

case western reserve2 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
## -----
## (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%
```

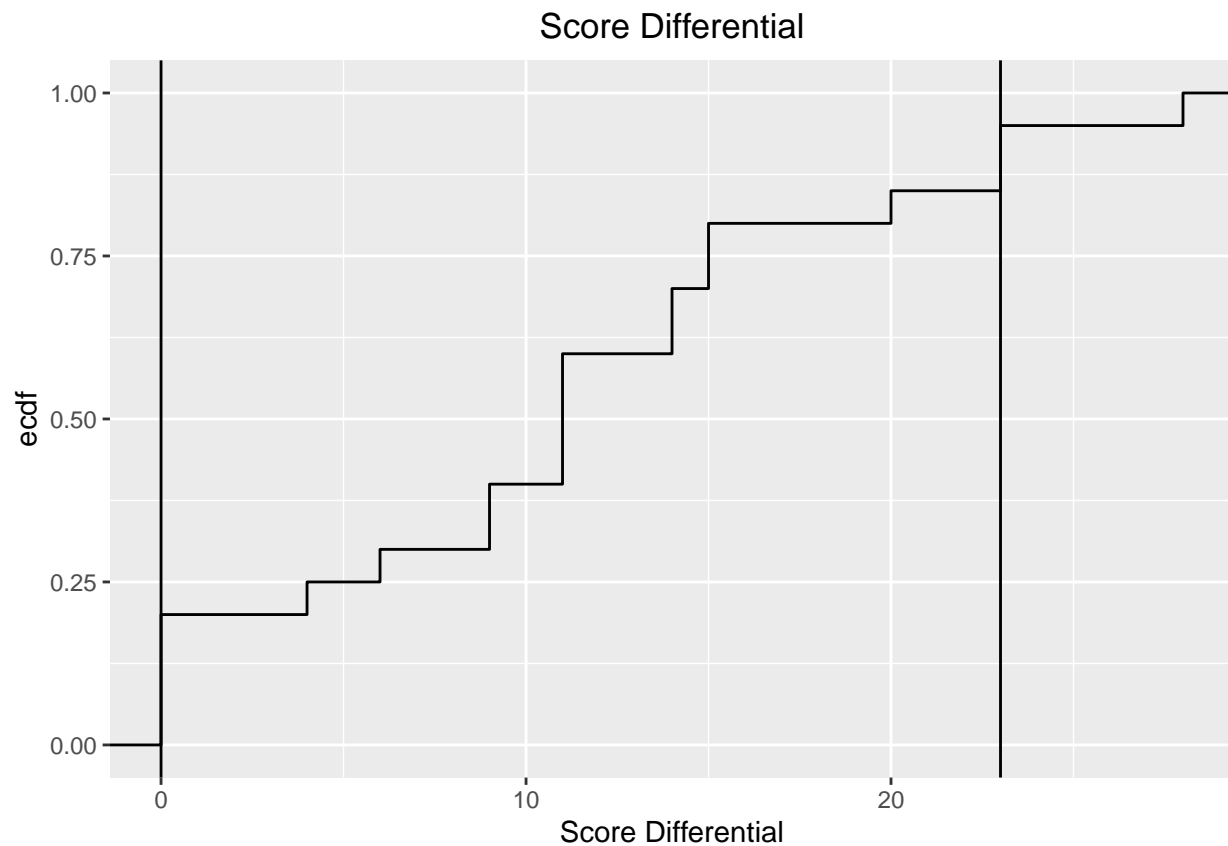
```
## 0
```

```
u
```

```
## 90%
```

```
## 23
```

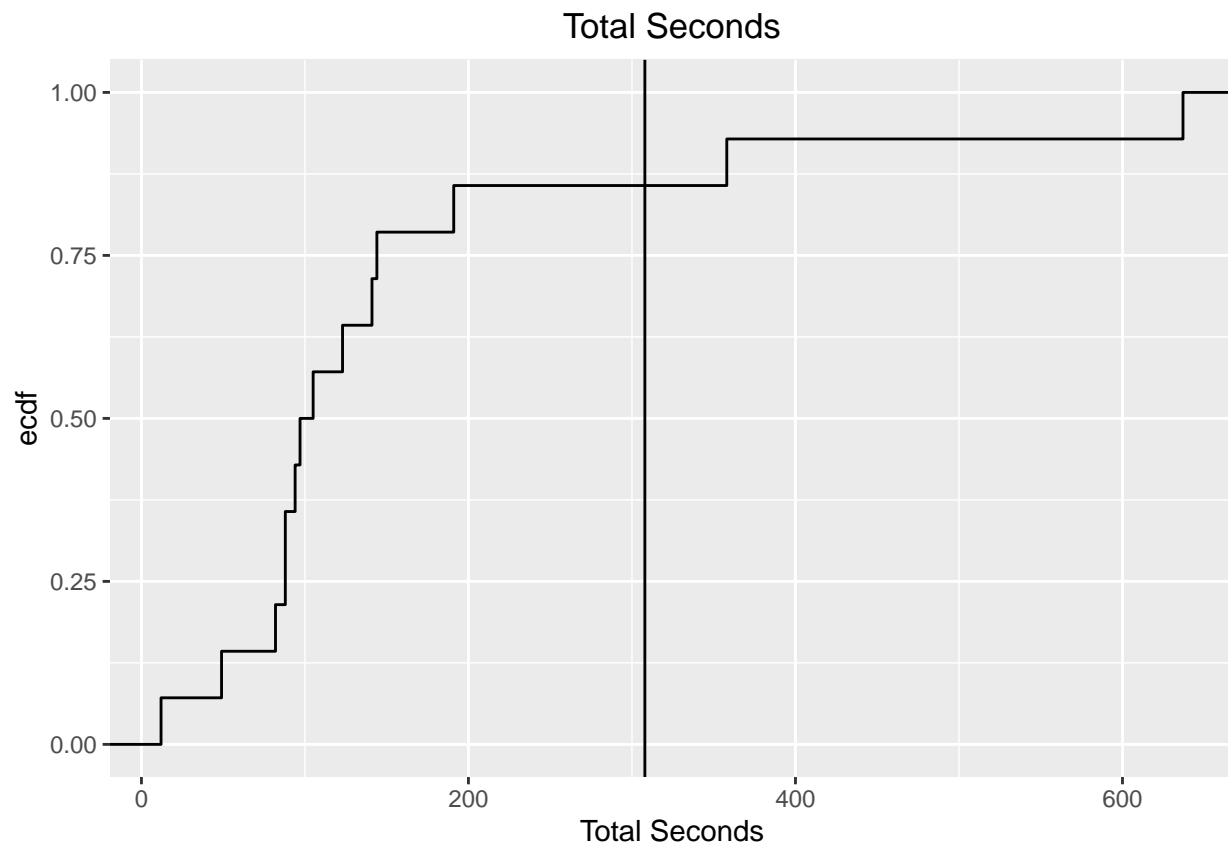
```
ggplot(singular_game, aes(x = `SCORE DIFFERENTIAL WHEN ENTER`)) + stat_ecdf() + geom_vline(xintercept =
```



```
game <- subset(game, !((`SCORE DIFFERENTIAL WHEN ENTER` <= 1 | `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
```

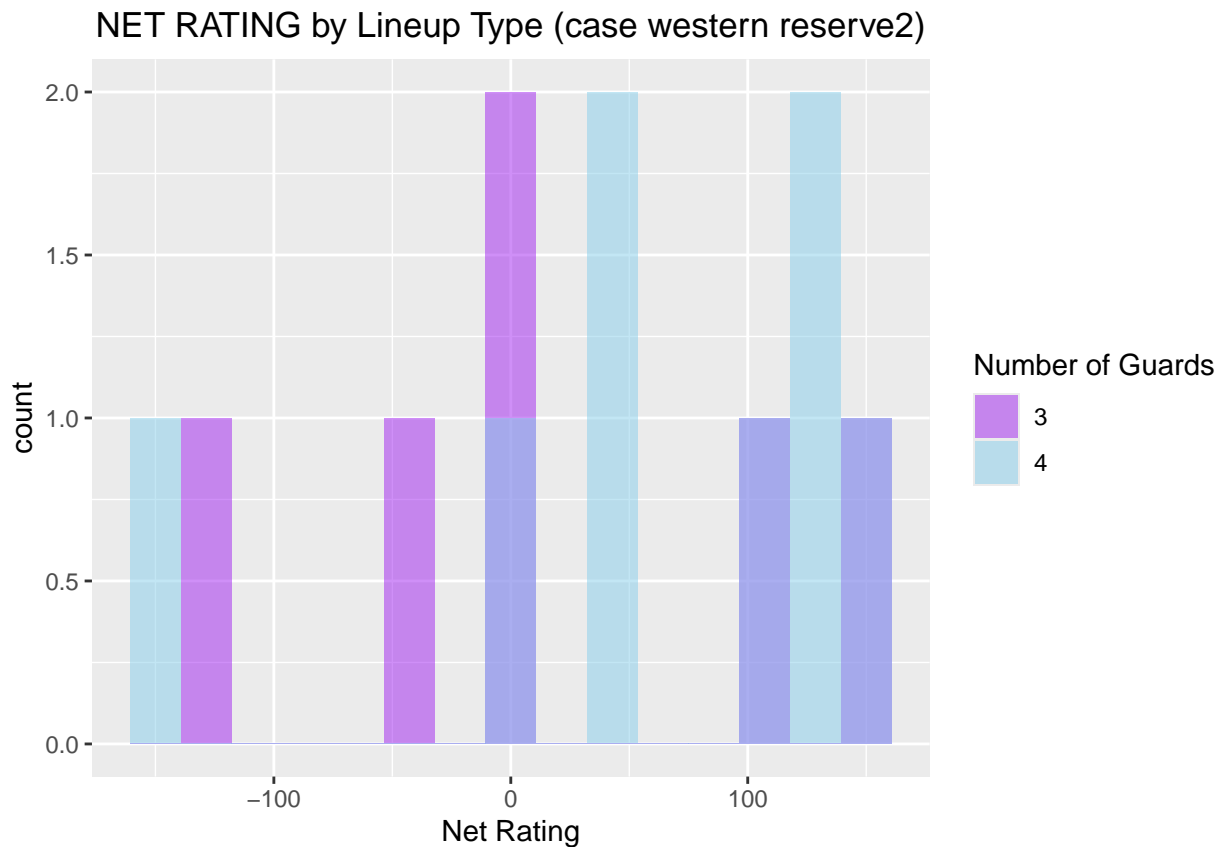
```
## 90%
```

```
## 307.9
```

```
#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(`
```

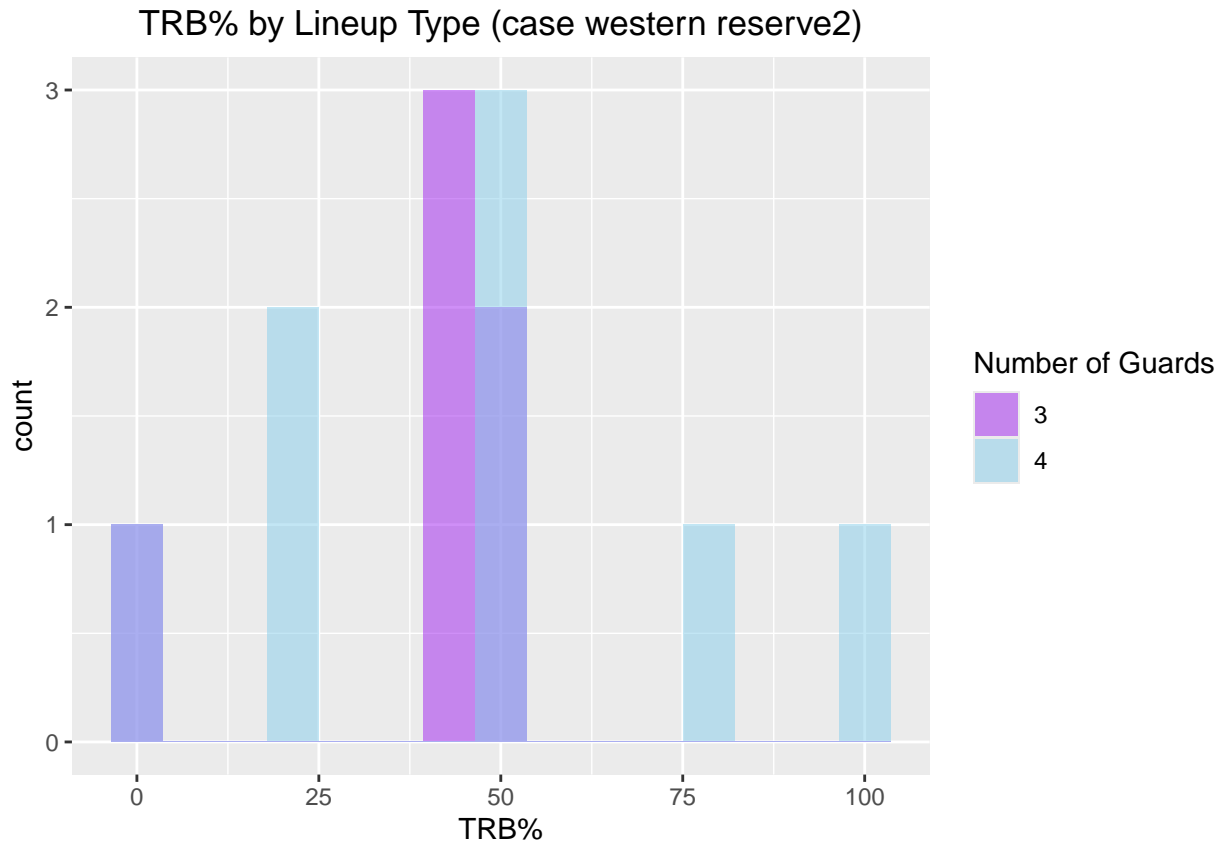


```
tapply(game$`NET RATING`[game$`NUMBER OF GUARDS` %in% t_f], game$`NUMBER OF GUARDS`[game$`NUMBER OF GUARDS` %in% t_f],
  FUN = function(x) {
    ## $`3`
    ##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    ## -133.33 -31.47    0.00   12.45   75.00   150.00
    ##
    ## $`4`
    ##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    ## -150.00  30.00   75.00   55.98  126.74   150.00
  },
  exact = TRUE)

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

##
## Wilcoxon rank sum test with continuity correction
##
## data: NET RATING by NUMBER OF GUARDS
## W = 17.5, p-value = 0.436
## 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`)))
```



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

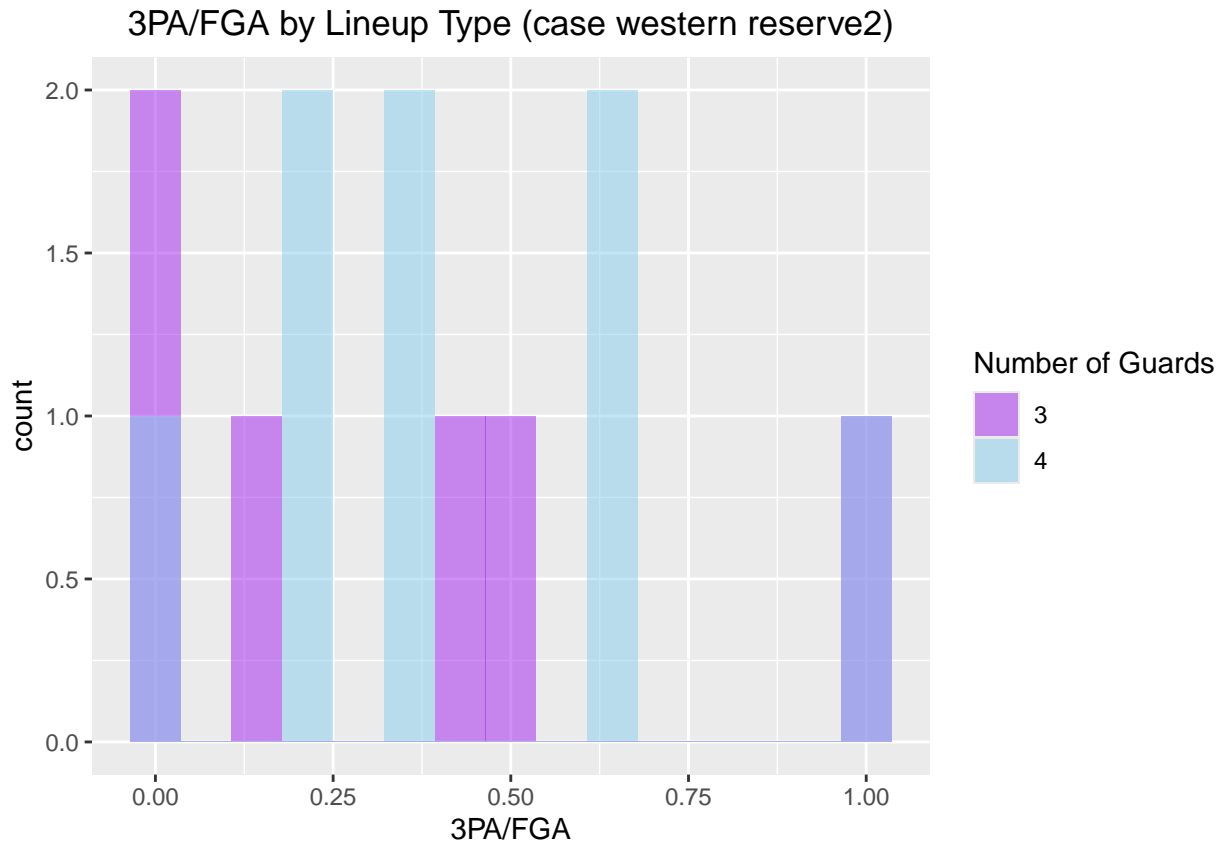
```
## $`3`
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.00  40.00   42.58   37.53  48.79   50.00
```

```
##
## $`4`
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.00  23.75   50.00   46.60  56.94   100.00
```

```
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 = 18.5, p-value = 0.5082
## 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`)))
```



```
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.
## 0.00000 0.04167 0.30833 0.35278 0.48750 1.00000
```

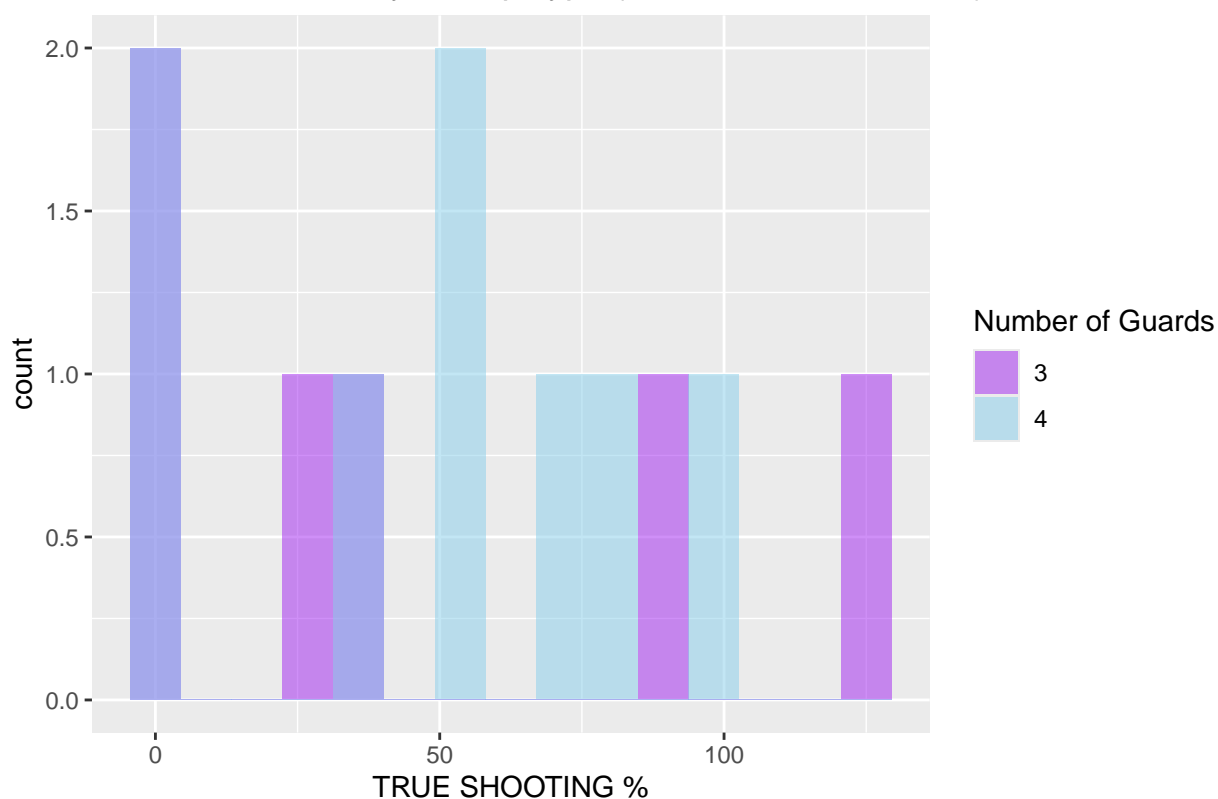
```
##
## $`4`
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00000 0.25000 0.33333 0.43750 0.66667 1.00000
```

```
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 = 19.5, p-value = 0.6024
## 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
```

TRUE SHOOTING % by Lineup Type (case western reserve2)



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

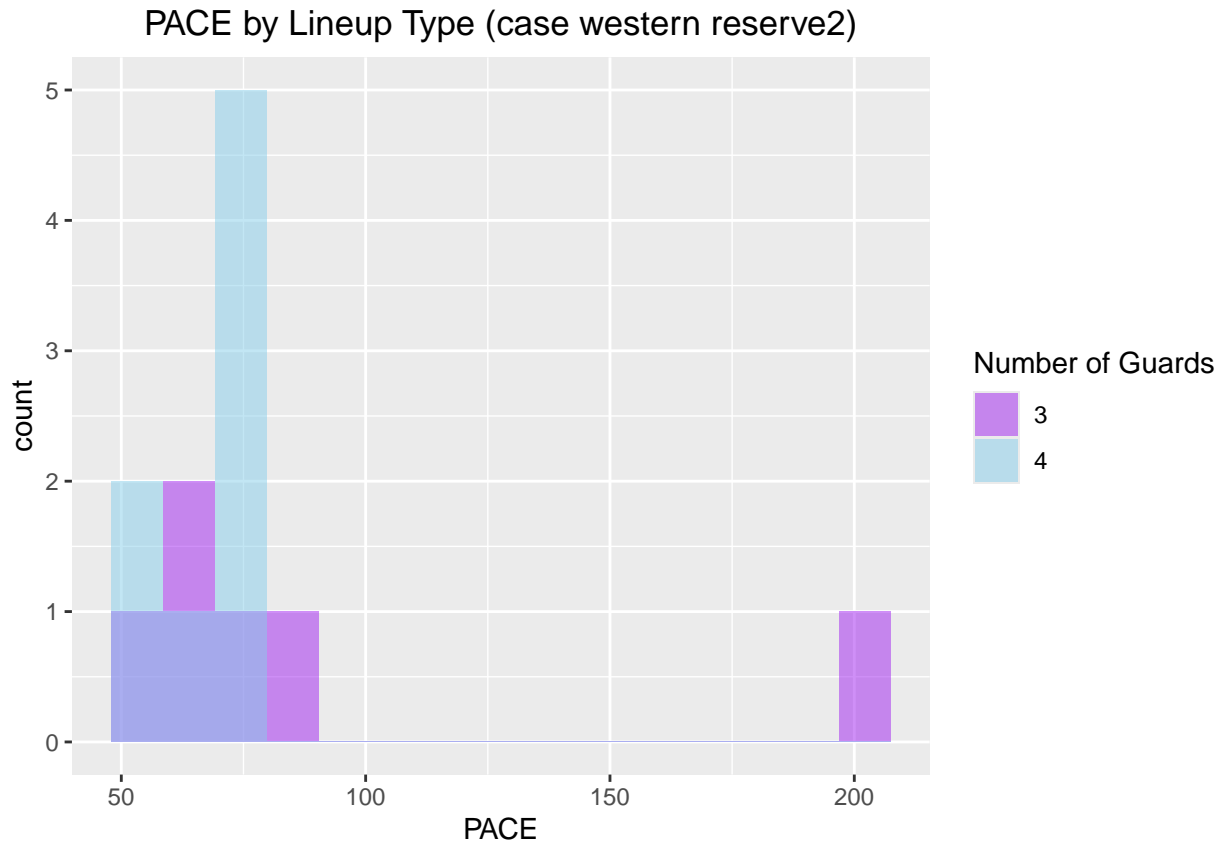
```
## $`3`
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.000   6.319   29.305   45.879   77.083  125.000
##
```

```
## $`4`
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.00   25.00   50.00   48.84   76.84  100.00
```

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

```
##
## Wilcoxon rank sum test with continuity correction
##
## data: TRUE SHOOTING % by NUMBER OF GUARDS
## W = 21.5, p-value = 0.7936
## 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` %in% t_f)))
```



```

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

```

```

## $`3`
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    57.14  58.97   67.84   88.86  80.21   200.00
##

```

```

## $`4`
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    51.06  64.70   71.78   67.49  73.66   75.00

```

```

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 = 30, p-value = 0.4777
## alternative hypothesis: true location shift is not equal to 0

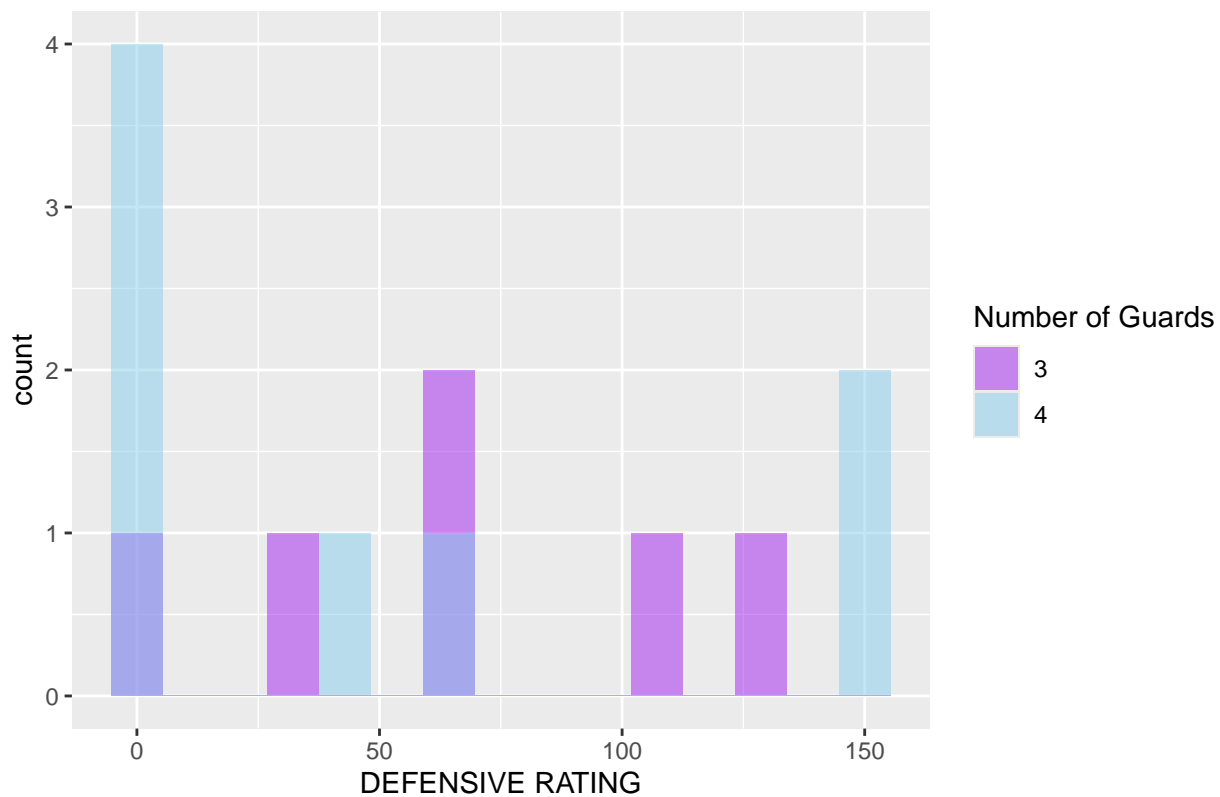
```

```

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

```


DEFENSIVE RATING by Lineup Type (case western reserve2)



```
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  41.67   66.67   67.78  96.67  133.33
##
```

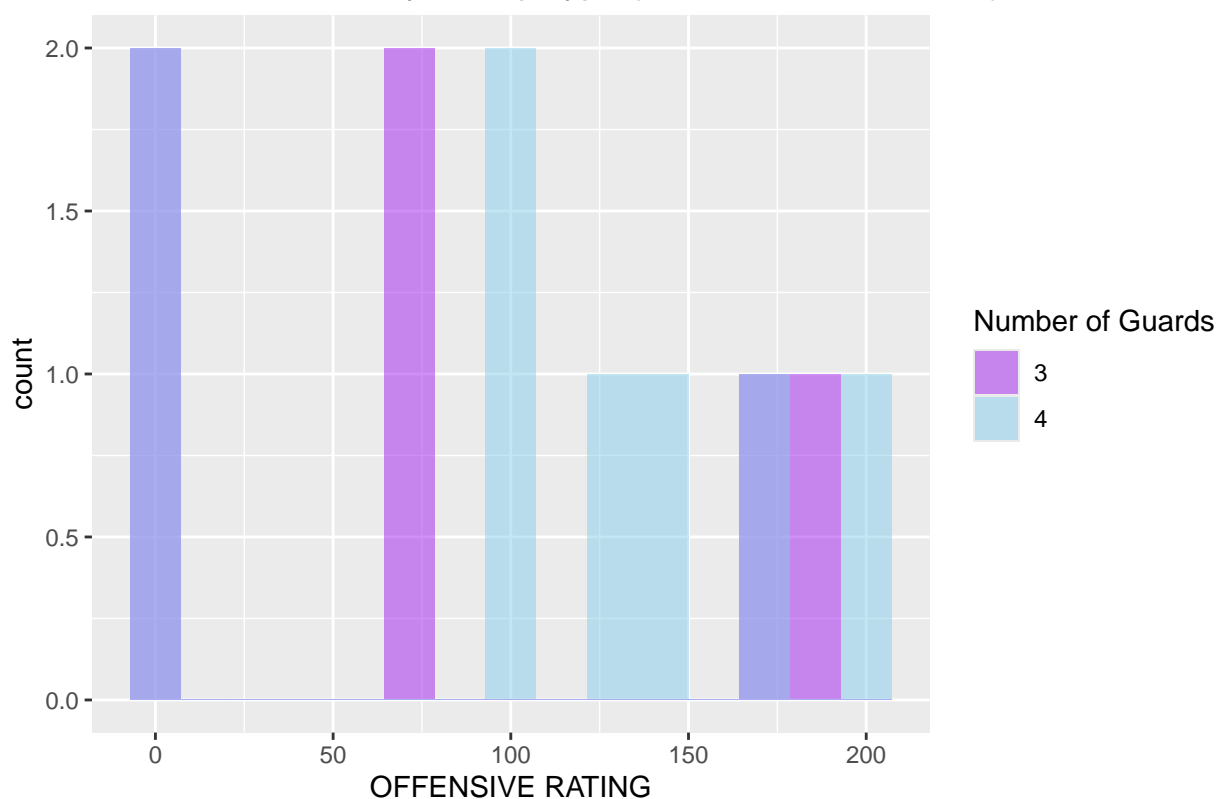
```
## $`4`
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.00   0.00   22.73   50.68  82.50  150.00
```

```
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 = 30, p-value = 0.4667
## 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 (case western reserve2)



```
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.
##      0.00  16.18   65.69   80.23 141.67  183.33
##
```

```
## $`4`
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      0.0   75.0   116.7   106.7  155.0   200.0
```

```
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 = 19, p-value = 0.5565
## alternative hypothesis: true location shift is not equal to 0
```

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