# la roche EDA

2025-07-02

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

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

```
## New names:
## Rows: 21 Columns: 22
## -- Column specification
## ------------------------------------------------------------------------------------ Delimiter: "," cl
## (1): LINEUP (NAMES) dbl (20): ...1, NUMBER OF GUARDS, OPPONENT POSSESSIONS, CMU POSSESSIONS, OPPONENT
## CMU PTS, SCORE ... time (1): LINEUP MINUTES
## i Use `spec()` to retrieve the full column specification for this data. i Specify the column types o
## `show_col_types = FALSE` to quiet this message.
## * `` -> `...1`
```

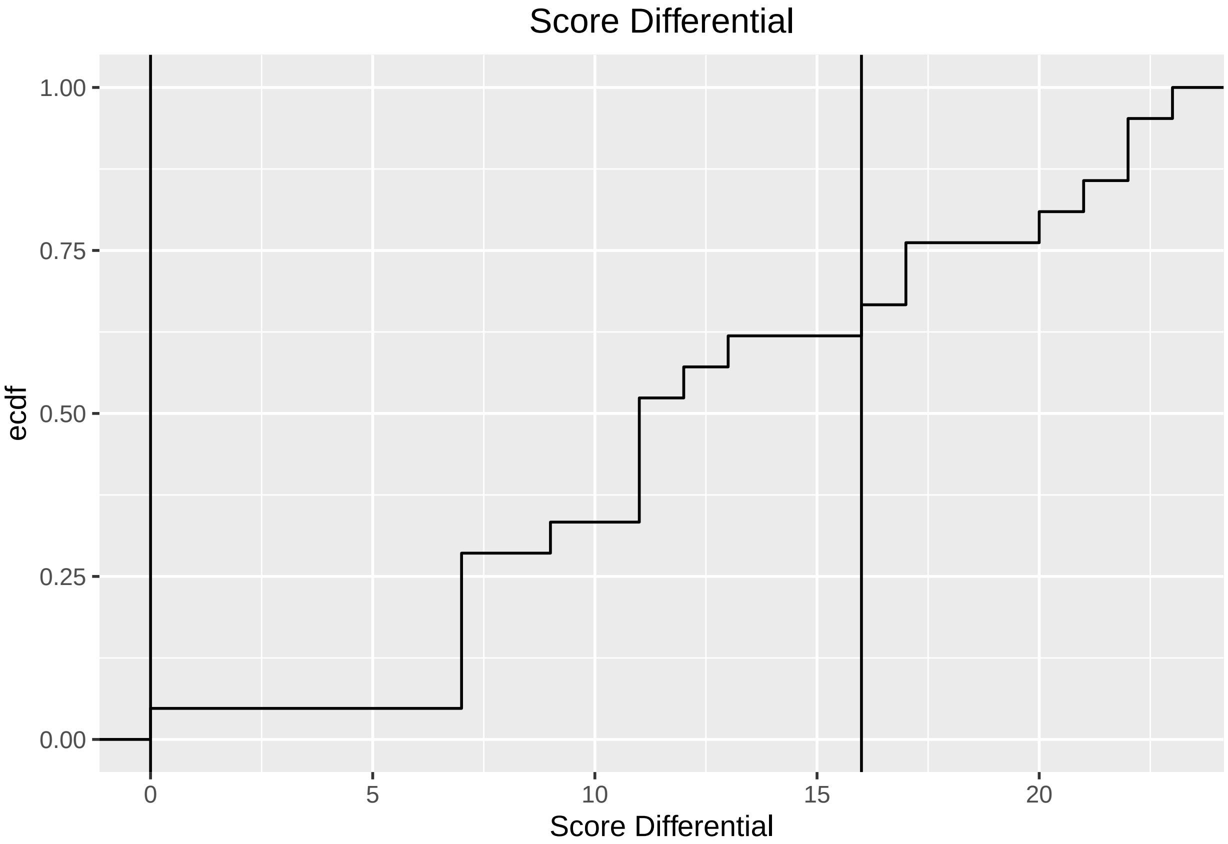
```r
# if negatives in any columns (specifically had problem in possession column)
for (colName in colnames(singular_game)){
  singular_game[[colName]][singular_game[[colName]] < 0] <- 0
}
```

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

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

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

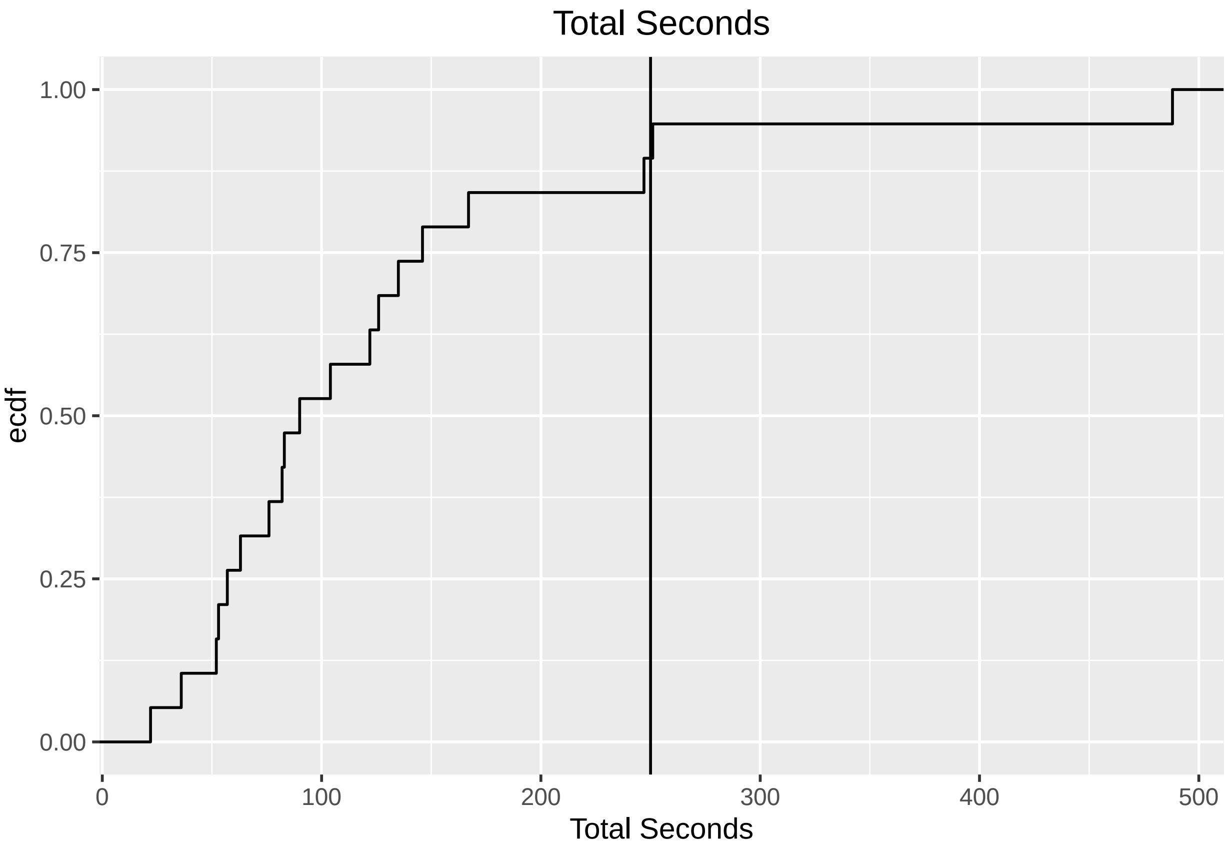


```
quantile(singular_game$`SCORE DIFFERENTIAL WHEN ENTER`,probs=c(0.1,0.9))
```

```
## 10% 90%
##   7  22
```

```
#game <- subset(game, !((`SCORE DIFFERENTIAL WHEN ENTER` <= 0 | `SCORE DIFFERENTIAL WHEN ENTER` >= 16) &
```

```
# see where to cut time -> SHOULD DO THIS AFTER OR BEFORE CUT SCRAP MINUTES?
ggplot(game, aes(x = `LINEUP SECONDS`)) + stat_ecdf() + geom_vline(xintercept = 250) + labs(title = "To
```



```
quantile(game$`LINEUP SECONDS`,probs=c(0.9))
```

```
##   90%
## 247.8
```

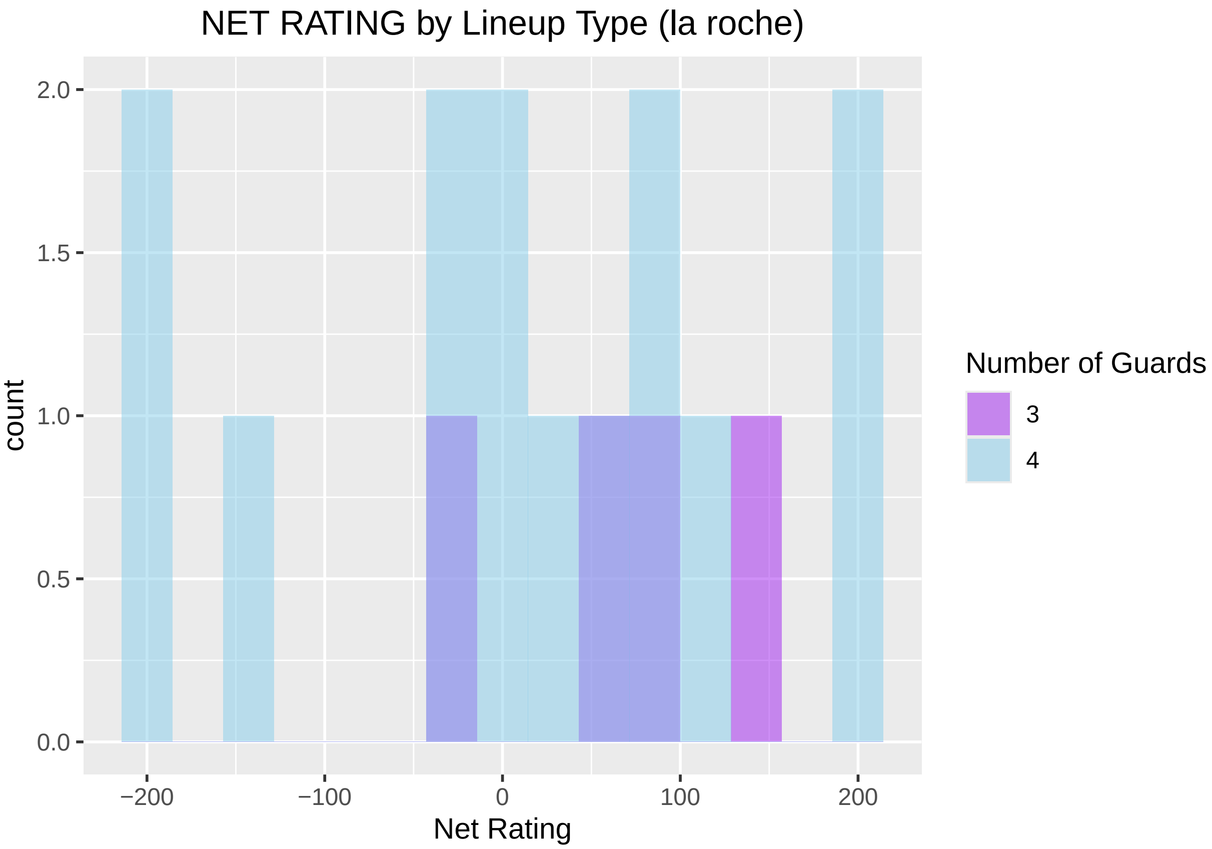
```
#game <- subset(game, `LINEUP SECONDS` >= 250)
```

```
#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(`N
```

```
## 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 GUAI
```

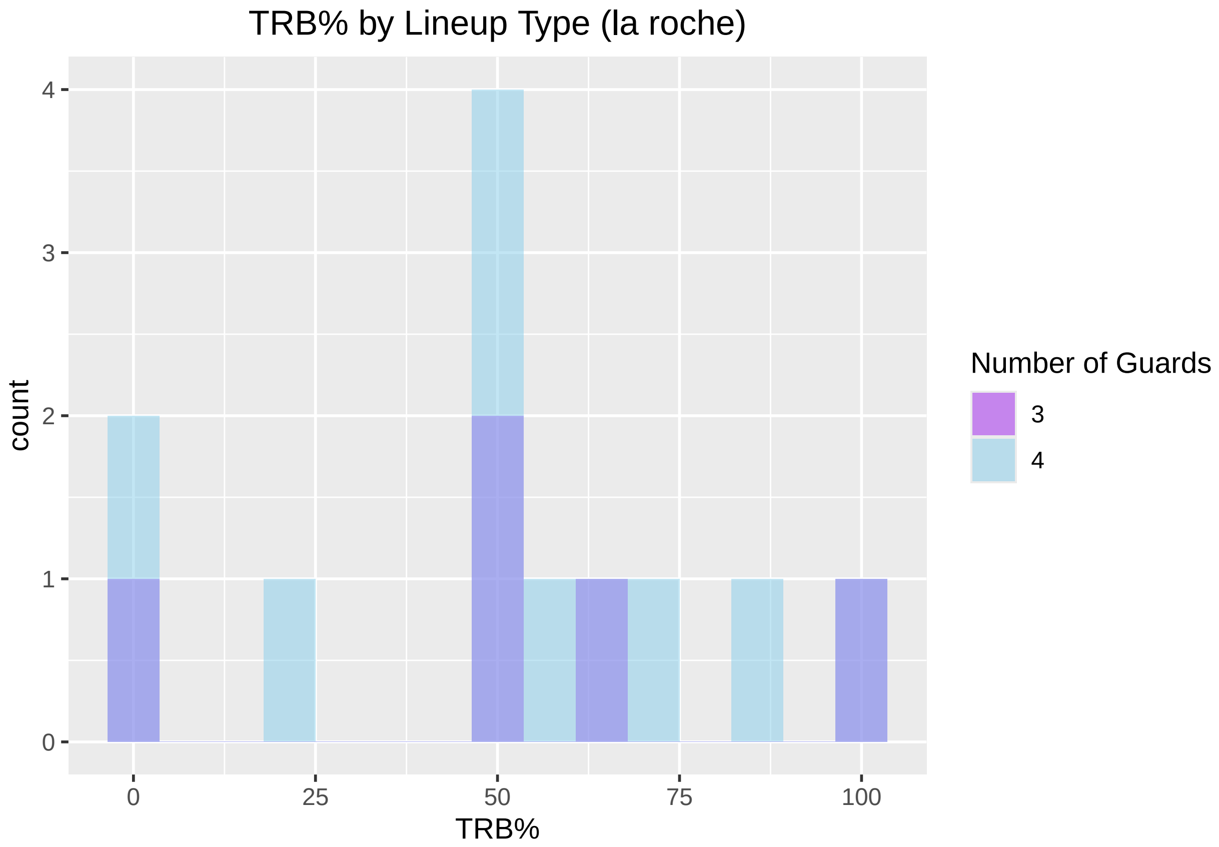
```
## $`3`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##  -33.33   40.00   82.22   70.28  112.50  150.00       1 
## 
## $`4`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
## -200.00  -30.42   16.67   15.56   98.21  200.00
```

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

```
## 
##  Wilcoxon rank sum test with continuity correction
## 
## data:  NET RATING by NUMBER OF GUARDS
## W = 35.5, p-value = 0.4563
## 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
```

```
## Warning: Removed 2 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` %i
```

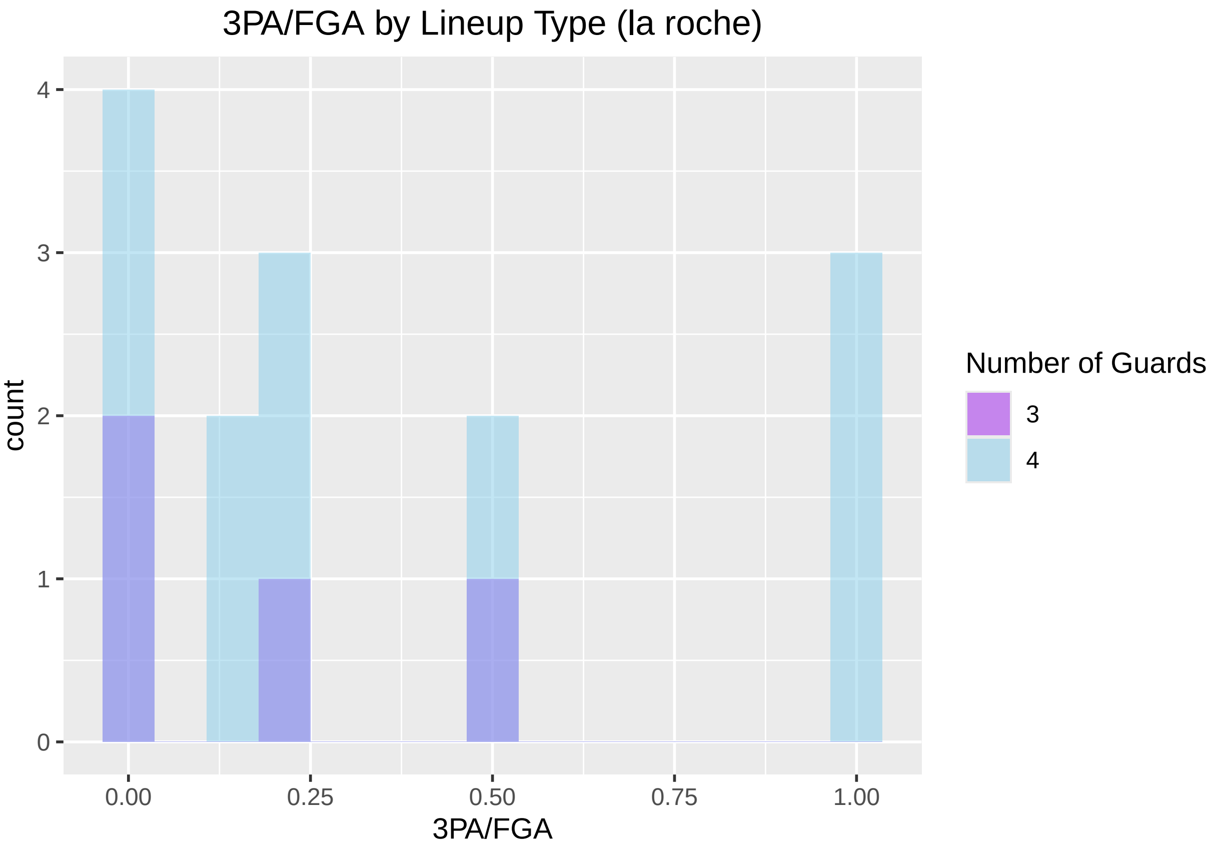
```
## $`3`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    0.00   50.00   50.00   53.33   66.67  100.00 
## 
## $`4`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##    0.00   42.50   50.00   49.96   67.50  100.00       2
```

```
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 = 31, p-value = 0.9569
## 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(`NUMB
```

```
## 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
```

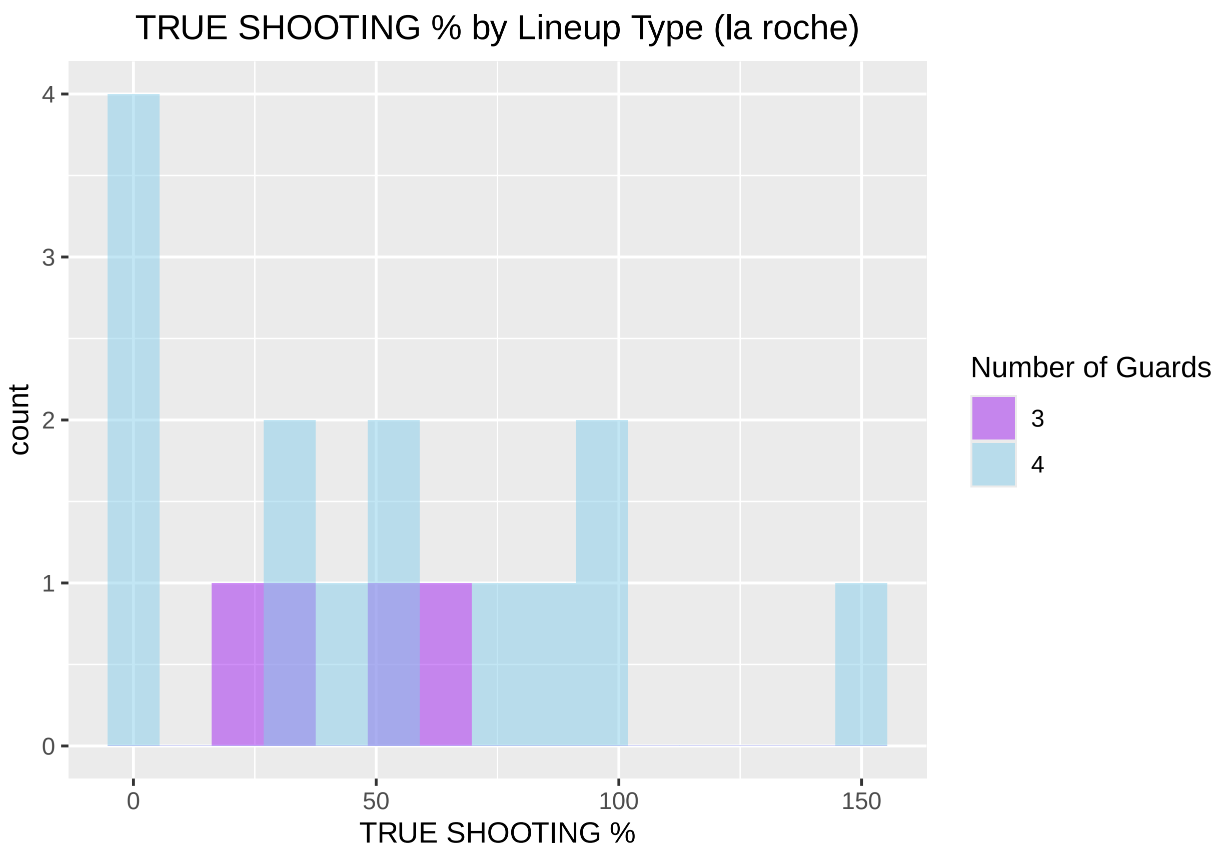
```
## $`3`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##  0.0000  0.0000  0.1250  0.1875  0.3125  0.5000       1
##
## $`4`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00000 0.03125 0.20000 0.35128 0.50000 1.00000
```

```
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 = 22.5, p-value = 0.5866
## 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()`).
```



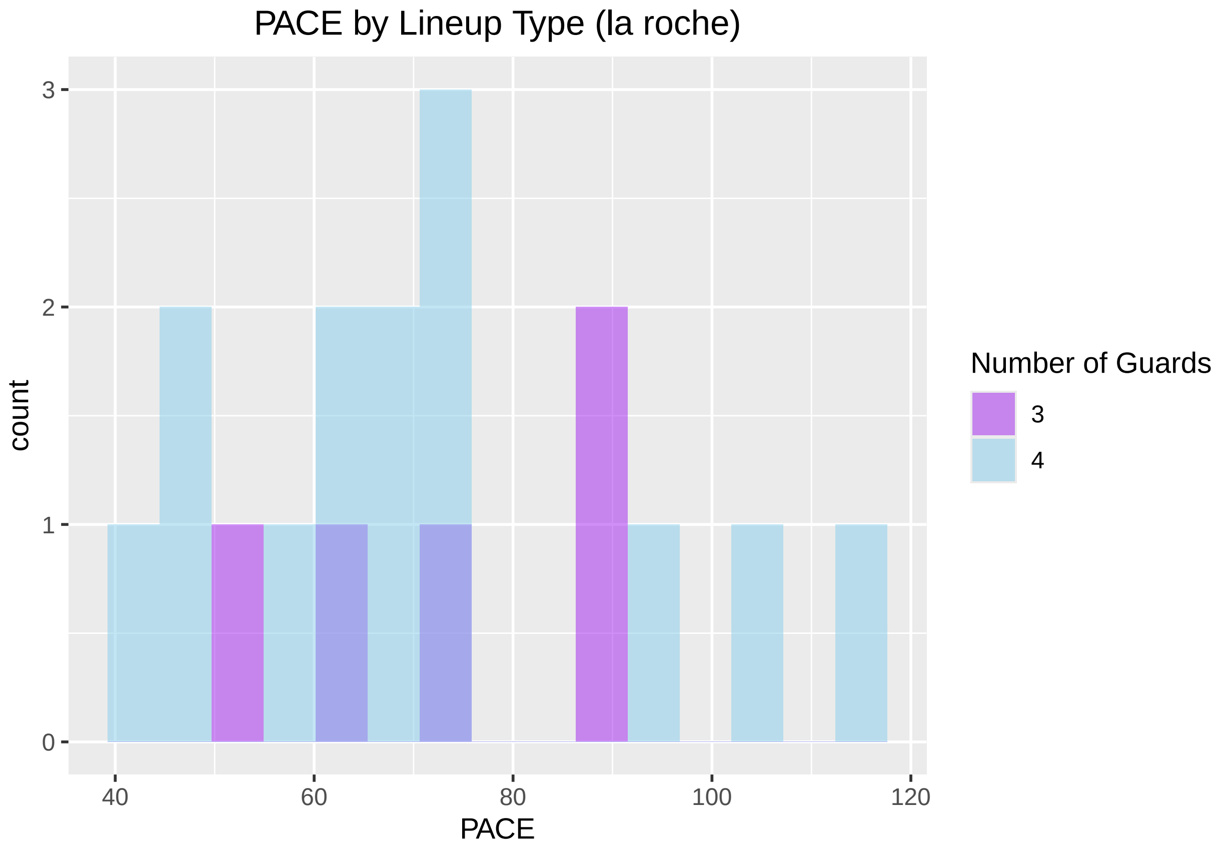
```
tapply(game$`TRUE SHOOTING %`[game$`NUMBER OF GUARDS` %in% t_f], game$`NUMBER OF GUARDS`[game$`NUMBER O
```

```
## $`3`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##   26.60   31.65   42.71   43.28   54.34   61.09       1 
## 
## $`4`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   0.000   7.143  45.000  51.140  82.155 150.000
```

```
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 = 26, p-value = 0.8727
## 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
```



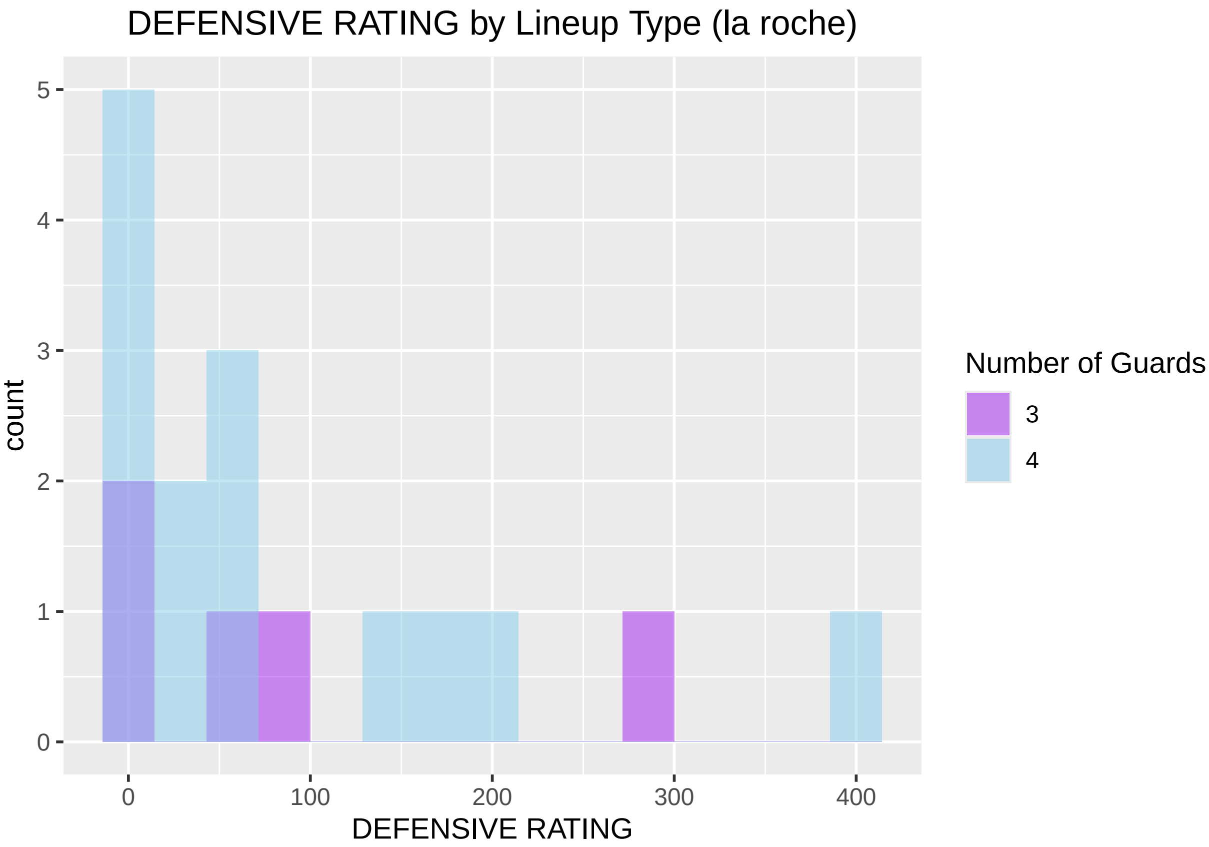
```
tapply(game$`PACE`[game$`NUMBER OF GUARDS` %in% t_f], game$`NUMBER OF GUARDS`[game$`NUMBER OF GUARDS` %
```

```
## $`3`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   54.55   63.16   72.29   73.73   87.80   90.84 
## 
## $`4`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   40.00   58.41   67.76   70.44   73.92  113.21
```

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

```
## 
##  Wilcoxon rank sum test with continuity correction
## 
## data:  PACE by NUMBER OF GUARDS
## W = 38, p-value = 0.8169
## 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
```



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

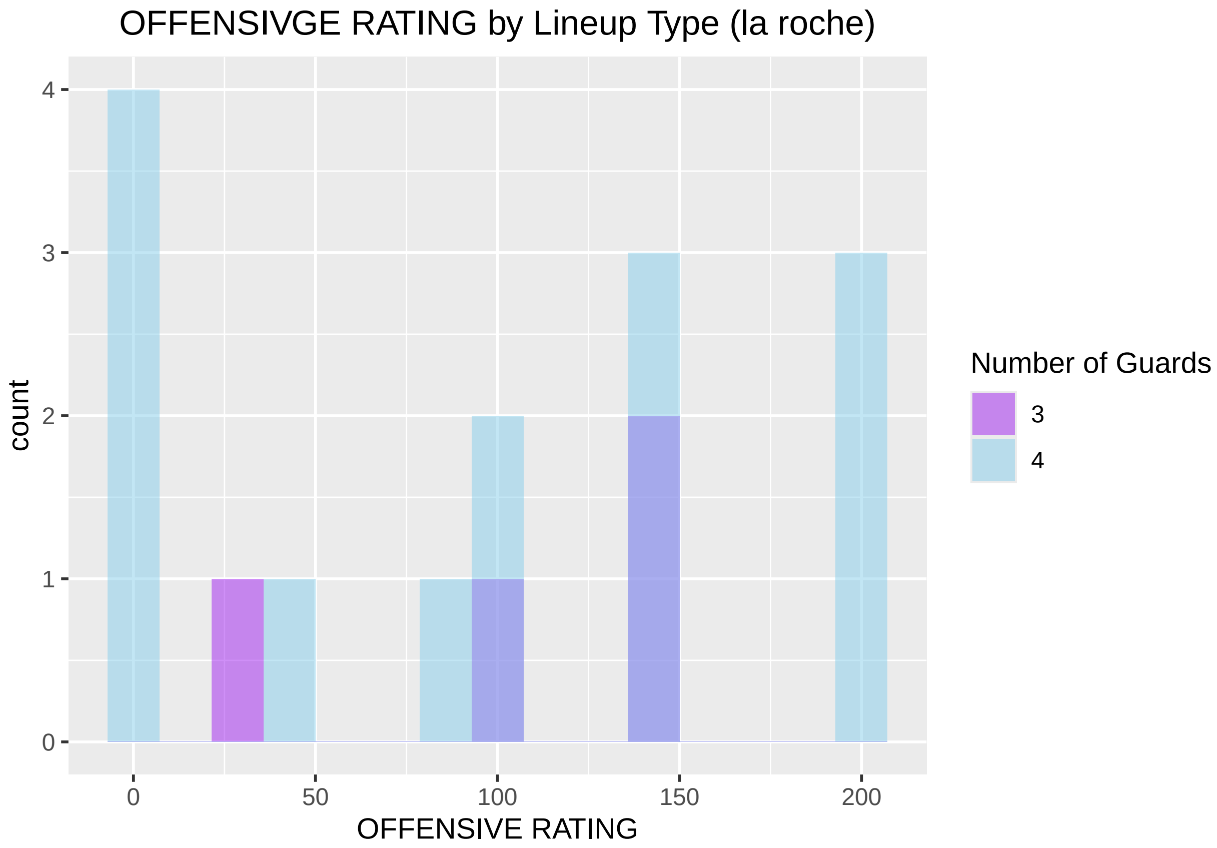
```
## $`3`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    0.00   66.67   89.33   80.00  300.00
##
## $`4`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    0.00   41.67   82.26  116.67  400.00
```

```
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 = 37, p-value = 0.8865
## 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()`).
```



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

```
## $`3`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##   33.33   83.33  122.22  106.94  145.83  150.00       1 
## 
## $`4`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    0.00   12.50  100.00   97.82  148.21  200.00
```

```
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 = 31.5, p-value = 0.7476
## alternative hypothesis: true location shift is not equal to 0
```

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