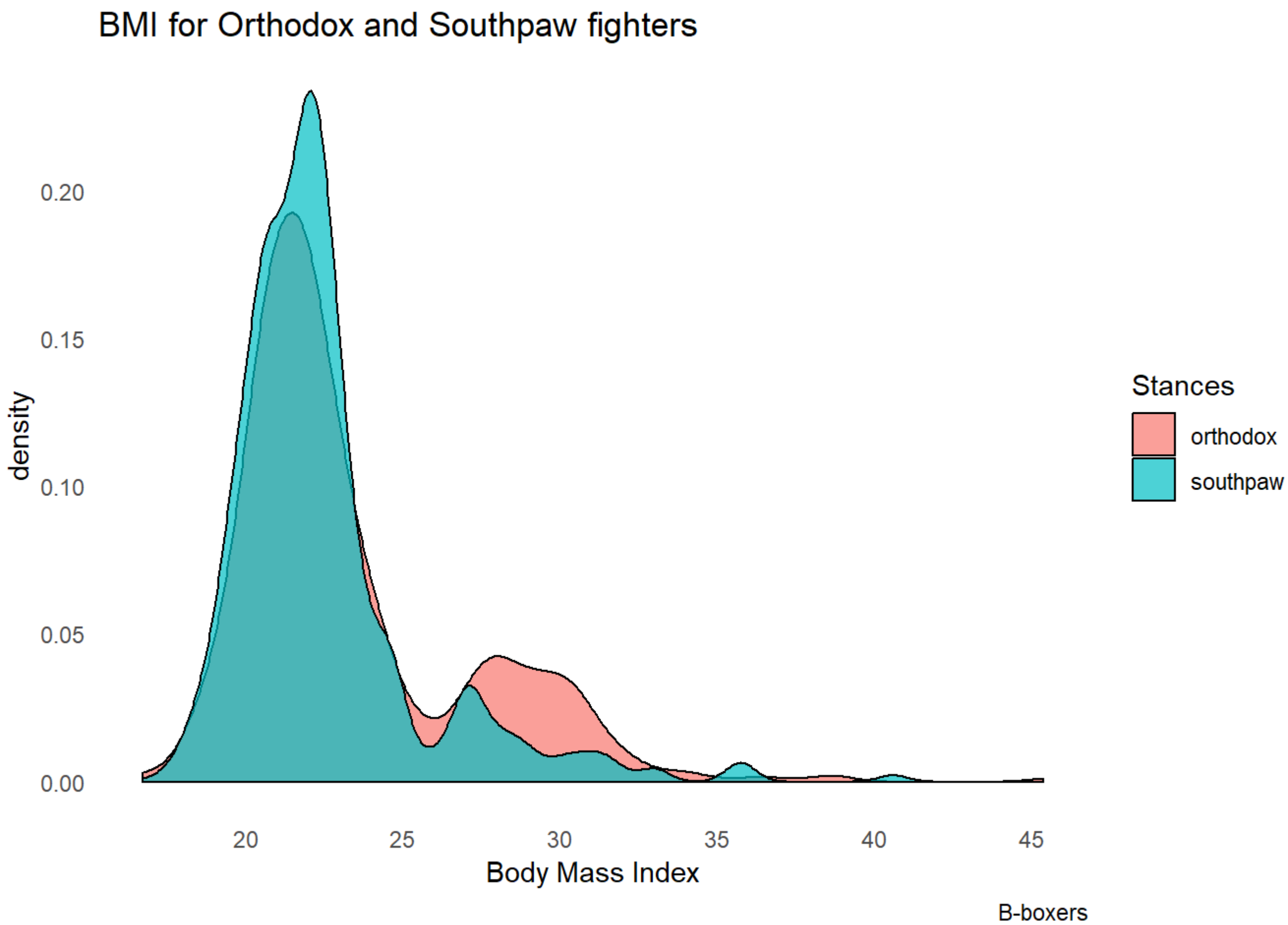

```
#B-boxer
ggplot(na.omit(boxing_copy), aes(x = bmi_B)) +
  geom_density(aes(fill = weight_group_B), alpha = 0.7) +
  labs(x = "Body Mass Index", title = "BMI for each weight group",
       caption = "B-boxers") +
  guides(fill=guide_legend(title="Weight Groups")) +
  theme(axis.line = element_blank(), axis.ticks = element_blank())
```




```
#A-boxer
ggplot(na.omit(boxing_copy), aes(x = bmi_A)) +
  geom_density(aes(fill = stance_A), alpha = 0.7) +
  labs(x = "Body Mass Index", title = "BMI for Orthodox and Southpaw fighters",
       caption = "A-boxers") +
  guides(fill=guide_legend(title="Stances")) +
  theme(axis.line = element_blank(), axis.ticks = element_blank())
```

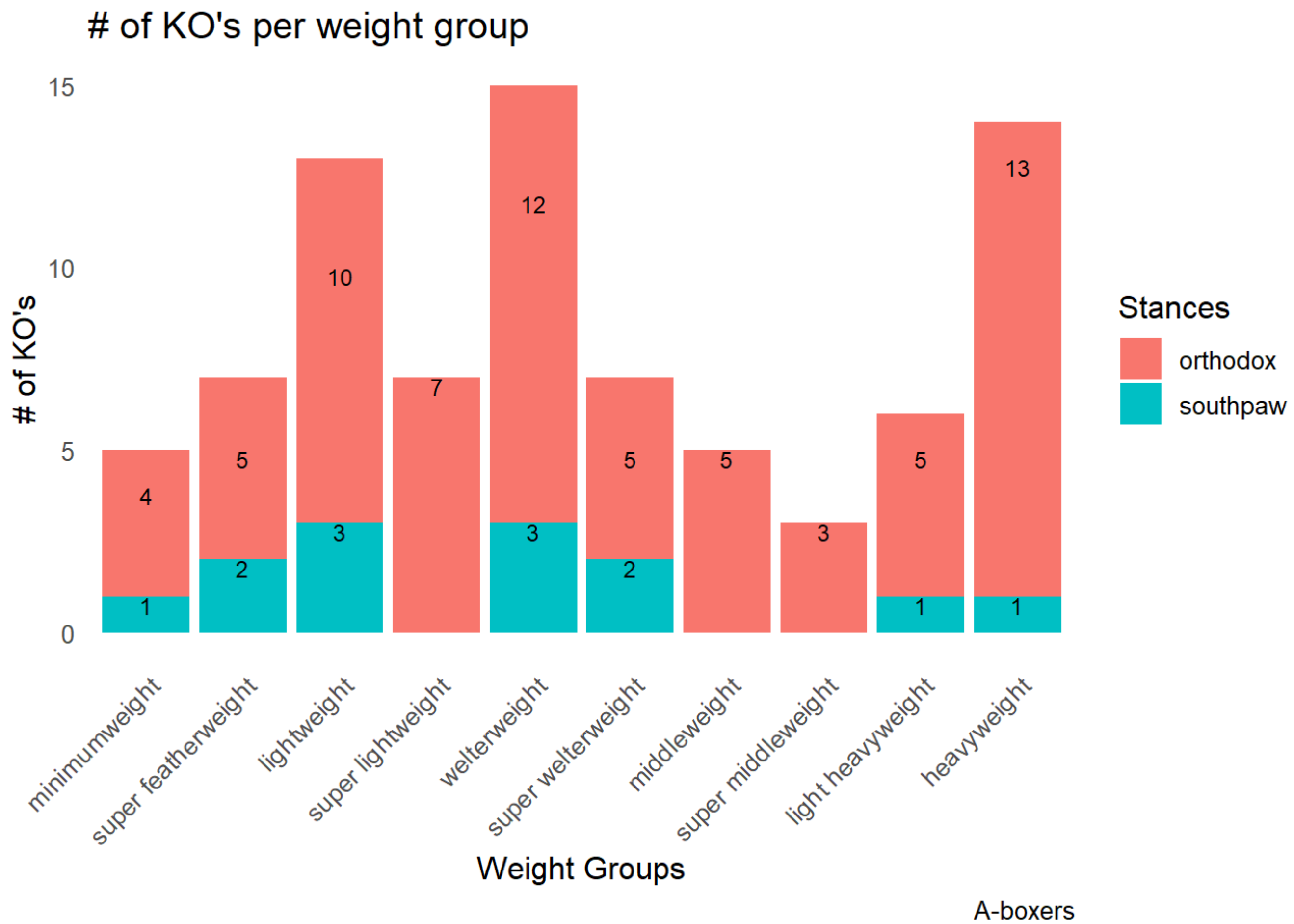



```
#B-boxer
ggplot(na.omit(boxing_copy), aes(x = bmi_B)) +
  geom_density(aes(fill = stance_B), alpha = 0.7) +
  labs(x = "Body Mass Index", title = "BMI for Orthodox and Southpaw fighters",
       caption = "B-boxers") +
  guides(fill=guide_legend(title="Stances")) +
  theme(axis.line = element_blank(), axis.ticks = element_blank())
```

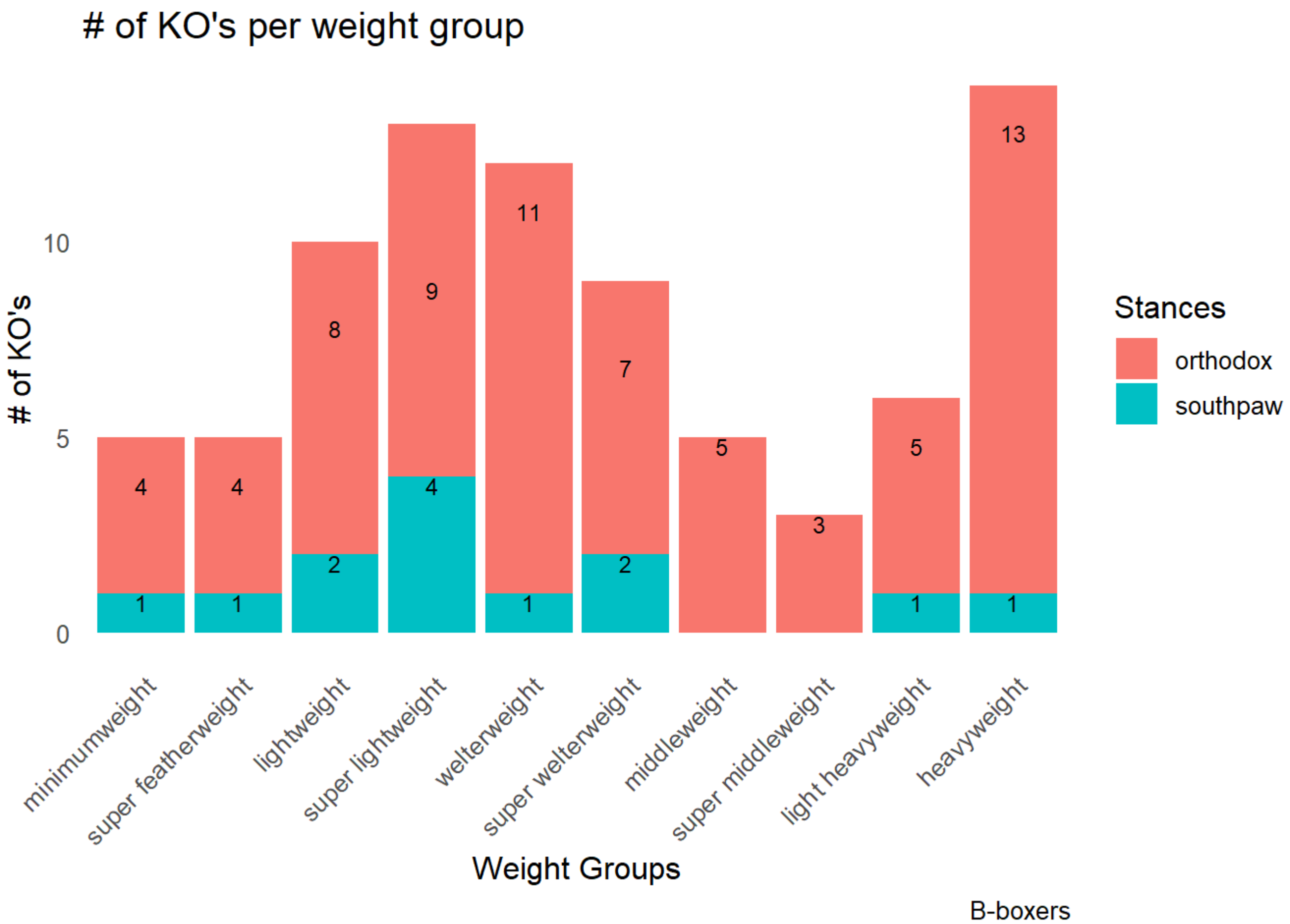


Viewing the different bout decisions (TKO, KO, UD, etc.) according to weight group and stance

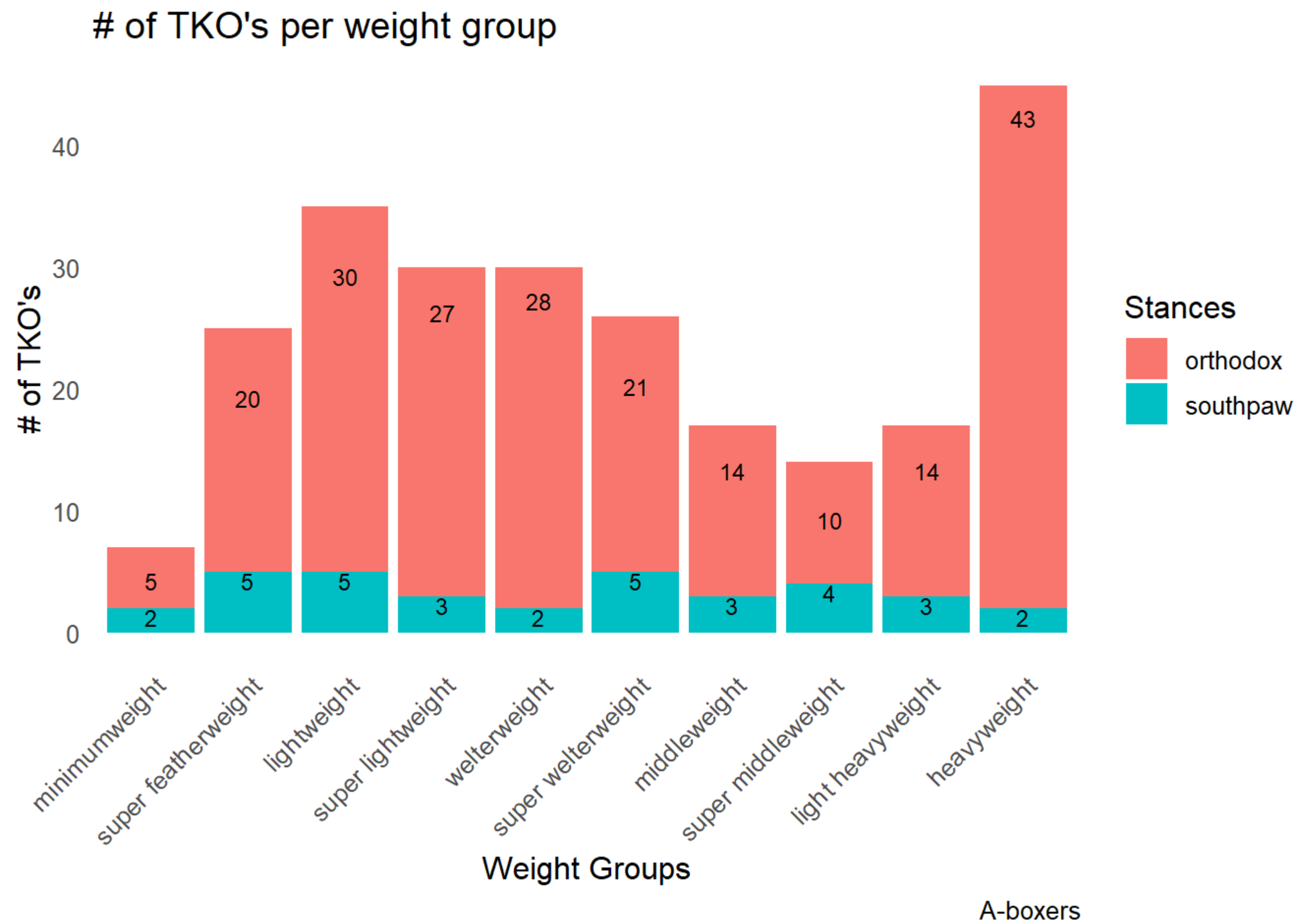
```
#PLOT 4
#KO's A
na.omit(boxing_copy) %>%
  group_by(decision, weight_group_A, stance_A) %>%
  filter(decision == "KO") %>%
  tally() %>%
  ggplot(aes(x=weight_group_A, y = n, fill = stance_A)) +
  geom_col() +
  theme(text = element_text(size = 12), axis.text.x = element_text(angle = 45, hjust = 1),
        axis.line = element_blank(), axis.ticks = element_blank()) +
  geom_text(aes(label= n), vjust=1, size = 3) +
  labs(x = "Weight Groups", y = "# of KO's",
       title = "# of KO's per weight group",
       caption = "A-boxers") +
  guides(fill=guide_legend(title="Stances"))
```



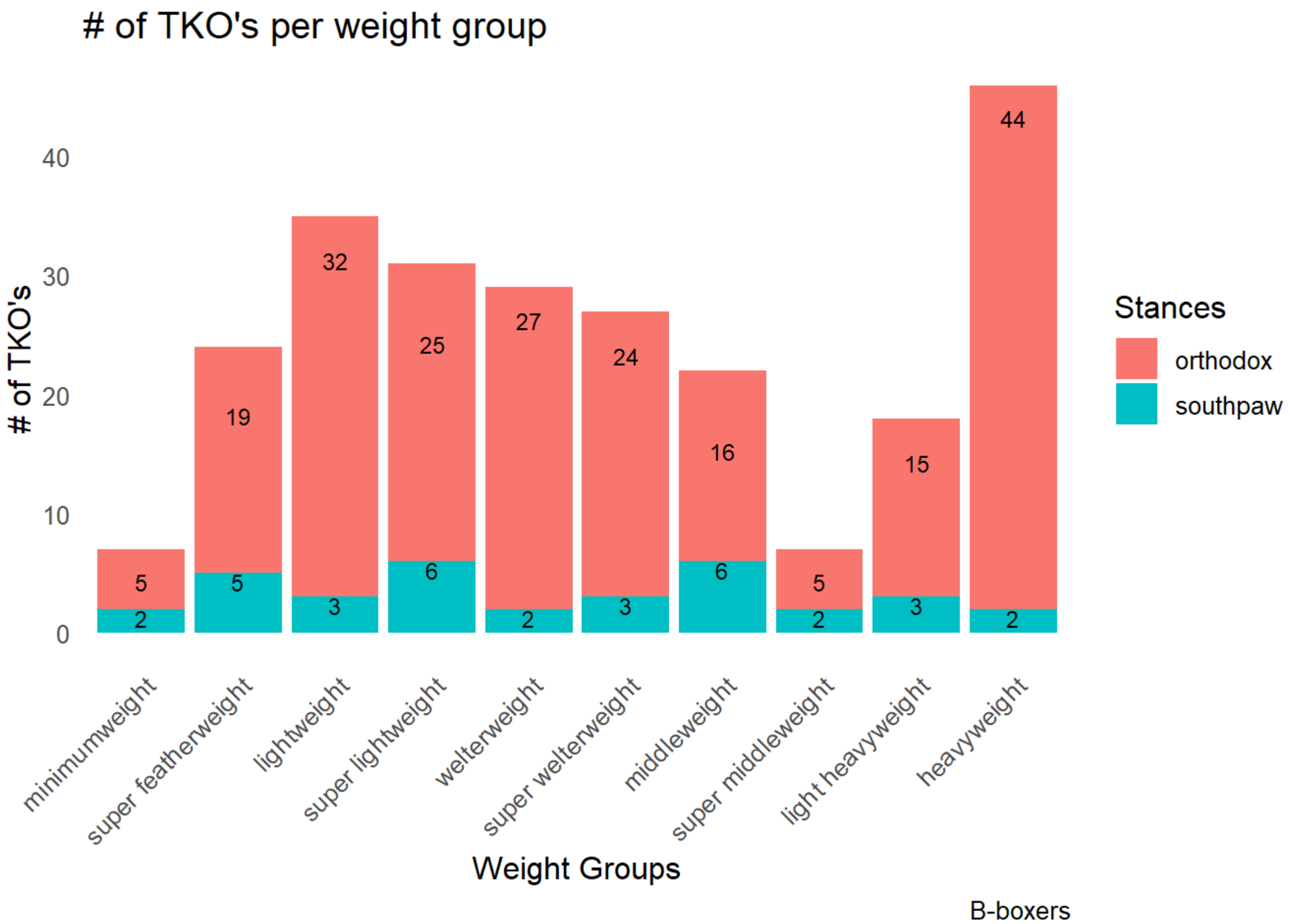

```
#KO's B
na.omit(boxing_copy) %>%
  group_by(decision, weight_group_B, stance_B) %>%
  filter(decision == "KO") %>%
  tally() %>%
  ggplot(aes(x=weight_group_B, y = n, fill = stance_B)) +
  geom_col() +
  theme(text = element_text(size = 12), axis.text.x = element_text(angle = 45, hjust = 1),
        axis.line = element_blank(), axis.ticks = element_blank()) +
  geom_text(aes(label= n), vjust=1, size = 3) +
  labs(x = "Weight Groups", y = "# of KO's",
       title = "# of KO's per weight group",
       caption = "B-boxers") +
  guides(fill=guide_legend(title="Stances"))
```



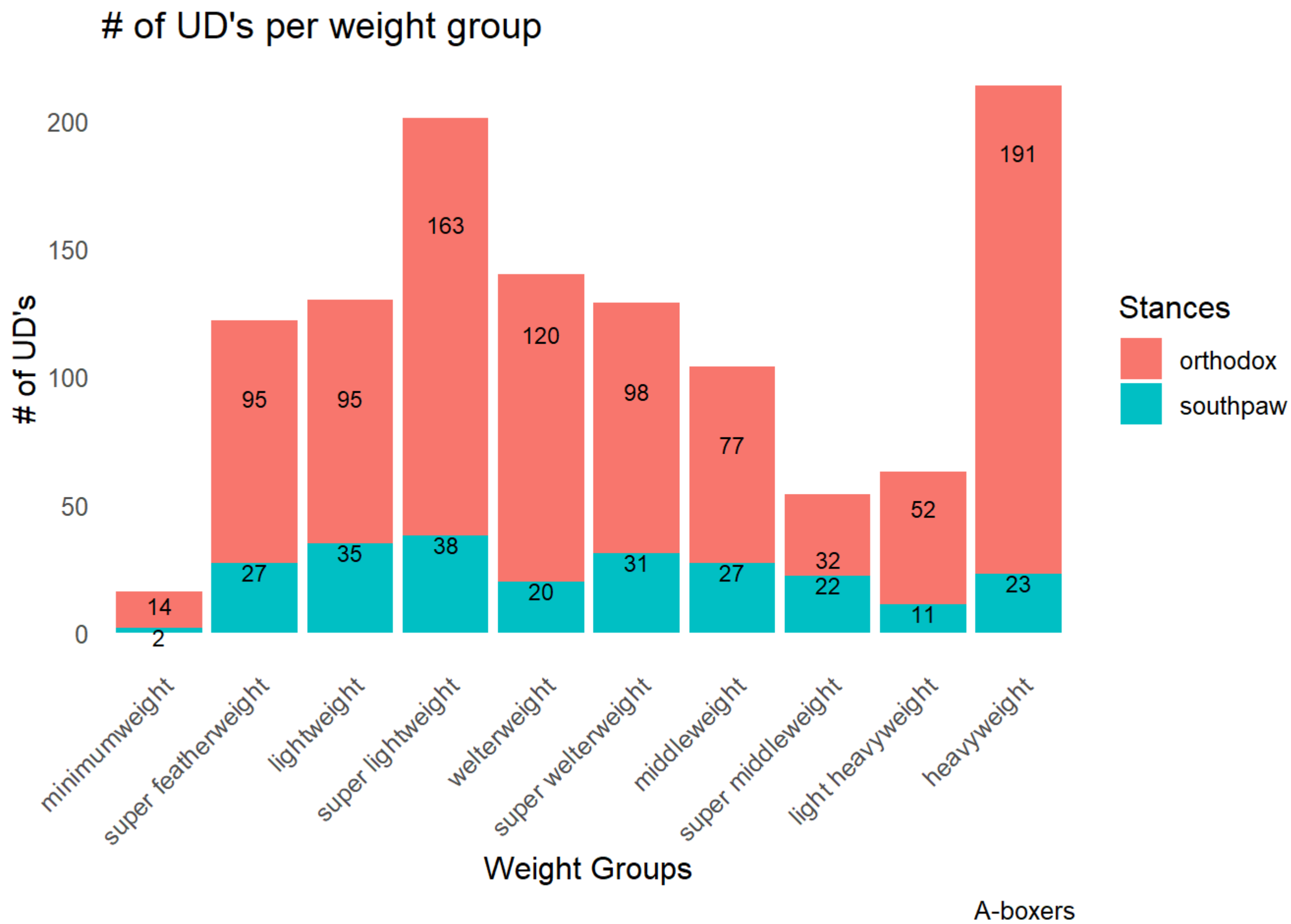

```
#TKO'S A
na.omit(boxing_copy) %>%
  group_by(decision, weight_group_A, stance_A) %>%
  filter(decision == "TKO") %>%
  tally() %>%
  ggplot(aes(x=weight_group_A, y = n, fill = stance_A)) +
  geom_col() +
  theme(text = element_text(size = 12), axis.text.x = element_text(angle = 45, hjust = 1),
        axis.line = element_blank(), axis.ticks = element_blank()) +
  geom_text(aes(label= n), vjust=1, size = 3) +
  labs(x = "Weight Groups", y = "# of TKO's",
       title = "# of TKO's per weight group",
       caption = "A-boxers") +
  guides(fill=guide_legend(title="Stances"))
```



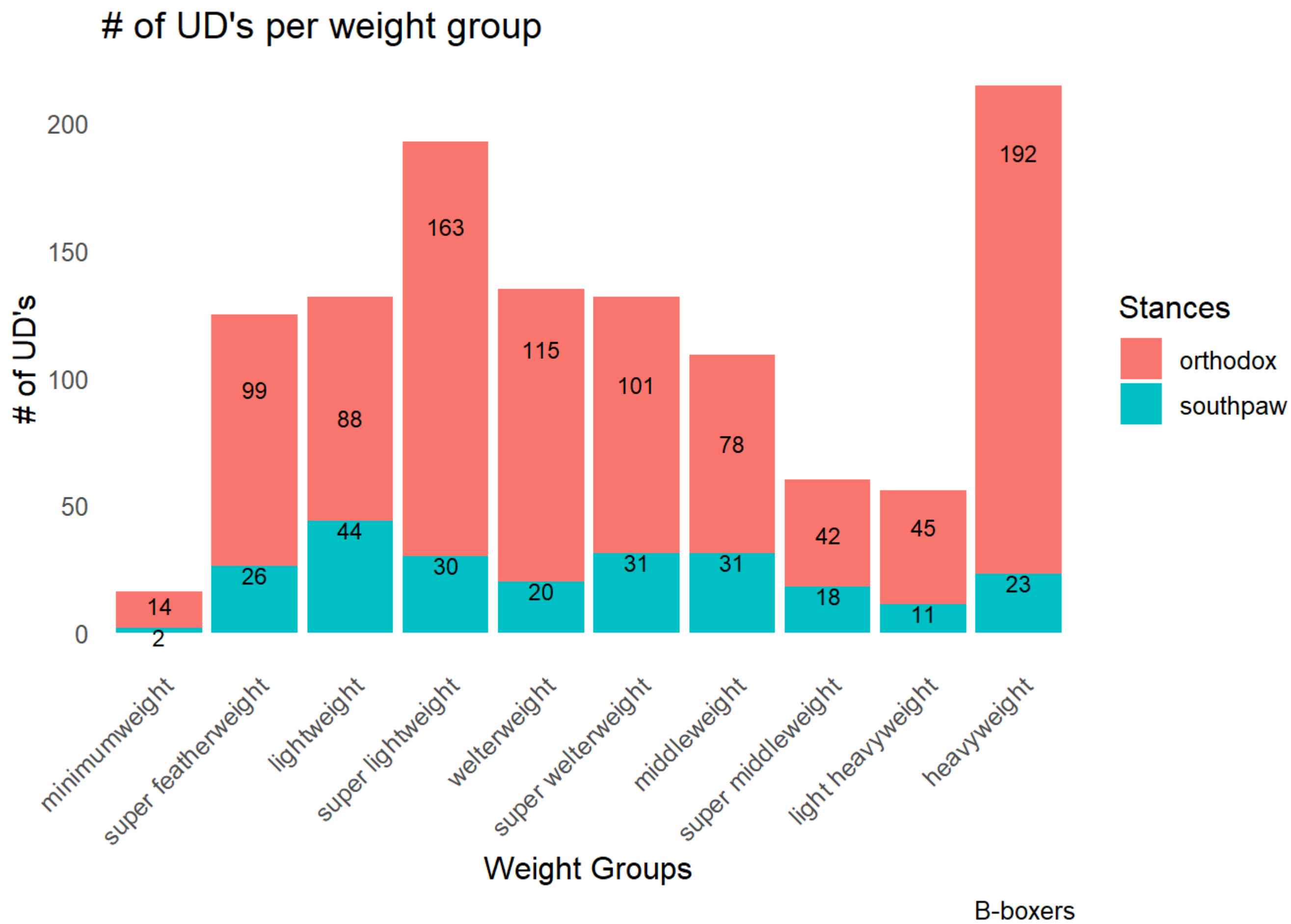

```
#TKO's B
na.omit(boxing_copy) %>%
  group_by(decision, weight_group_B, stance_B) %>%
  filter(decision == "TKO") %>%
  tally() %>%
  ggplot(aes(x=weight_group_B, y = n, fill = stance_B)) +
  geom_col() +
  theme(text = element_text(size = 12), axis.text.x = element_text(angle = 45, hjust = 1),
        axis.line = element_blank(), axis.ticks = element_blank()) +
  geom_text(aes(label= n), vjust=1, size = 3) +
  labs(x = "Weight Groups", y = "# of TKO's",
       title = "# of TKO's per weight group",
       caption = "B-boxers") +
  guides(fill=guide_legend(title="Stances"))
```




```
#UD's A
na.omit(boxing_copy) %>%
  group_by(decision, weight_group_A, stance_A) %>%
  filter(decision == "UD") %>%
  tally() %>%
  ggplot(aes(x=weight_group_A, y = n, fill = stance_A)) +
  geom_col() +
  theme(text = element_text(size = 12), axis.text.x = element_text(angle = 45, hjust = 1),
        axis.line = element_blank(), axis.ticks = element_blank()) +
  geom_text(aes(label= n), vjust= 1, size = 3) +
  labs(x = "Weight Groups", y = "# of UD's",
       title = "# of UD's per weight group",
       caption = "A-boxers") +
  guides(fill=guide_legend(title="Stances"))
```




```
#UD's B
na.omit(boxing_copy) %>%
  group_by(decision, weight_group_B, stance_B) %>%
  filter(decision == "UD") %>%
  tally() %>%
  ggplot(aes(x=weight_group_B, y = n, fill = stance_B)) +
  geom_col() +
  theme(text = element_text(size = 12), axis.text.x = element_text(angle = 45, hjust = 1),
        axis.line = element_blank(), axis.ticks = element_blank()) +
  geom_text(aes(label= n), vjust= 1, size = 3) +
  labs(x = "Weight Groups", y = "# of UD's",
       title = "# of UD's per weight group",
       caption = "B-boxers") +
  guides(fill=guide_legend(title="Stances"))
```

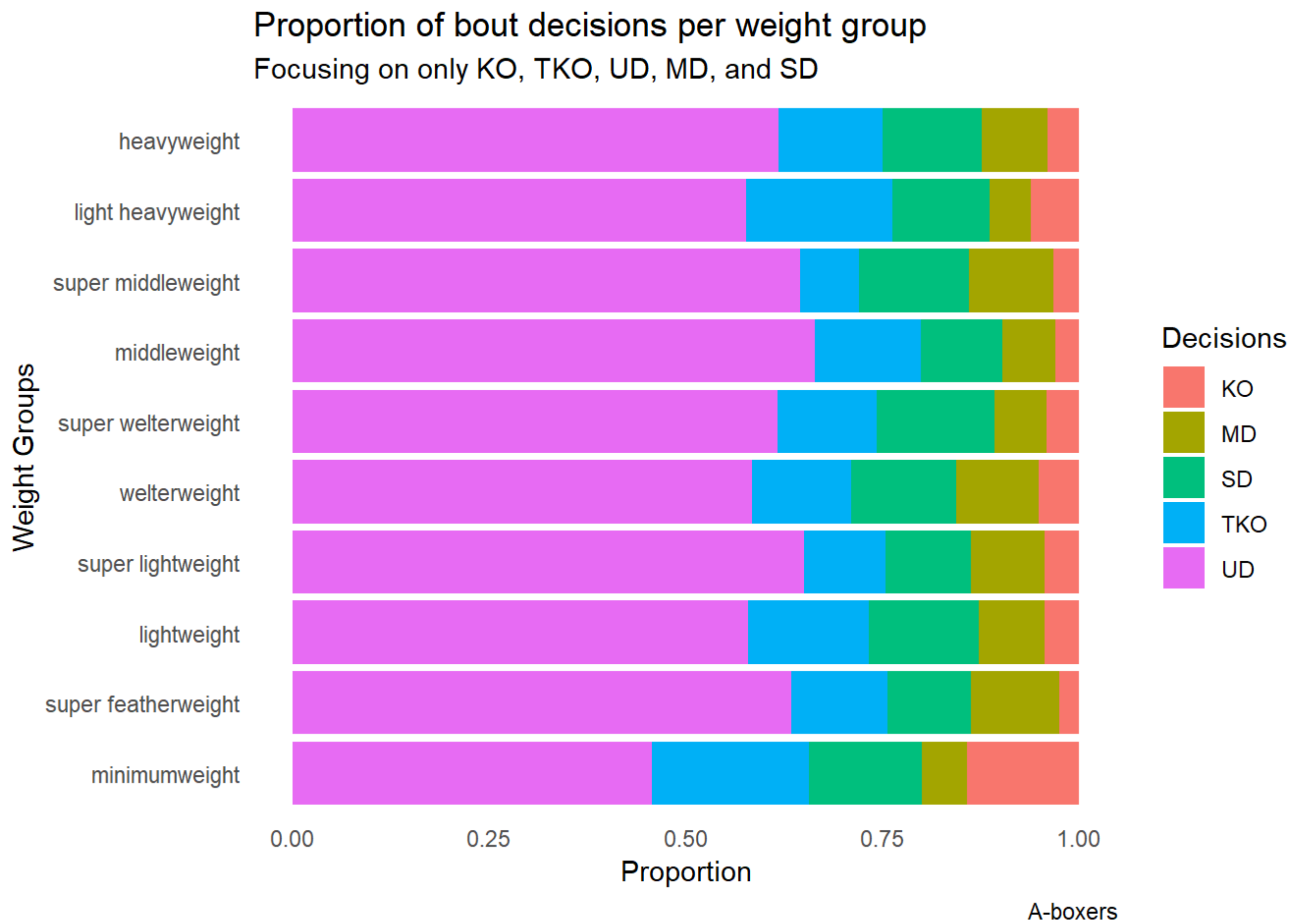


Viewing the proportion of bout decisions according to weight group

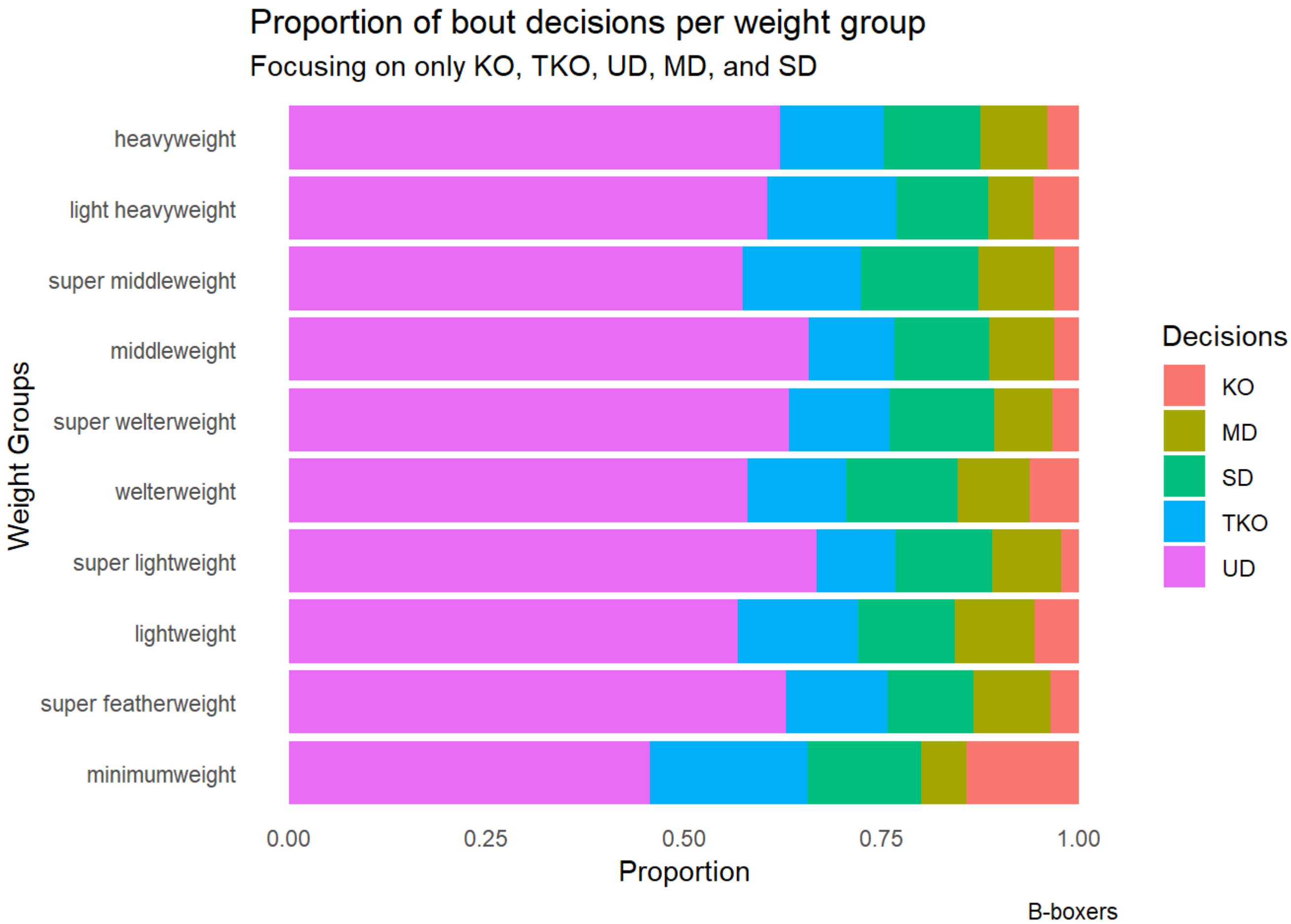
```
decision_sub <- c("KO", "TKO", "UD", "MD", "SD")

decision_df <- boxing_copy %>%
  filter(decision %in% decision_sub)

#PLOT 5
ggplot(na.omit(decision_df), aes(x = weight_group_B, fill = decision)) +
  geom_bar(position = "fill") + coord_flip() +
  labs(y = "Proportion",
       x = "Weight Groups",
       title = "Proportion of bout decisions per weight group",
       subtitle = "Focusing on only KO, TKO, UD, MD, and SD",
       caption = "A-boxers") +
  theme(axis.line = element_blank(), axis.ticks = element_blank()) +
  guides(fill=guide_legend(title="Decisions"))
```

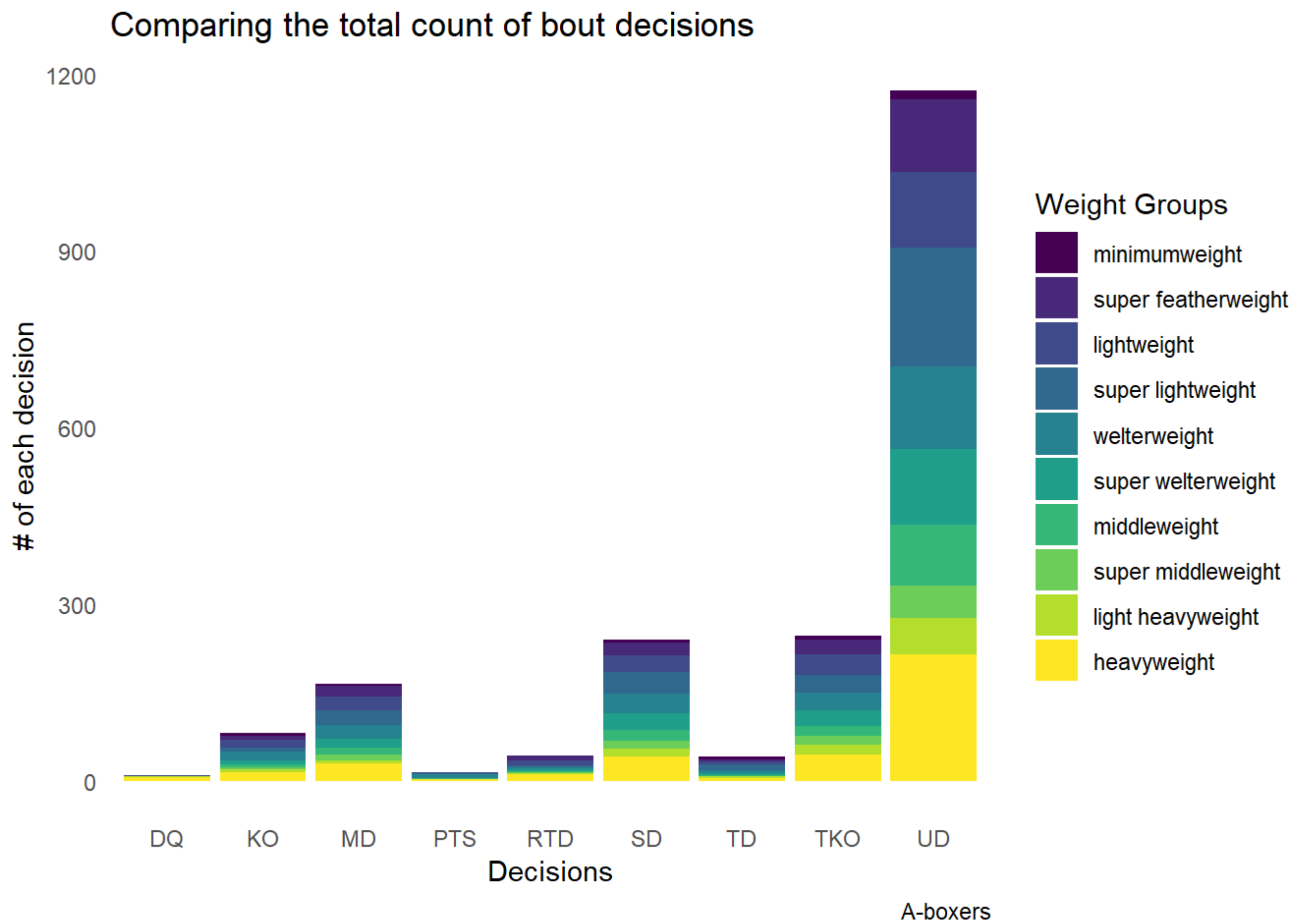



```
ggplot(na.omit(decision_df), aes(x = weight_group_A, fill = decision)) +
  geom_bar(position = "fill") + coord_flip() +
  labs(y = "Proportion",
       x = "Weight Groups",
       title = "Proportion of bout decisions per weight group",
       subtitle = "Focusing on only KO, TKO, UD, MD, and SD",
       caption = "B-boxers") +
  theme(axis.line = element_blank(), axis.ticks = element_blank()) +
  guides(fill=guide_legend(title="Decisions"))
```

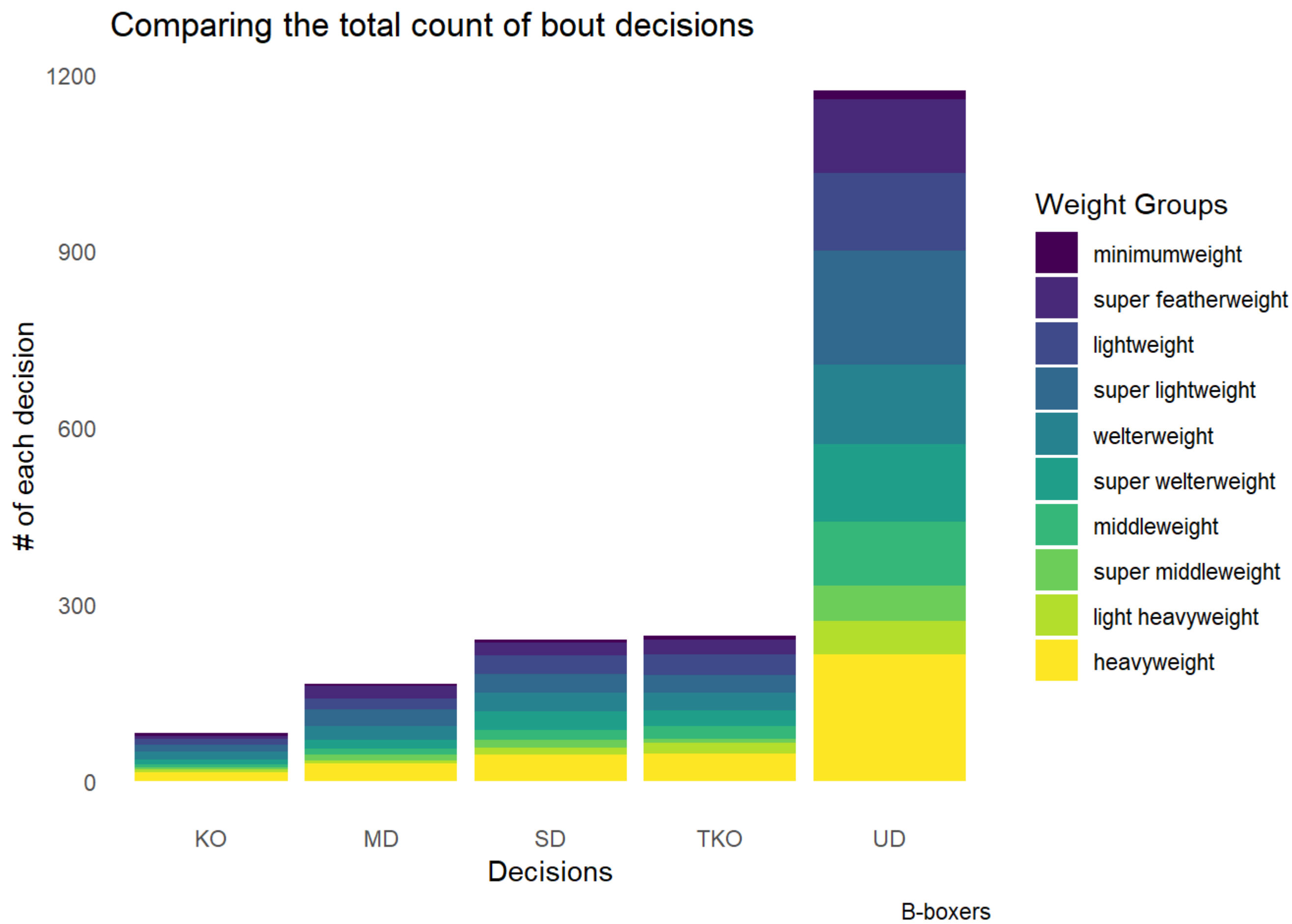


Comparing the total count of bout decisions according to weight group

```
#PLOT 6
#A-boxers
ggplot(na.omit(boxing_copy), aes(x = decision, fill = weight_group_A)) +
  geom_bar() +
  labs(x = "Decisions",
       y = "# of each decision",
       title = "Comparing the total count of bout decisions",
       caption = "A-boxers") +
  theme(axis.line = element_blank(), axis.ticks = element_blank()) +
   guides(fill=guide_legend(title="Weight Groups"))
```




```
#B-boxers
ggplot(na.omit(decision_df), aes(x = decision, fill = weight_group_B)) +
  geom_bar() +
  labs(x = "Decisions",
       y = "# of each decision",
       title = "Comparing the total count of bout decisions",
       caption = "B-boxers") +
  theme(axis.line = element_blank(), axis.ticks = element_blank()) +
   guides(fill=guide_legend(title="Weight Groups"))
```



Executive Summary:

Findings for plot 1:

For this plot, I wanted to visualize the distribution of boxers within the dataset. I also mapped the stance of boxers to the color aesthetic to visualize the different stances.

It appears as if the distribution of fighters is similar for both A-boxers and B-boxers. This dataset has a lot of heavyweight and super lightweight boxers. There isn't much of a difference between the distribution of weight groups across both boxers. As for the stance of the fighters, we see that most boxers are orthodox fighters. Orthodox fighters are fighters that lead with their left hand and use their right hand as their power punch. Southpaws lead with their right hand and use their left hand as their power punch.

Findings for plot 2:

For this plot, I wanted to see the distribution of each boxers BMI per weight group. I chose to also map stances to color to compare stance across weight groups.

From the plots, one can see that the BMI of heavyweight fighters is quite high - with BMI's ranging from low 20's to high 30's. Most people would categorize BMI's in that range as "obese", but this cannot be the case. E.g. think of the heavyweight boxer Evander Holyfield: 189 cm height, 197 cm reach, and he has fought at weights ranging from 175 lbs to over 200 lbs. His BMI was over 30 but he was certainly not obese - he had a lot of muscle mass. This is one of the shortcomings of using BMI as a metric for health since it overestimates the amount of fat in people with muscular builds. Nevertheless, it can still be used alongside other metrics.

Findings for plot 3:

For this plot, I wanted to visualize the shape of the distribution to see whether or not the distribution was symmetrical, whether it was bimodal or not, etc.

For the lower weight groups (e.g. lightweight) it appears as if the shape is very narrow - which tells me that the BMI within that weight group doesn't have a lot of variation. But for the heavier weight groups (e.g. heavyweights), the shape is much more broad - which tell me that there is greater variation within those groups.

Findings for plot 4:

I created these plots to see if there were any differences in the amount of KO's, TKO's, etc among weight groups. The plots reveal that heavyweights had the most KO's, TKO's, and UD's, but that is mainly due to a large number of heavyweights within this dataset. One can also see the large number of bouts that were UD. This is something I wanted to look into, so I wanted to look at their proportions across weight groups.

Findings for plot 5:

For this plot, I wanted to visualize the proportion, instead of the count of decisions.

This plot allowed me to see the proportion of bout decisions across weight groups. The plot reveals that the most common type of decision is a unanimous decision (UD). Which tells me that most boxing matches went the distance - i.e. fighters usually fight all 12 rounds and don't get knocked out. I think this at least shows that boxing is much more than a mindless brawl that ends quickly, but oftentimes an endurance sport that requires the strategic use of one's energy. To quickly verify my hypothesis that most fights go the distance, I want to visualize the bout decisions to compare the total counts of each decision. I'll also map the weight groups onto the plot to see if there are any differences across weight groups - e.g. what happens if KO's are more common in the lighter weight groups?

Findings for plot 6:

For this plot, I wanted to confirm my hypothesis that most boxing matches last 12 rounds; or that most of them don't end in a knockout.

This plot confirms my hypothesis. The most common type of decisions are the ones that require boxers to box all 12 rounds (UD, SD, MD). Note: TKO's don't always go the distance since the fight by definition ended early. The difference between a knockout and technical knockout is the referee who decides to end the fight early - usually because the fighter losing (about to be knocked out "technically") is in danger of being badly hurt. Knockouts are different because it is one of the boxers who ends the fight early by knocking the other fighter out. For this reason, I will not include TKO's among the decisions that require fighters going the distance. Nevertheless, from this plot, we still see that knockouts (TKO, KO), boxers quitting (RTD), and boxers being disqualified (DQ), are not the most common type of decision. This goes to further show that boxing is a technical sport that requires strategy and endurance, not mindless bunches of punches.

Here is an app I made to explore the dataset interactively:

https://aponted.shinyapps.io/applications/ (https://aponted.shinyapps.io/applications/)

The web application allows users to change between A-boxers and B-boxers; as well as the ability to filter the bout decisions and the total counts across features. There also is a data frame displayed at the bottom of the exploratory plot, in which a user can hover above points to inquire about a particular boxing bout and its various features.

```
library(shiny)
library(DT)
```

```
##
## Attaching package: 'DT'
```

```
## The following objects are masked from 'package:shiny':
##
##     dataTableOutput, renderDataTable
```

```
#UI
ui_boxing <- fluidPage(
  titlePanel("Boxing Dataset Visualization"),

  sidebarLayout(

    sidebarPanel(
      #SELECTING A OR B BOXERS for plot 1
      selectInput(
        inputId = "plot1",
        label = "Distribution Barplot",
        choices = c("Weight of A" = "weight_group_A",
                    "Weight of B" = "weight_group_B"),
        selected = "weight_group_A"),
      br(), br(),
      #SELECTING A OR B BOXERS for plot 2
      selectInput(
        inputId = "plot2",
        label = "Boxplot for BMI",
        choices = c("Weight of A" = "weight_group_A",
                    "Weight of B" = "weight_group_B",
                    "Stance A" = "stance_A",
                    "Stance B" = "stance_B"),
        selected = "weight_group_A"),
      br(), br(),
      #SELECTING A OR B BOXERS for plot 3
      selectInput(
        inputId = "plot3",
        label = "Density plot for BMI",
        choices = c("Weight of A" = "weight_group_A",
                    "Weight of B" = "weight_group_B",
                    "Stance A" = "stance_A",
                    "Stance B" = "stance_B"),
        selected = "weight_group_A"),
      br(), br(),
      #SELECTING BOUT DECISION
      selectInput(
        inputId = "decision",
        label = "Choose decision for bar plot",
        choices = c("Knockout" = "KO",
                    "Technical knockout" = "TKO",
                    "Unanimous decision" = "UD",
                    "Majority decision" = "MD",
                    "Split decision" = "SD"),
        selected = "KO"),
      br(), br(),
      #decision plot
      selectInput(
        inputId = "plot5",
        label = "Proportion of bout decisions",
        choices = c("Weight of A" = "weight_group_A",
                    "Weight of B" = "weight_group_B"),
        selected = "weight_group_A"),
```

```
      br(), br(),
      #SELECTING X-AXIS
      selectInput(
        inputId = "x",
        label = "X-axis",
        choices = c("Age of A" = "age_A",
                    "Age of B" = "age_B",
                    "BMI of A" = "bmi_A",
                    "BMI of B" = "bmi_B",
                    "Reach of A" = "reach_A",
                    "Reach of B" = "reach_B",
                    "Height of A (cm)" = "height_A",
                    "Height of B (cm)" = "height_B",
                    "Weight of A (lbs)" = "weight_A",
                    "Weight of B (lbs)" = "weight_B"),
        selected = "bmi_A"),
      br(), br(),
      #SELECTING Y-AXIS
      selectInput(
        inputId = "y",
        label = "Y-axis",
        choices = c("Age of A" = "age_A",
                    "Age of B" = "age_B",
                    "BMI of A" = "bmi_A",
                    "BMI of B" = "bmi_B",
                    "Reach of A" = "reach_A",
                    "Reach of B" = "reach_B",
                    "Height of A (cm)" = "height_A",
                    "Height of B (cm)" = "height_B",
                    "Weight of A (lbs)" = "weight_A",
                    "Weight of B (lbs)" = "weight_B"),
        selected = "age_A")),

    mainPanel(
      plotOutput(outputId = "barplot"),
      br(), br(),
      plotOutput(outputId = "boxplot"),
      br(), br(),
      plotOutput(outputId = "densityplot"),
      br(), br(),
      plotOutput(outputId = "barplot2"),
      br(), br(),
      plotOutput(outputId = "proportionplot"),
      br(), br(),
      plotOutput(outputId = "scatterplot", hover = "plot_hover"),
      br(), br(),
      dataTableOutput(outputId = "boxingdf")
    )
  )
)

server_boxing <- function(input, output) {

  output$barplot <- renderPlot({
```

```
    #plot 1
    na.omit(boxing_copy) %>%
      ggplot(aes_string(x = input$plot1, fill = "stance_A")) +
      geom_bar() +
      labs(x = "Weight Groups", y = "# of boxers in dataset", title = "Distribution Barplot") +
      coord_flip() +
      guides(fill=guide_legend(title="Stances")) +
      theme(axis.line = element_blank(), axis.ticks = element_blank()) +
      geom_text(aes(label=..count..),stat="count",position=position_stack(0.5), size = 3)
  })

  #plot 2
  output$boxplot <- renderPlot( {
    ggplot(na.omit(boxing_copy), aes_string(x = input$plot2, y = "bmi_A", fill = "stance_A")) +
      geom_boxplot() +
      theme(text = element_text(size= 12), axis.text.x = element_text(angle = 45, hjust = 1),
            axis.line = element_blank(), axis.ticks = element_blank()) +
      labs(x = "Weight Groups", y = "BMI", title = "Boxplot for BMI",
           subtitle = "The body mass index (BMI) calculates body fat based on weight and height"
) +
      guides(fill=guide_legend(title="Stances"))
  })

  #plot 3
  output$densityplot <- renderPlot( {
    ggplot(na.omit(boxing_copy), aes(x = bmi_A)) +
      geom_density(aes_string(fill = input$plot3), alpha = 0.7) +
      labs(x = "Body Mass Index",
           title = "Density Plot for BMI") +
      guides(fill=guide_legend(title="Weight Groups or Stances")) +
      theme(axis.line = element_blank(), axis.ticks = element_blank())
  })

  #plot 4
  output$barplot2 <- renderPlot( {

    na.omit(boxing_copy) %>%
      group_by(decision, weight_group_A, stance_A) %>%
      filter(decision == input$decision) %>%
      tally() %>%
      ggplot(aes_string(x= "weight_group_A", y = "n", fill = "stance_A")) +
      geom_col() +
      theme(text = element_text(size = 12), axis.text.x = element_text(angle = 45, hjust = 1),
            axis.line = element_blank(), axis.ticks = element_blank()) +
      geom_text(aes(label= n), vjust=1, size = 3) +
      labs(x = "Weight Groups", y = "# of decisions",
           title = "# of decisions per weight group") +
      guides(fill = guide_legend(title = "Stances"))
  })

  #plot 5
  output$proportionplot <- renderPlot({
    ggplot(na.omit(decision_df), aes_string(x = input$plot5, fill = "decision")) +
      geom_bar(position = "fill") + coord_flip() +
```

```
      labs(x = "Weight Groups", y = "Proportion", title = "Barplot for the proportion of bout de
cisions",
           subtitle = "Focusing on only KO, TKO, UD, MD, and SD bout decisions") +
      theme(axis.line = element_blank(), axis.ticks = element_blank())
  })

  #plot 6
  output$scatterplot <- renderPlot({
    ggplot(boxing_copy, aes_string(x = input$x, y = input$y)) +
      geom_point() +
      labs(x = "X variable", y = "Y variable", title = "Explore the variables within the dataset
 by changing the X and Y axis") +
      theme(axis.line = element_blank(), axis.ticks = element_blank())
  })

  output$boxingdf <- DT::renderDataTable({
    nearPoints(boxing_copy, coordinfo = input$plot_hover) %>%
      select("Age of A" = age_A, "Age of B" = age_B, "BMI of A" = bmi_A, "BMI of B" = bmi_B, "Re
ach of A" = reach_A, "Reach of B" = reach_B, "Height of A" = height_A, "Height of B" = height_B,
 "Weight of A" = weight_A, "Weight of B" = weight_B)
  })
}

shinyApp(ui = ui_boxing, server = server_boxing)
```

Boxing Dataset Visualization

Distribution Barplot

Weight of A

Boxplot for BMI

Weight of A

Density plot for BMI

Weight of A