

# oberlin 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: 26 Columns: 22
## -- Column specification
## -----
## (1): LINEUP (NAMES) dbl (20): ...1, NUMBER OF GUARDS, OPPONENT POSSESSIONS, CMU POSSESSIONS, OPPONENT
## DIFFERENTIAL WHEN ENTE... time (1): LINEUP MINUTES
## i Use `spec()` to retrieve the full column specification for this data. i Specify the column types o
## 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%
```

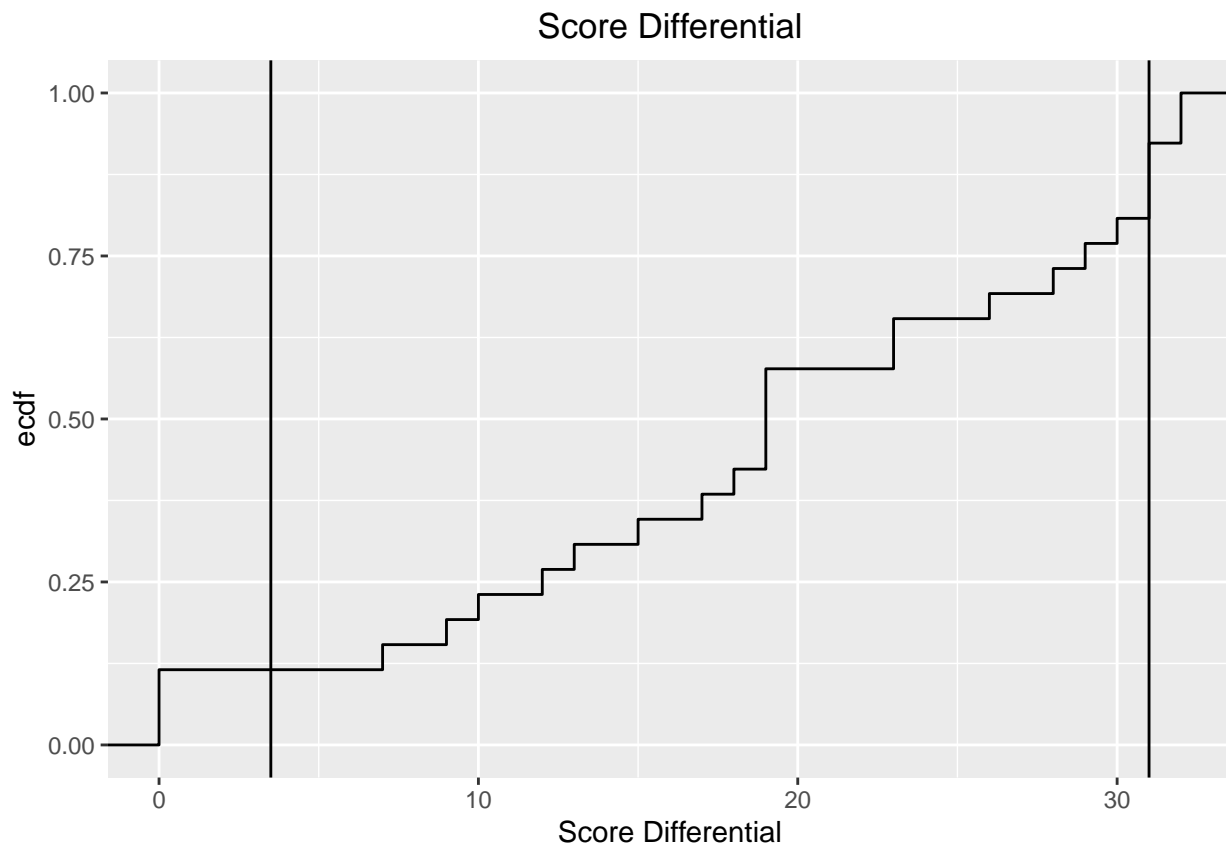
```
## 3.5
```

```
u
```

```
## 90%
```

```
## 31
```

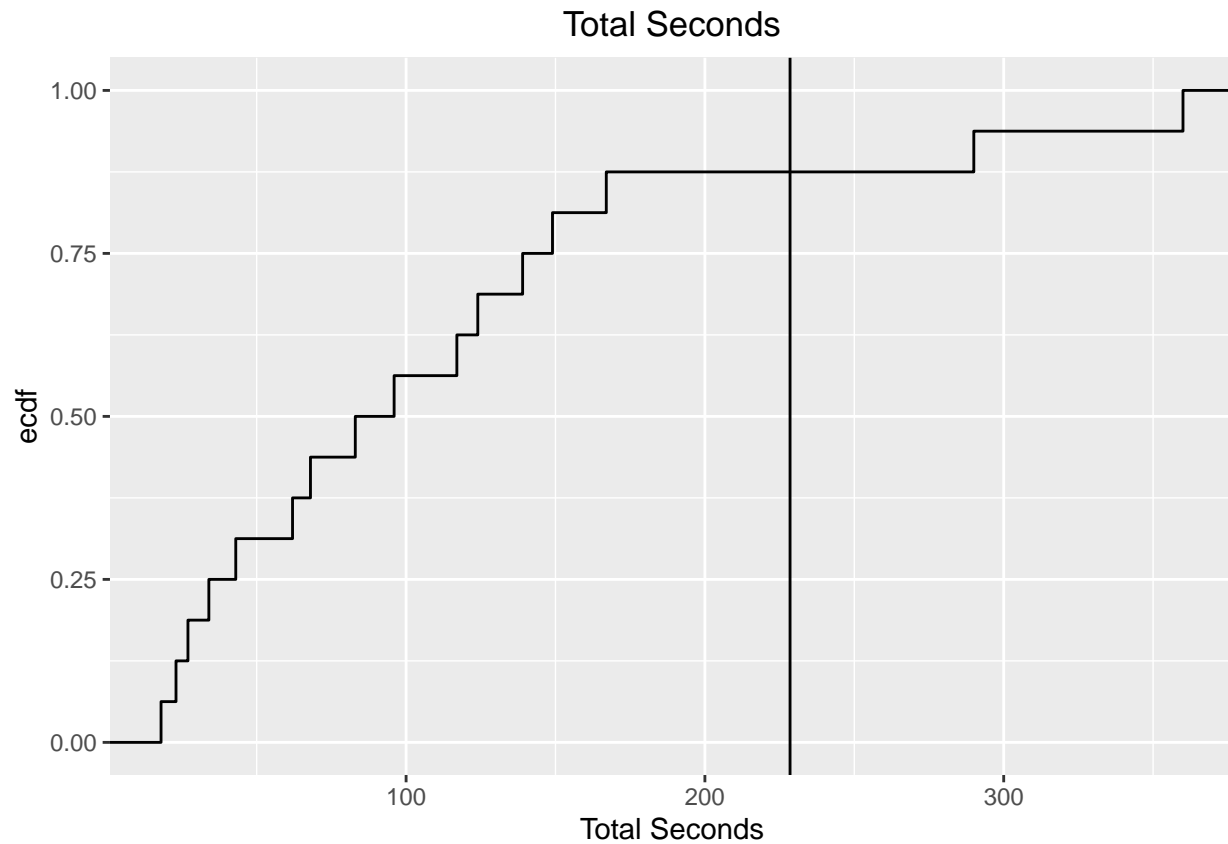
```
ggplot(singular_game, aes(x = `SCORE DIFFERENTIAL WHEN ENTER`)) + stat_ecdf() + geom_vline(xintercept =
```



```
game <- subset(game, !((`SCORE DIFFERENTIAL WHEN ENTER` <= l | `SCORE DIFFERENTIAL WHEN ENTER` >= u) &
```

```
# 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%
```

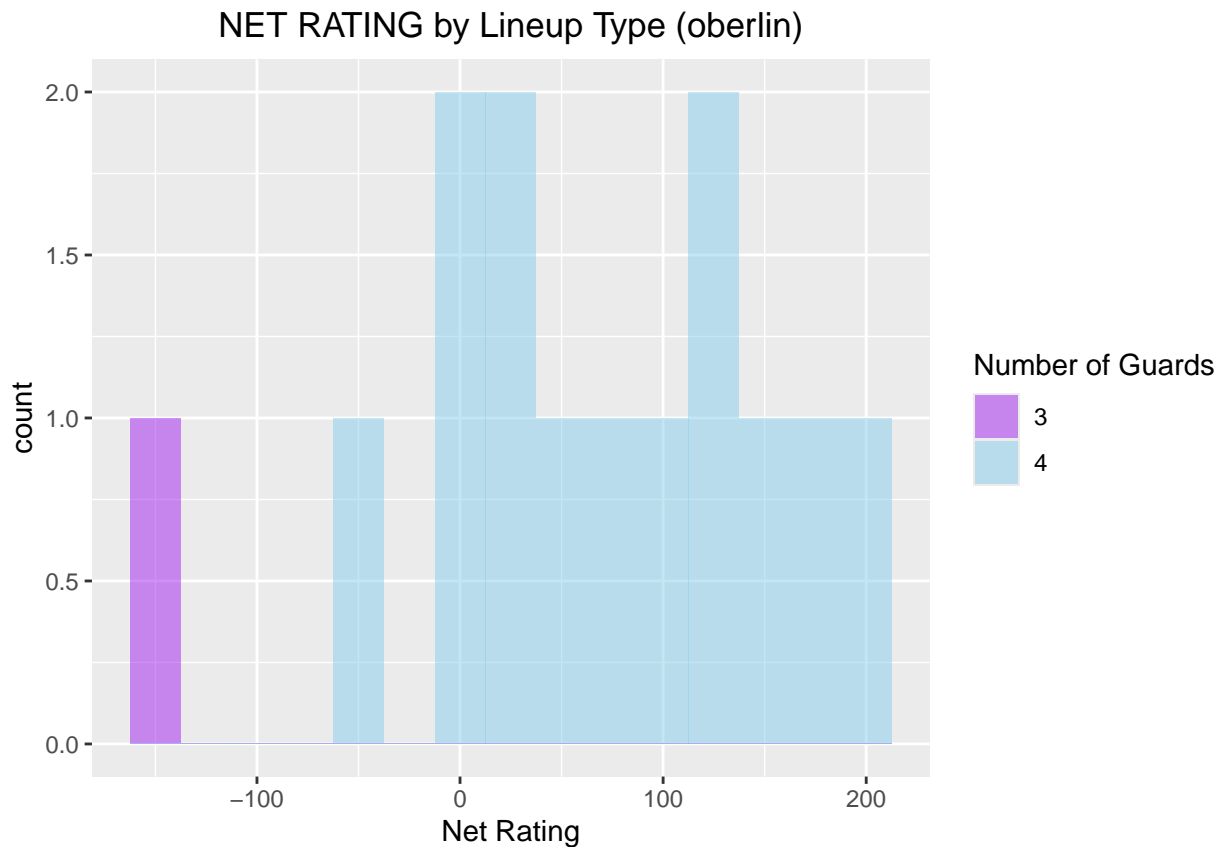
```
## 228.5
```

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

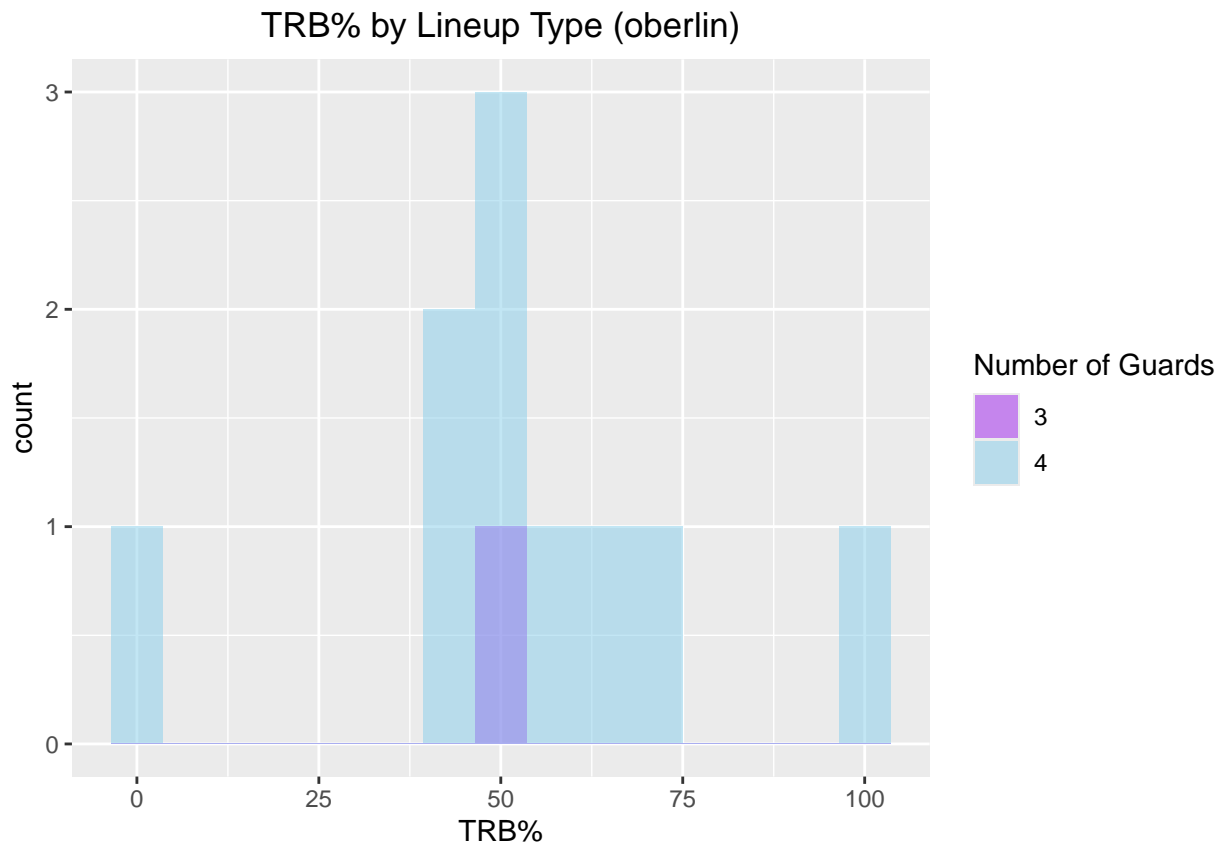
```
t_f <- c("3", "4")
```

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

```
## Warning: Removed 1 row 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.
  ##   -150   -150   -150   -150   -150   -150
  ##
  ## $`4`
  ##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   NA's
  ##   -60.00  25.00  83.33  76.03 126.67 200.00      1
  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 = 0, p-value = 0.1362
  ## 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`)))
  ## Warning: Removed 4 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])
```

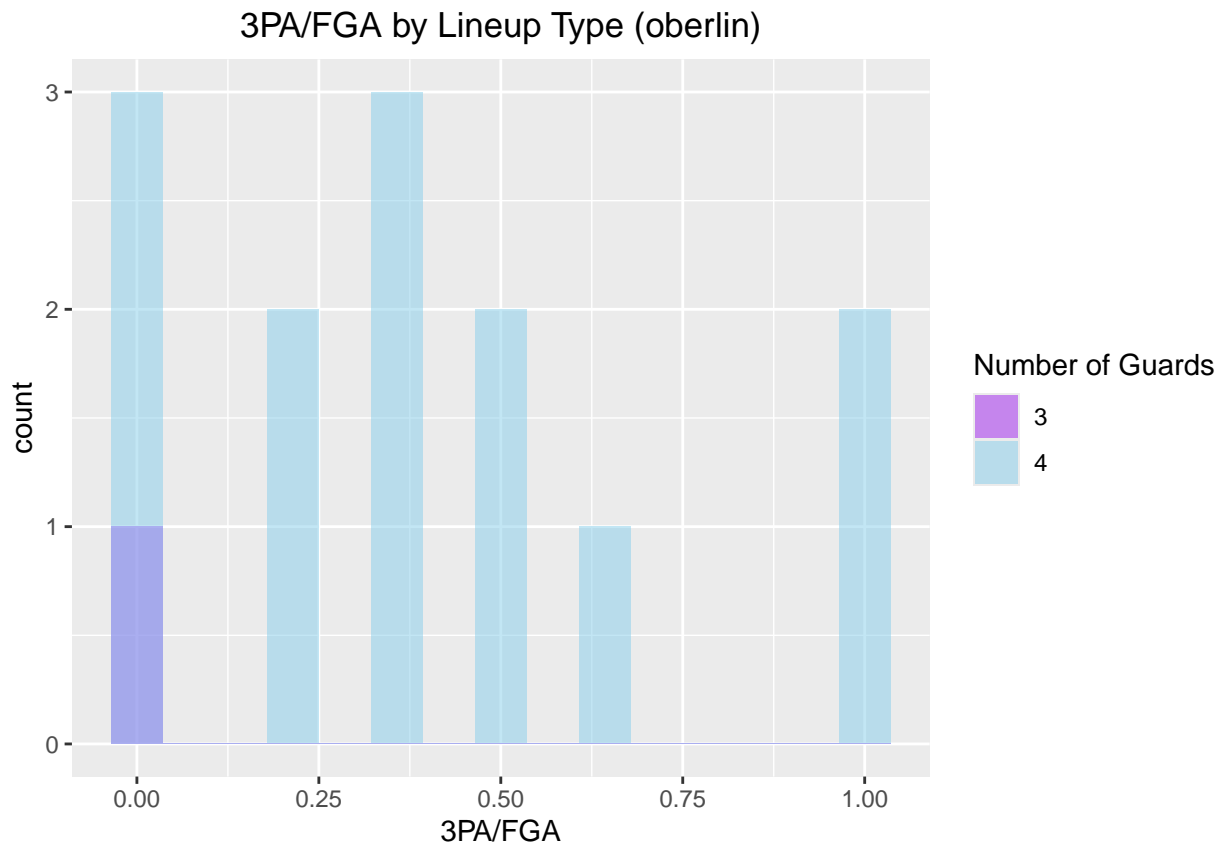
```
## $`3`
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       50      50      50      50      50      50
##
## $`4`
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   NA's
##       0.00  45.83   50.00   53.61  65.00  100.00      4
```

```
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 = 4.5, p-value = 1
## 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`
```

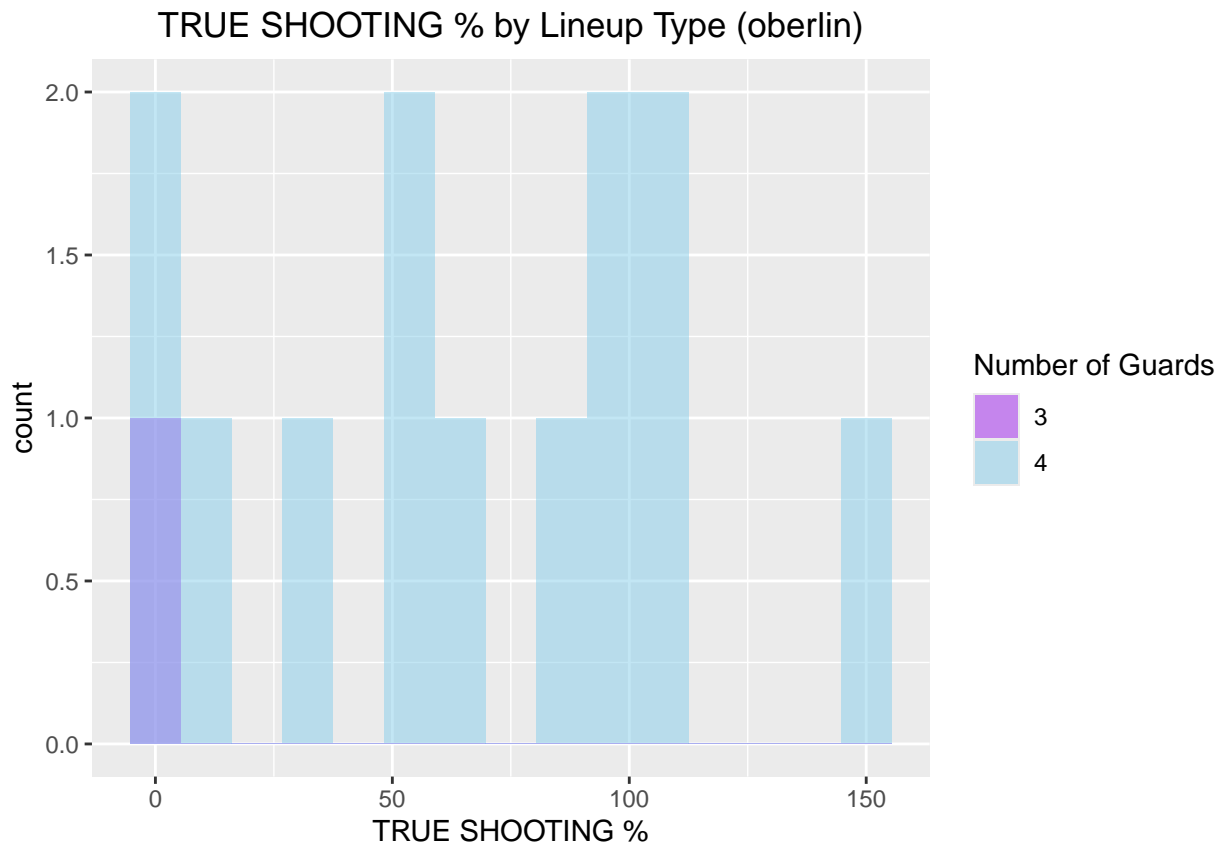
```
## $`3`
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0       0       0       0       0       0
##
## $`4`
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   NA's
## 0.0000 0.2000 0.3333 0.3874 0.5000 1.0000      1
```

```
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 = 1.5, p-value = 0.2552
## 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()`).
```



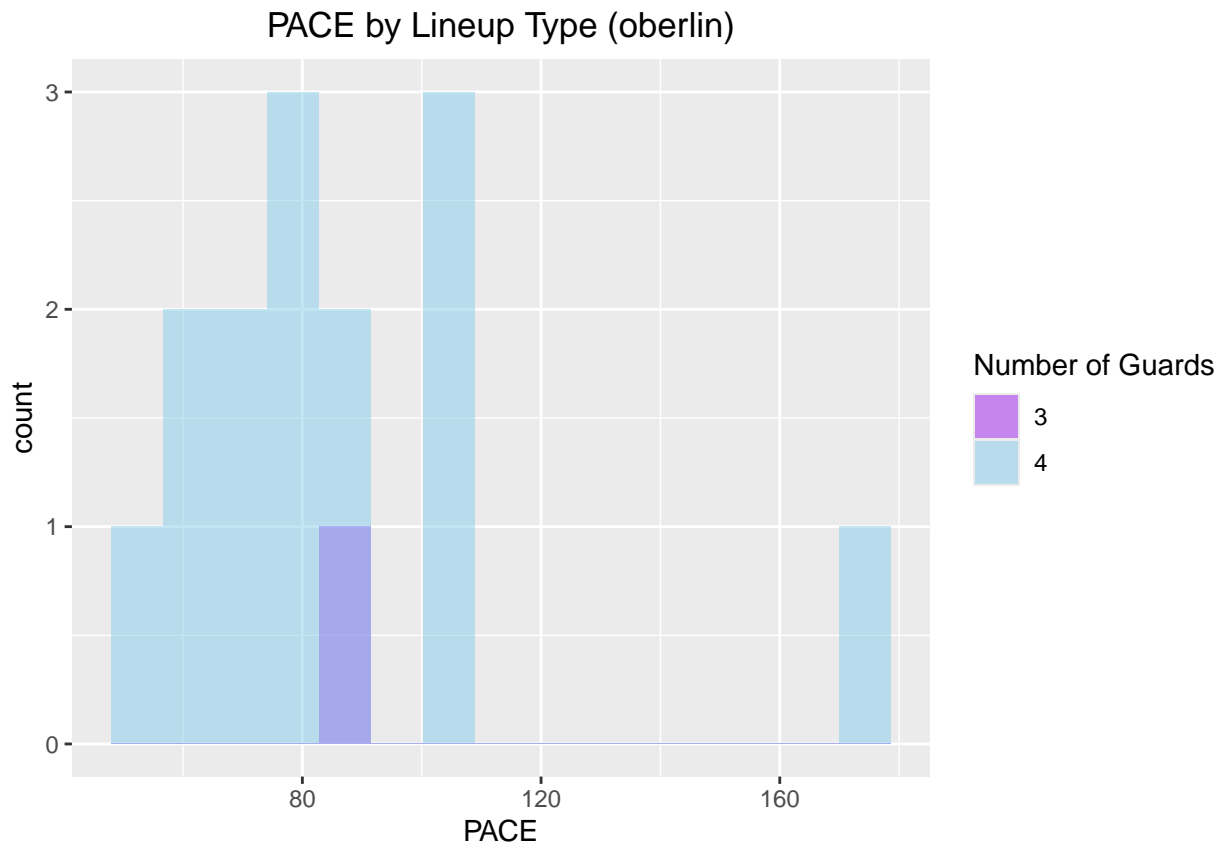
```
tapply(game$TRUE SHOOTING % [game$NUMBER OF GUARDS %in% t_f], game$NUMBER OF GUARDS [game$NUMBER OF GUARDS %in% t_f], FUN = function(x) {
  summary(x)
})
```

```
## $`3`
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##         0         0         0         0         0         0
##
## $`4`
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   NA's
##      0.00  34.01   67.34   67.20  100.00  150.00      1
```

```
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 = 1, p-value = 0.2123
## 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)))
```

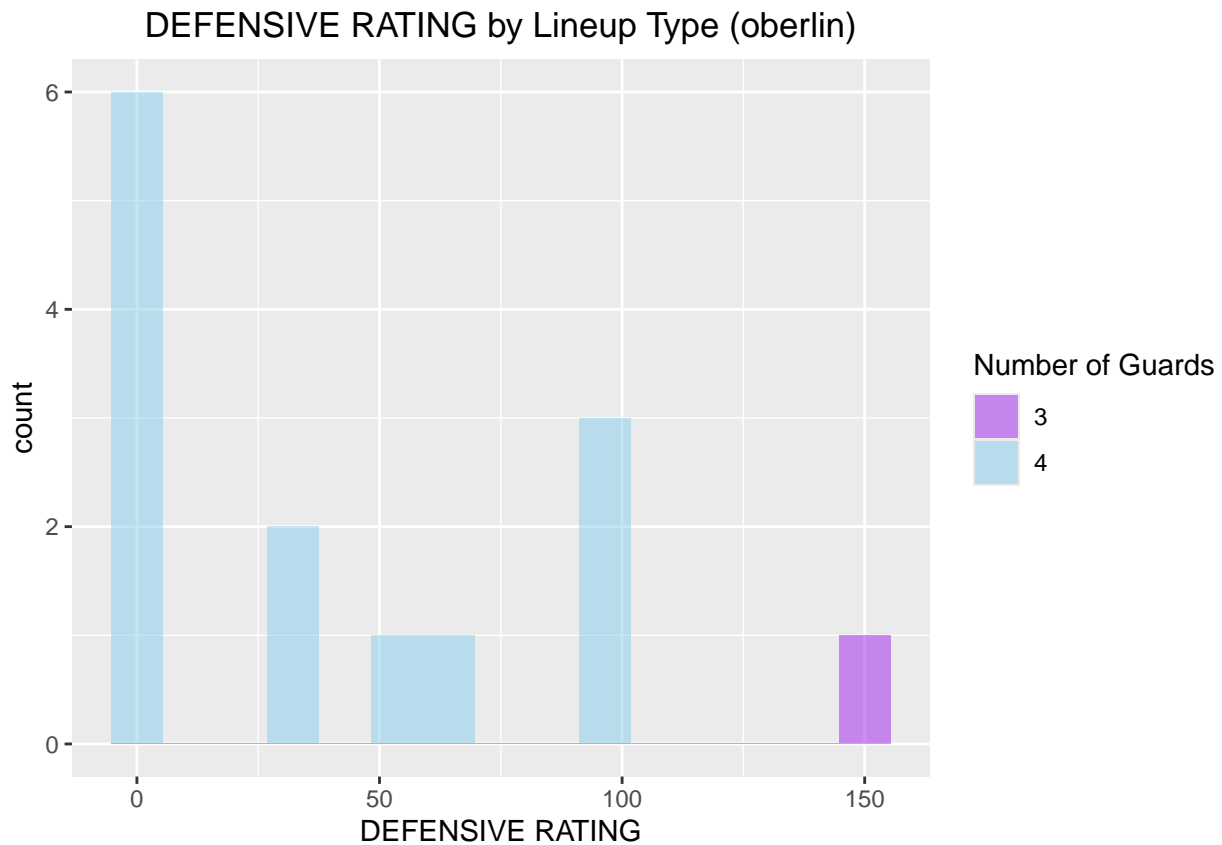


```

tapply(game$`PACE`[game$`NUMBER OF GUARDS` %in% t_f], game$`NUMBER OF GUARDS`[game$`NUMBER OF GUARDS` %in% t_f], FUN = function(x) {
  ## $`3`
  ##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  ##   87.5   87.5   87.5   87.5   87.5   87.5
  ##
  ## $`4`
  ##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  ##   55.81  67.95   78.16   86.78   98.05  177.78
  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 = 9, p-value = 0.7282
  ## 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()`).

```





```

tapply(game$`DEFENSIVE RATING`[game$`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.
##   150    150    150    150    150    150
##

```

```

## $`4`
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   NA's
##   0.00   0.00   33.33   37.18   66.67  100.00     1

```

```

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

```

```

##
## Wilcoxon rank sum test with continuity correction
##
## data: DEFENSIVE RATING by NUMBER OF GUARDS
## W = 13, p-value = 0.1191
## alternative hypothesis: true location shift is not equal to 0

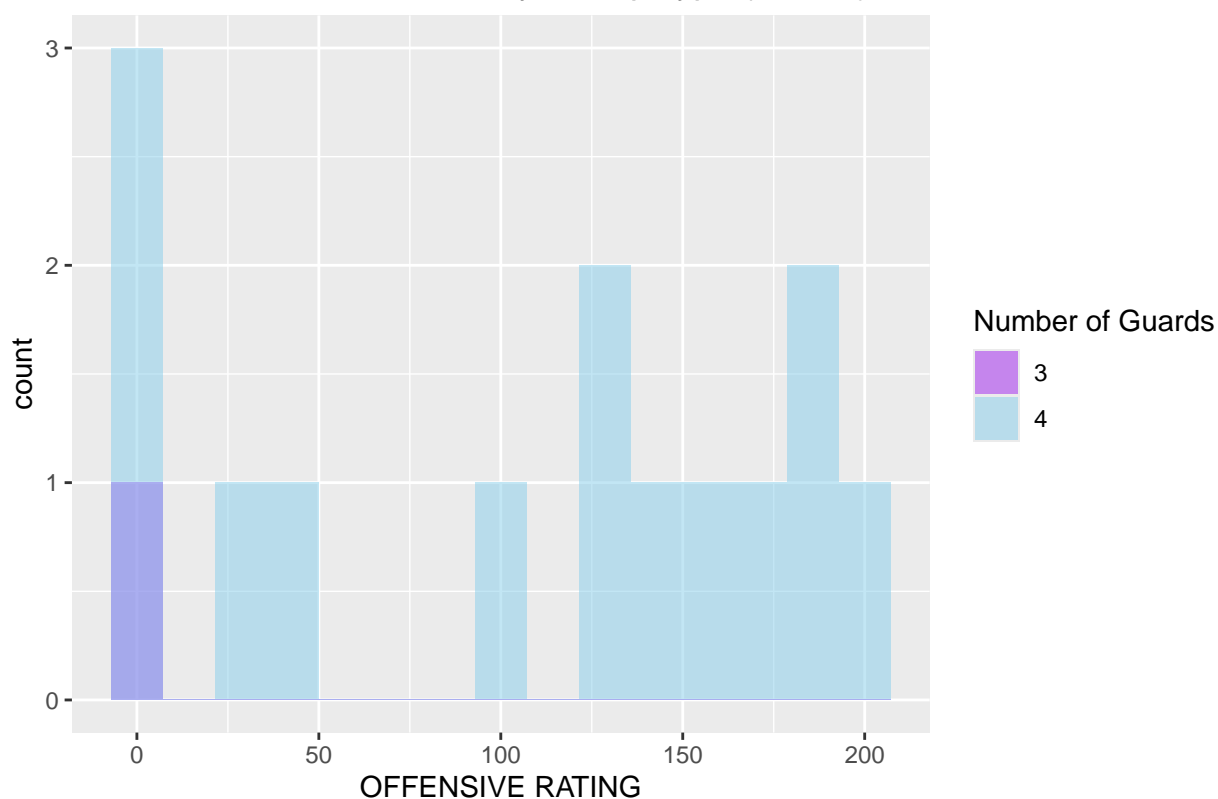
```

```

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

```

## OFFENSIVGE RATING by Lineup Type (oberlin)



```
tapply(game$`OFFENSIVE RATING`[game$`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.
##         0         0         0         0         0         0
##
## $`4`
##      Min. 1st Qu.  Median      Mean 3rd Qu.     Max.
##         0.00   28.75  133.33  105.12  165.00   200.00
```

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

```
##
## Wilcoxon rank sum test with continuity correction
##
## data:  OFFENSIVE RATING by NUMBER OF GUARDS
## W = 1.5, p-value = 0.2425
## alternative hypothesis: true location shift is not equal to 0
```

```
#dev.off()
```