

# baldwin wallace EDA

2025-07-02

```
library("readr")
library("dplyr")
library("ggplot2")
library("readr")
library("stringr")
library("glue")
```

```
g <- params$category
singular_game <- readr::read_csv(glue("Desktop/SURA project code/extended_cmu_data/extended_cmu_data_",
```

```
## New names:
## Rows: 20 Columns: 22
## -- Column specification
## ----- Delimiter: "," chr
## (1): LINEUP (NAMES) dbl (20): ...1, NUMBER OF GUARDS, OPPONENT POSSESSIONS, CMU
## POSSESSIONS, OP... time (1): LINEUP MINUTES
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `` -> `...1`
```

```
# if negatives in any columns (specifically had problem in possession column)
for (colName in colnames(singular_game)){
  singular_game[[colName]][singular_game[[colName]] < 0] <- 0
}
```

```
singular_game$`LINEUP MINUTES` <- sapply(singular_game$`LINEUP MINUTES`, function(t){
  parts <- as.integer(strsplit(as.character(t),":")[[1]])
  parts[1]*60 + parts[2]
})
```

```
singular_game <- singular_game %>% rename(`LINEUP SECONDS` = `LINEUP MINUTES`) %>% mutate(LINEUP_SORTED =
  if (is.na(1)) return(NA)
  paste(sort(strsplit(1, ", ")[1]), collapse = " "))
}))
```

```
game <- singular_game %>% group_by(`LINEUP_SORTED`) %>% summarise(
  `NUMBER OF GUARDS` = mean(`NUMBER OF GUARDS`),
  `OPPONENT POSSESSIONS` = sum(`OPPONENT POSSESSIONS`, na.rm = TRUE),
  `CMU POSSESSIONS` = sum(`CMU POSSESSIONS`, na.rm = TRUE),
  `LINEUP SECONDS` = sum(`LINEUP SECONDS`, na.rm = TRUE),
  `OPPONENT PTS` = sum(`OPPONENT PTS`, na.rm = TRUE),
  `CMU PTS` = sum(`CMU PTS`, na.rm = TRUE),
  `CMU 3PA` = sum(`CMU 3PA`, na.rm = TRUE),
  `CMU FGA` = sum(`CMU FGA`, na.rm = TRUE),
  `CMU FTA` = sum(`CMU FTA`, na.rm = TRUE),
  `CMU REBOUNDS` = sum(`CMU REBOUNDS`, na.rm = TRUE),
  `TOTAL REBOUNDS` = sum(`TOTAL REBOUNDS`, na.rm = TRUE),
```

```

`SCORE DIFFERENTIAL WHEN ENTER` = paste(`SCORE DIFFERENTIAL WHEN ENTER`, collapse = ", "),
`QUARTER` = paste(`QUARTER`, collapse = ", ")
) %>%mutate(`PACE` = 40 * ((`CMU POSSESSIONS` + `OPPONENT POSSESSIONS`) / (2 * `LINEUP SECONDS`/60)),
`OFFENSIVE RATING` = 100 * (`CMU PTS` / `CMU POSSESSIONS`),
`DEFENSIVE RATING` = 100 * (`OPPONENT PTS` / `OPPONENT POSSESSIONS`),
`NET RATING` = `OFFENSIVE RATING` - `DEFENSIVE RATING`,
`3PA/FGA` = `CMU 3PA` / `CMU FGA`,
`TRUE SHOOTING %` = 100 * (`CMU PTS` / (2 * (`CMU FGA` + (0.44* `CMU FTA`)))),
`TRB%` = 100 * (`CMU REBOUNDS` / `TOTAL REBOUNDS`)

```

```

# see where to score differential cut off time -> SHOULD DO THIS AFTER OR BEFORE CUT SCRAP MINUTES?
l <- quantile(singular_game$`SCORE DIFFERENTIAL WHEN ENTER`,probs=c(0.1))
u <- quantile(singular_game$`SCORE DIFFERENTIAL WHEN ENTER`,probs=c(0.9))

```

```
l
```

```
## 10%
```

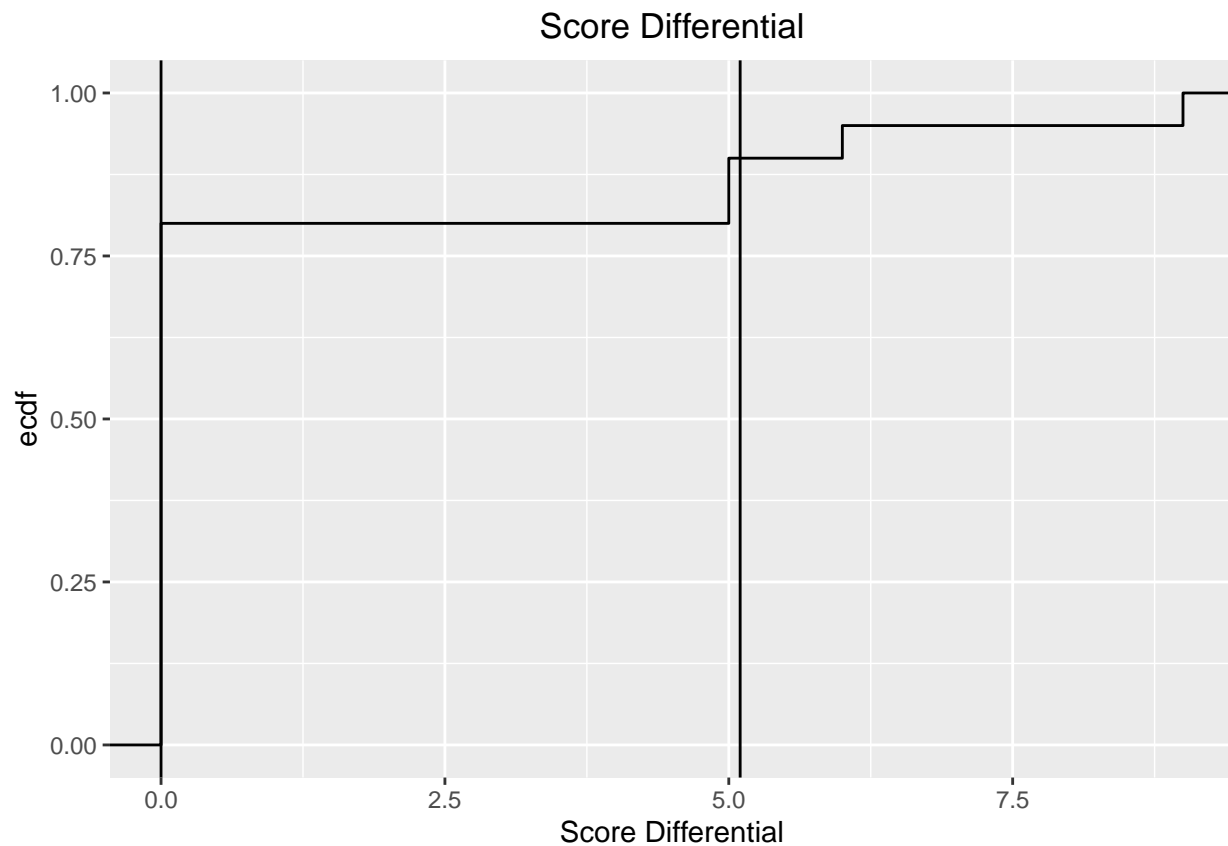
```
## 0
```

```
u
```

```
## 90%
```

```
## 5.1
```

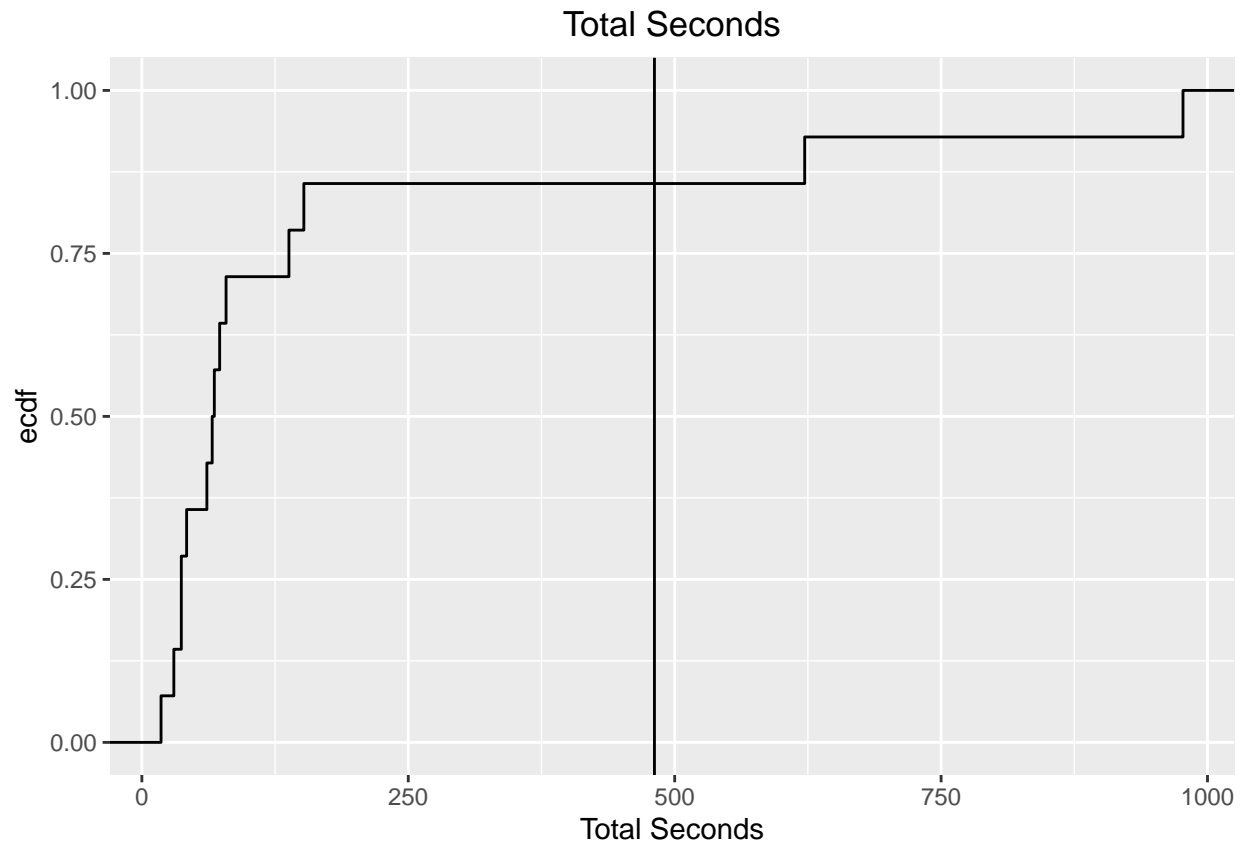
```
ggplot(singular_game, aes(x = `SCORE DIFFERENTIAL WHEN ENTER`)) + stat_ecdf() + geom_vline(xintercept =
```



```
game <- subset(game, !((`SCORE DIFFERENTIAL WHEN ENTER` <= -11 | `SCORE DIFFERENTIAL WHEN ENTER` >= 15))
```

```
# see where to cut time -> SHOULD DO THIS AFTER OR BEFORE CUT SCRAP MINUTES?
```

```
p <- quantile(game$`LINEUP SECONDS`, probs=c(0.9))
ggplot(game, aes(x = `LINEUP SECONDS`)) + stat_ecdf() + geom_vline(xintercept = p) + labs(title = "Total
```



```
#game <- subset(game, `LINEUP SECONDS` >= p)

p

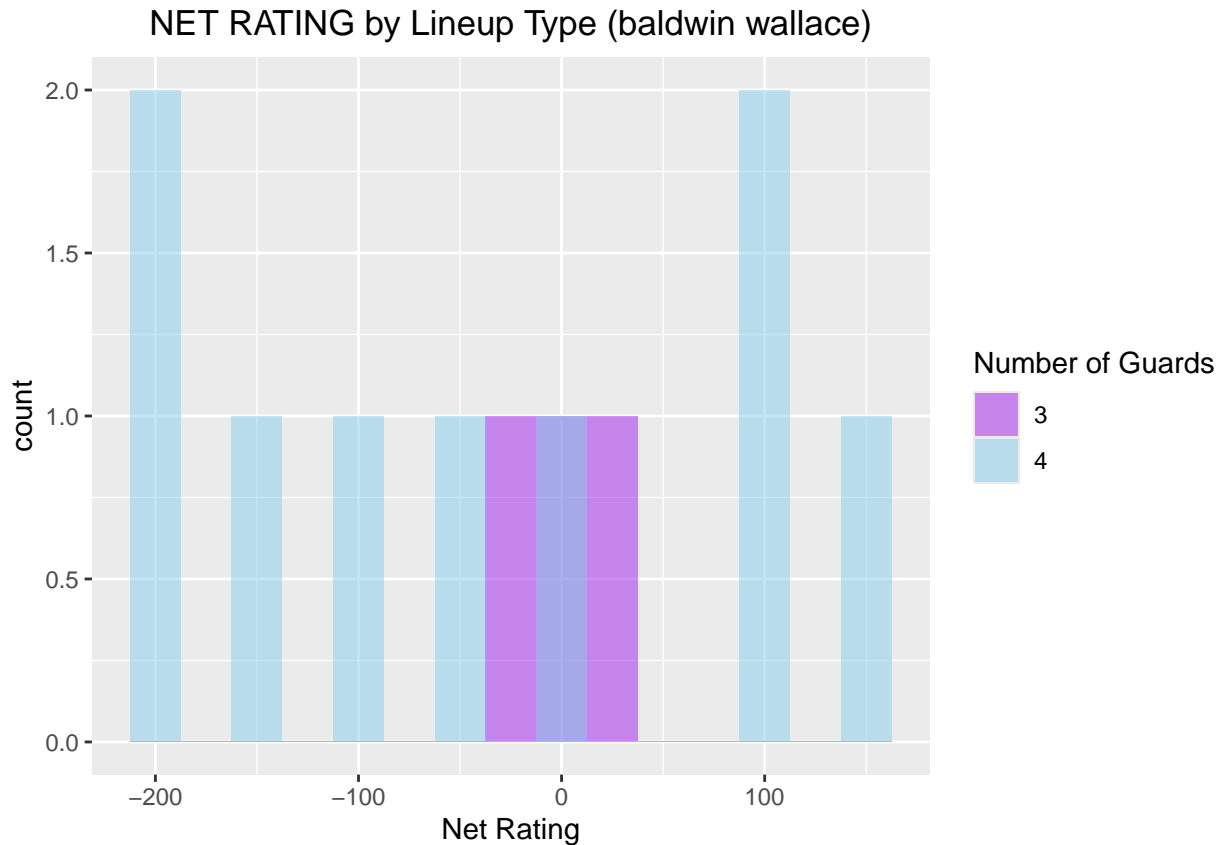
## 90%
## 481

#pdf(file = glue("Desktop/SURA project code/sing_game_EDA/{g}_plot.pdf"), width = 6, height = 5)

t_f <- c("3", "4")

ggplot(data = subset(game, subset = `NUMBER OF GUARDS` %in% t_f), aes(x = `NET RATING`, fill = factor(`

## Warning: Removed 2 rows containing non-finite outside the scale range
## (`stat_bin()`).
```



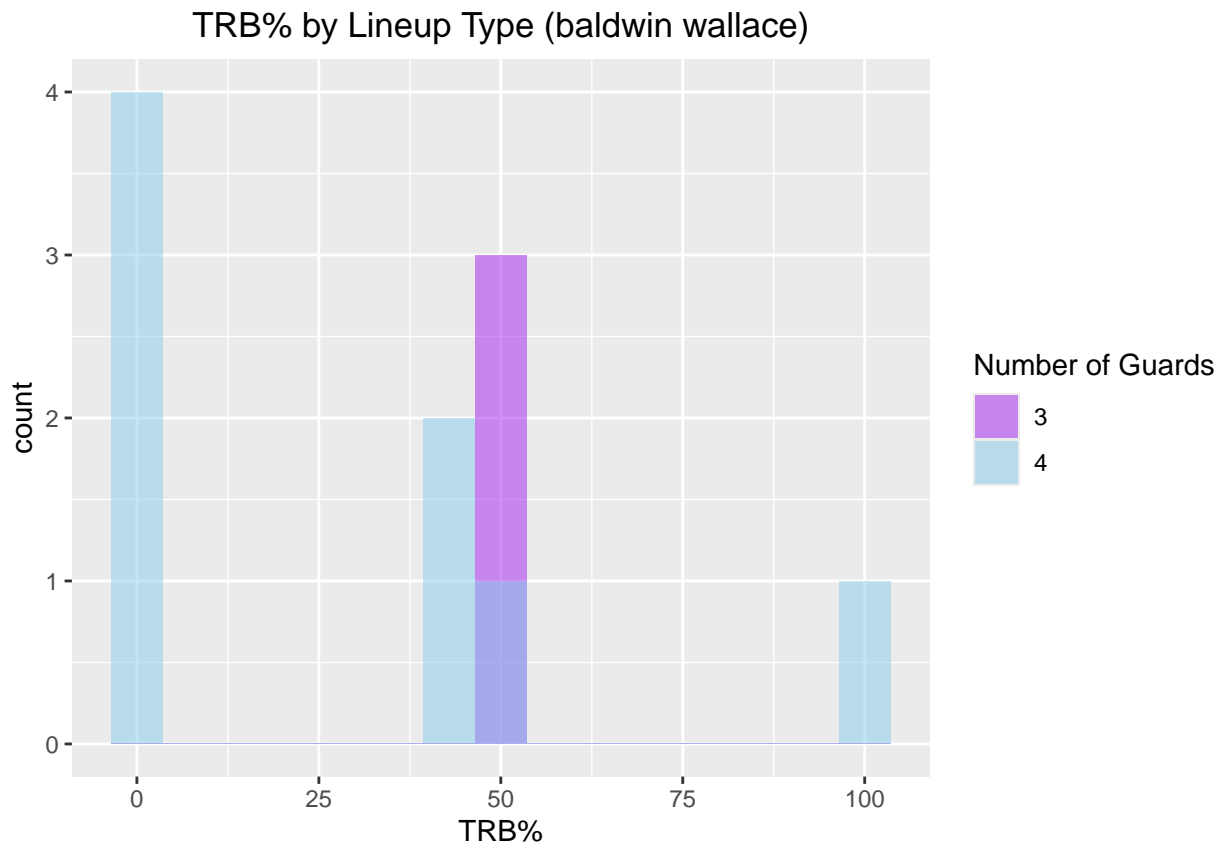
```
tapply(game$`NET RATING`[game$`NUMBER OF GUARDS` %in% t_f], game$`NUMBER OF GUARDS`[game$`NUMBER OF GUARDS` %in% t_f],
  FUN = function(x) {
    ## $`3`
    ##      Min. 1st Qu.  Median      Mean 3rd Qu.     Max.      NA's
    ## -30.7190 -15.3595   0.0000  -0.2397  15.0000  30.0000         1
    ##
    ## $`4`
    ##      Min. 1st Qu.  Median      Mean 3rd Qu.     Max.      NA's
    ##    -200    -160     -50     -40    100    150         1
  },
  simplify = FALSE)

wilcox.test(`NET RATING` ~ `NUMBER OF GUARDS`, data = subset(game, `NUMBER OF GUARDS` %in% t_f), exact = FALSE)

##
## Wilcoxon rank sum test with continuity correction
##
## data:  NET RATING by NUMBER OF GUARDS
## W = 16.5, p-value = 0.6422
## alternative hypothesis: true location shift is not equal to 0

ggplot(data = subset(game, subset = `NUMBER OF GUARDS` %in% t_f), aes(x = `TRB%`, fill = factor(`NUMBER OF GUARDS`))) +
  facet_wrap(~ `NUMBER OF GUARDS`)

## Warning: Removed 3 rows containing non-finite outside the scale range
## (`stat_bin()`).
```



```
tapply(game$`TRB%`[game$`NUMBER OF GUARDS` %in% t_f], game$`NUMBER OF GUARDS`[game$`NUMBER OF GUARDS` %in% t_f], FUN = function(x) {
  summary(x)
})
```

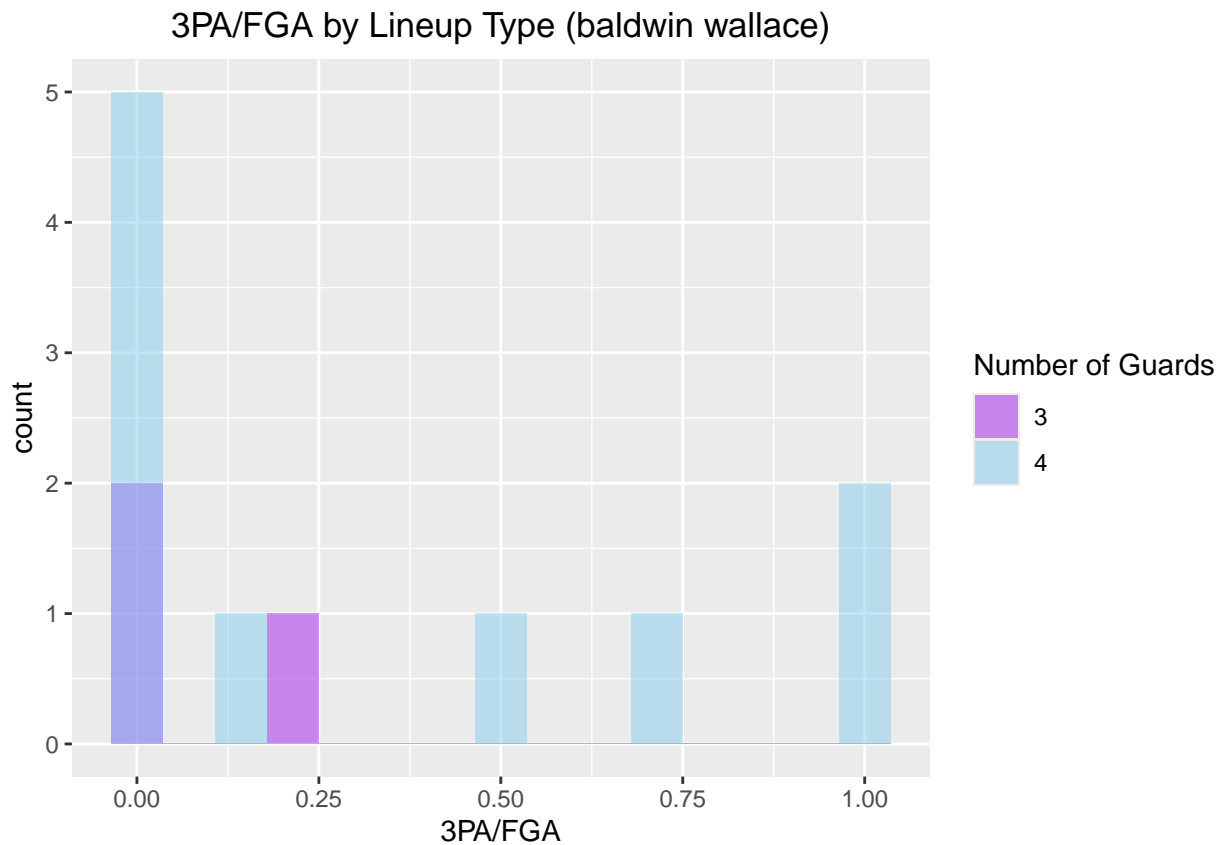
```
## $`3`
##      Min. 1st Qu.  Median      Mean 3rd Qu.     Max.    NA's
##       50      50      50       50      50      50         1
##
## $`4`
##      Min. 1st Qu.  Median      Mean 3rd Qu.     Max.    NA's
##       0.00    0.00   20.00   28.96   43.75   100.00         2
```

```
wilcox.test(`TRB%` ~ `NUMBER OF GUARDS`, data = subset(game, `NUMBER OF GUARDS` %in% t_f), exact = FALSE)
```

```
##
## Wilcoxon rank sum test with continuity correction
##
## data:  TRB% by NUMBER OF GUARDS
## W = 19.5, p-value = 0.134
## alternative hypothesis: true location shift is not equal to 0
```

```
ggplot(data = subset(game, subset = `NUMBER OF GUARDS` %in% t_f), aes(x = `3PA/FGA`, fill = factor(`NUMBER OF GUARDS`)))
```

```
## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_bin()`).
```



```
tapply(game$`3PA/FGA`[game$`NUMBER OF GUARDS` %in% t_f], game$`NUMBER OF GUARDS`[game$`NUMBER OF GUARDS`
```

```
## $`3`
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   NA's
## 0.00000 0.00000 0.00000 0.06667 0.10000 0.20000     1
##
```

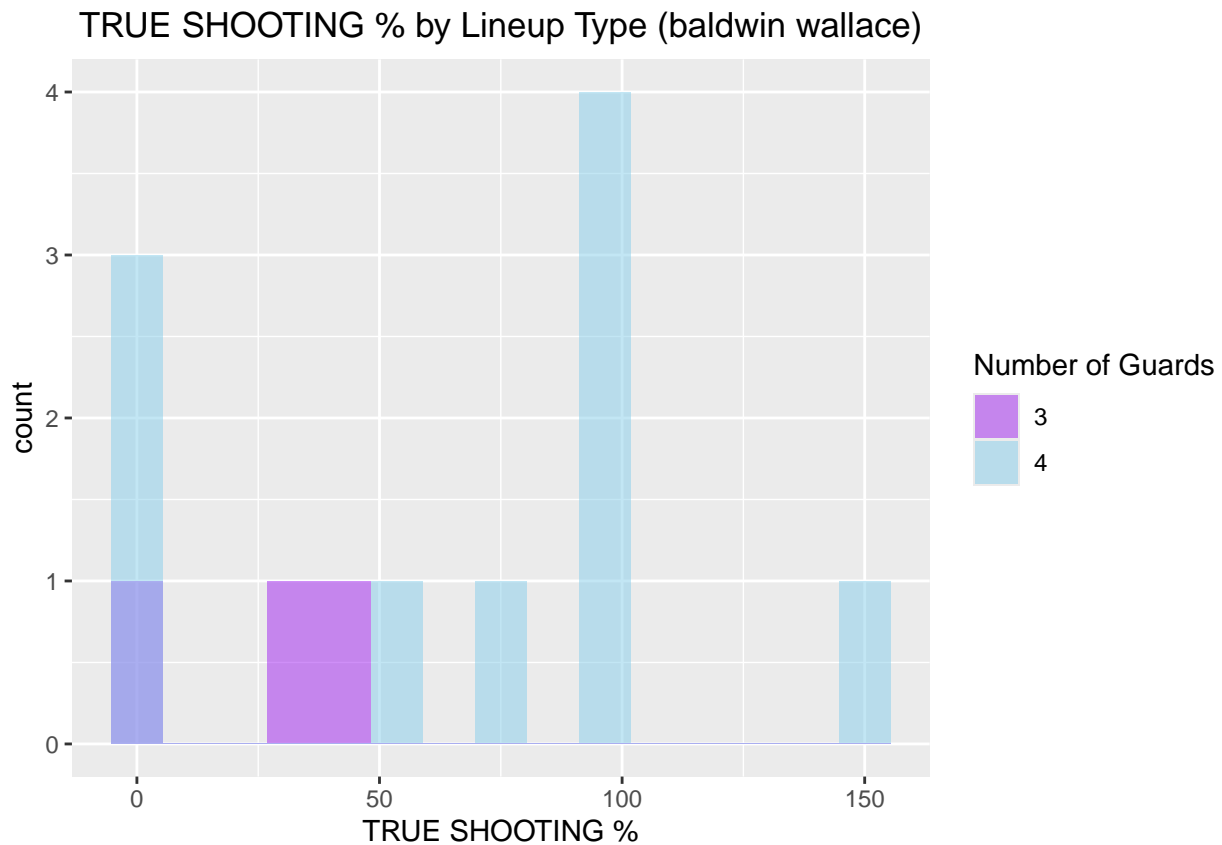
```
## $`4`
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00000 0.00000 0.08696 0.34239 0.68750 1.00000
```

```
wilcox.test(`3PA/FGA` ~ `NUMBER OF GUARDS`, data = subset(game, `NUMBER OF GUARDS` %in% t_f), exact = F
```

```
##
## Wilcoxon rank sum test with continuity correction
##
## data: 3PA/FGA by NUMBER OF GUARDS
## W = 11, p-value = 0.5195
## alternative hypothesis: true location shift is not equal to 0
```

```
ggplot(data = subset(game, subset = `NUMBER OF GUARDS` %in% t_f), aes(x = `TRUE SHOOTING %`, fill = fac
```

```
## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_bin()`).
```



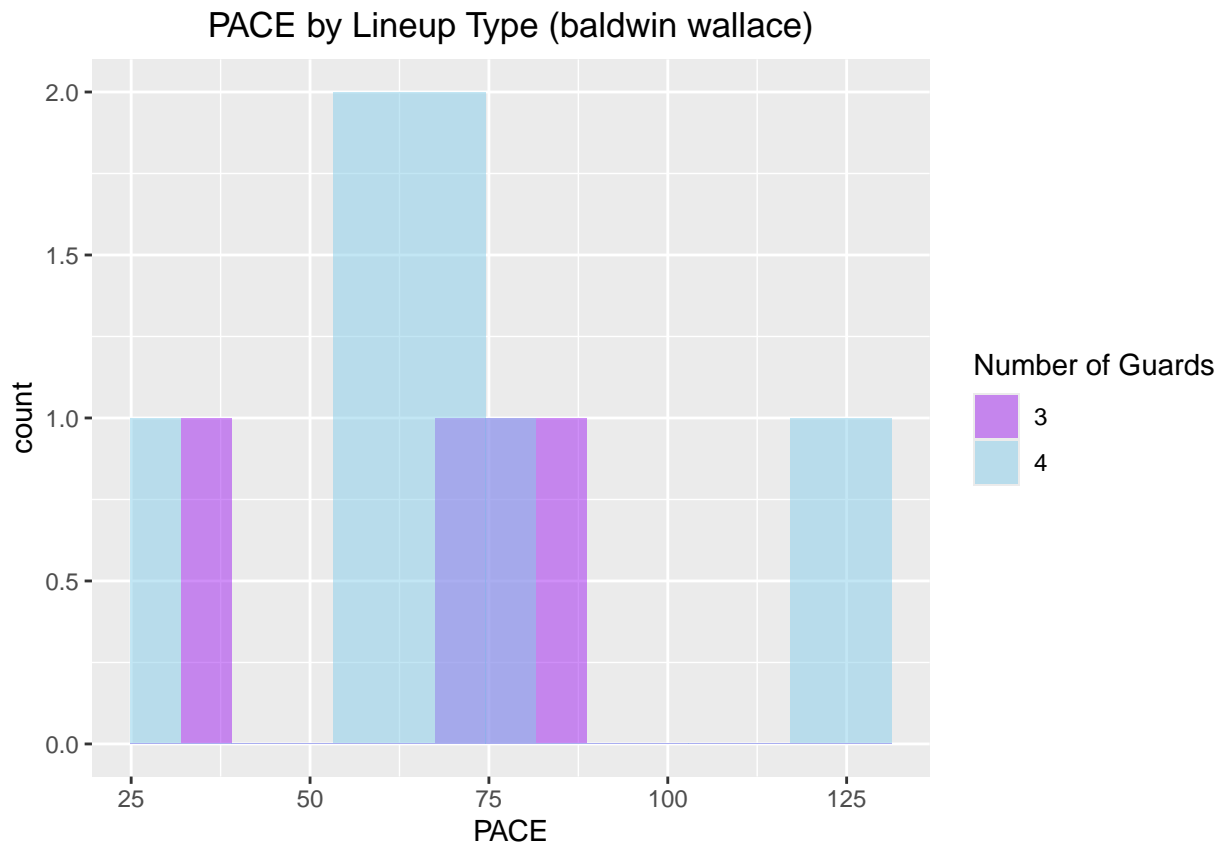
```
tapply(game$TRUE SHOOTING % [game$NUMBER OF GUARDS %in% t_f], game$NUMBER OF GUARDS [game$NUMBER OF GUARDS %in% t_f], FUN = function(x) {
  summary(x)
})
```

```
## $`3`
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.      NA's
##      0.00  15.53   31.06   23.69  35.53   40.00         1
##
## $`4`
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      0.00  13.33   87.50   67.83  100.00   150.00
```

```
wilcox.test(TRUE SHOOTING % ~ NUMBER OF GUARDS, data = subset(game, NUMBER OF GUARDS %in% t_f), exact = FALSE)
```

```
##
## Wilcoxon rank sum test with continuity correction
##
## data: TRUE SHOOTING % by NUMBER OF GUARDS
## W = 7.5, p-value = 0.2236
## alternative hypothesis: true location shift is not equal to 0
```

```
ggplot(data = subset(game, subset = NUMBER OF GUARDS %in% t_f), aes(x = PACE, fill = factor(NUMBER OF GUARDS)))
```



```
tapply(game$`PACE`[game$`NUMBER OF GUARDS` %in% t_f], game$`NUMBER OF GUARDS`[game$`NUMBER OF GUARDS` %in% t_f],
```

```
## $`3`
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  32.43  58.75   73.11   66.37  80.73   86.84
##
```

```
## $`4`
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  30.38  59.30   68.34   74.31  76.34  129.73
```

```
wilcox.test(`PACE` ~ `NUMBER OF GUARDS`, data = subset(game, `NUMBER OF GUARDS` %in% t_f), exact = FALSE)
```

```
##
## Wilcoxon rank sum test with continuity correction
##
```

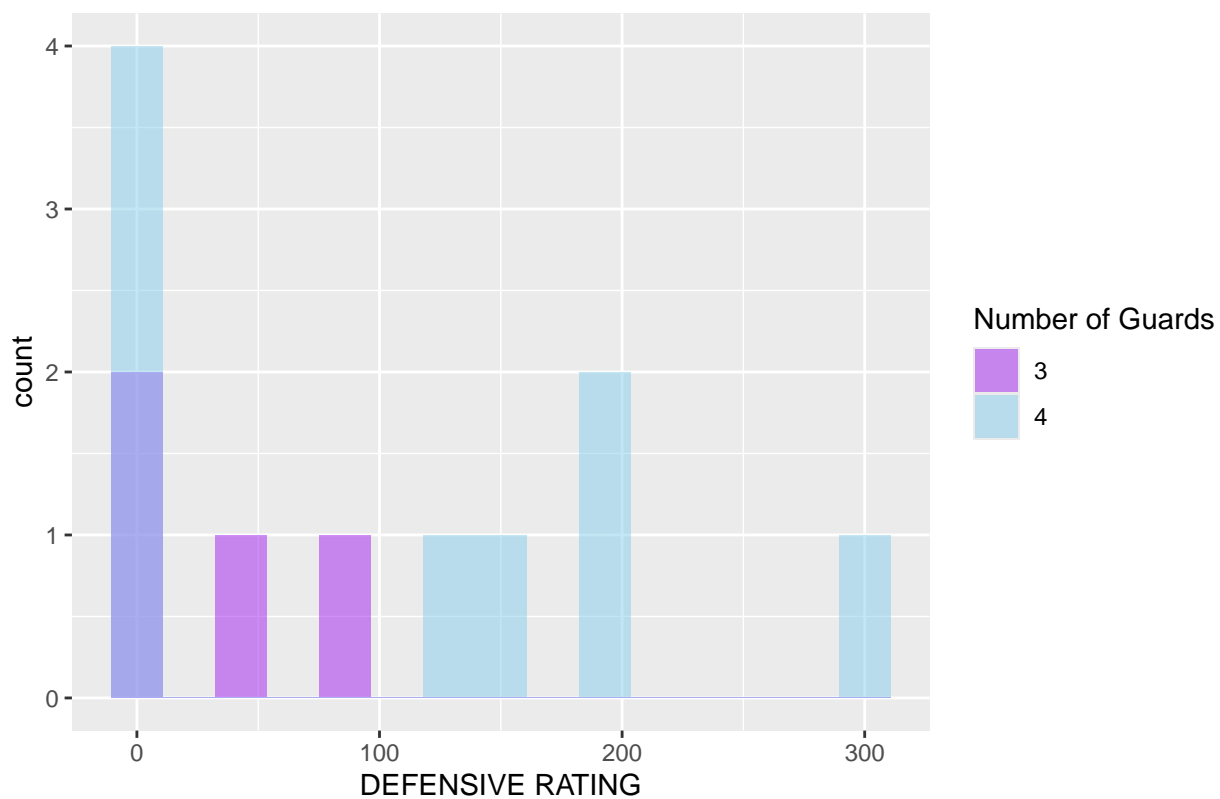
```
## data: PACE by NUMBER OF GUARDS
## W = 22, p-value = 0.832
## alternative hypothesis: true location shift is not equal to 0
```

```
ggplot(data = subset(game, subset = `NUMBER OF GUARDS` %in% t_f), aes(x = `DEFENSIVE RATING`, fill = factor(`NUMBER OF GUARDS`)))
```

```
## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_bin()`).
```



## DEFENSIVE RATING by Lineup Type (baldwin wallace)



```
tapply(game$`DEFENSIVE RATING`[game$`NUMBER OF GUARDS` %in% t_f], game$`NUMBER OF GUARDS`[game$`NUMBER OF GUARDS` %in% t_f],
```

```
## $`3`
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.00   0.00   25.00   31.94   56.94   77.78
##
```

```
## $`4`
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   NA's
##   0.0    0.0   133.3   110.4   200.0   300.0     1
```

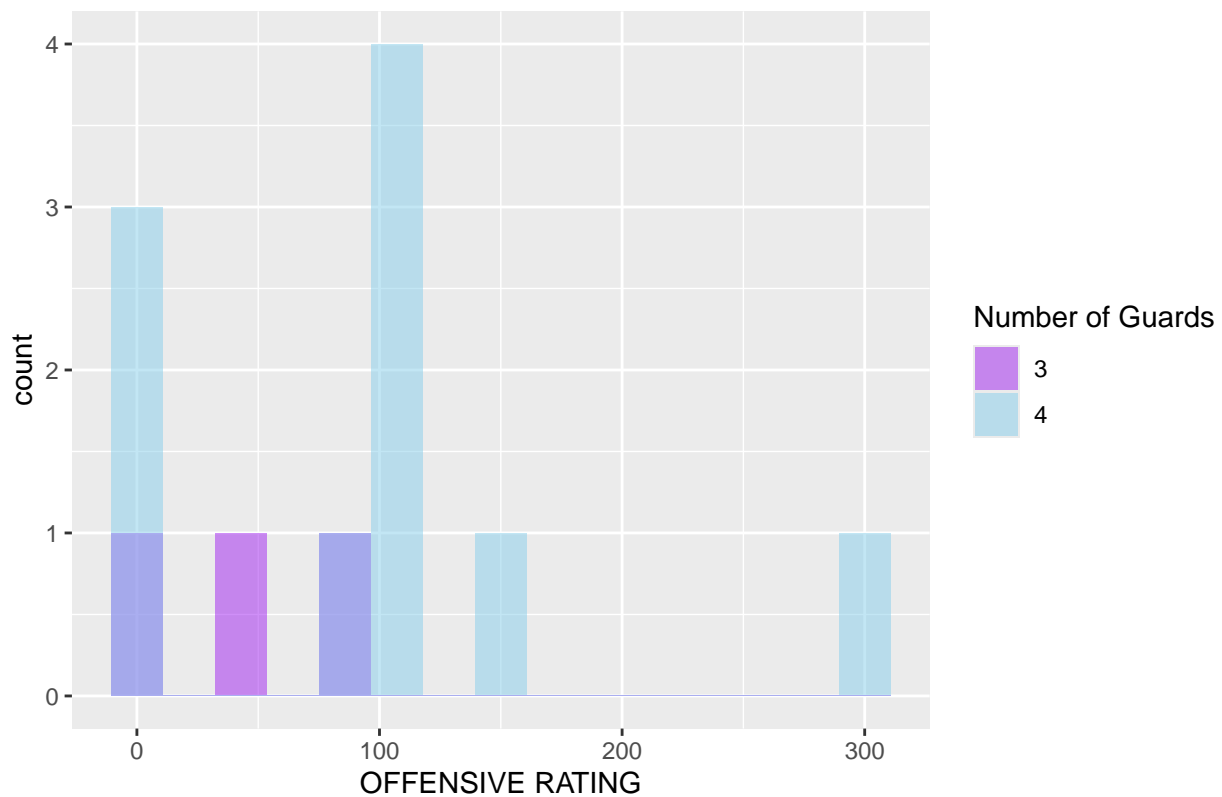
```
wilcox.test(`DEFENSIVE RATING` ~ `NUMBER OF GUARDS`, data = subset(game, `NUMBER OF GUARDS` %in% t_f),
```

```
##
## Wilcoxon rank sum test with continuity correction
##
## data: DEFENSIVE RATING by NUMBER OF GUARDS
## W = 12, p-value = 0.3713
## alternative hypothesis: true location shift is not equal to 0
```

```
ggplot(data = subset(game, subset = `NUMBER OF GUARDS` %in% t_f), aes(x = `OFFENSIVE RATING`, fill = fa
```

```
## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_bin()`).
```

## OFFENSIVE RATING by Lineup Type (baldwin wallace)



```

tapply(game$`OFFENSIVE RATING`[game$`NUMBER OF GUARDS` %in% t_f], game$`NUMBER OF GUARDS`[game$`NUMBER OF GUARDS` %in% t_f],

```

```

## $`3`
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   NA's
##   0.00  23.53   47.06   42.35  63.53   80.00     1
##

```

```

## $`4`
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.00  20.83  100.00   93.33  100.00  300.00

```

```

wilcox.test(`OFFENSIVE RATING` ~ `NUMBER OF GUARDS`, data = subset(game, `NUMBER OF GUARDS` %in% t_f),

```

```

##
## Wilcoxon rank sum test with continuity correction
##
## data:  OFFENSIVE RATING by NUMBER OF GUARDS
## W = 7.5, p-value = 0.2236
## alternative hypothesis: true location shift is not equal to 0

```

```

#dev.off()

```