johns hopkins 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: 19 Columns: 22
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (1): LINEUP (NAMES) dbl (20): ...1, NUMBER OF GUARDS, OPPONENT POSSESSIONS,
## CMU POSSESSIONS, ... 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(c("CMU POSSESSIONS", "OPPONENT POSSESSIONS"))){
  singular_game[[colName]][singular_game[[colName]] < 0] <- 0
}
```

```
#individual_games <- readr::read_csv("Desktop/SURA project code/data frames/shortened.csv")

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

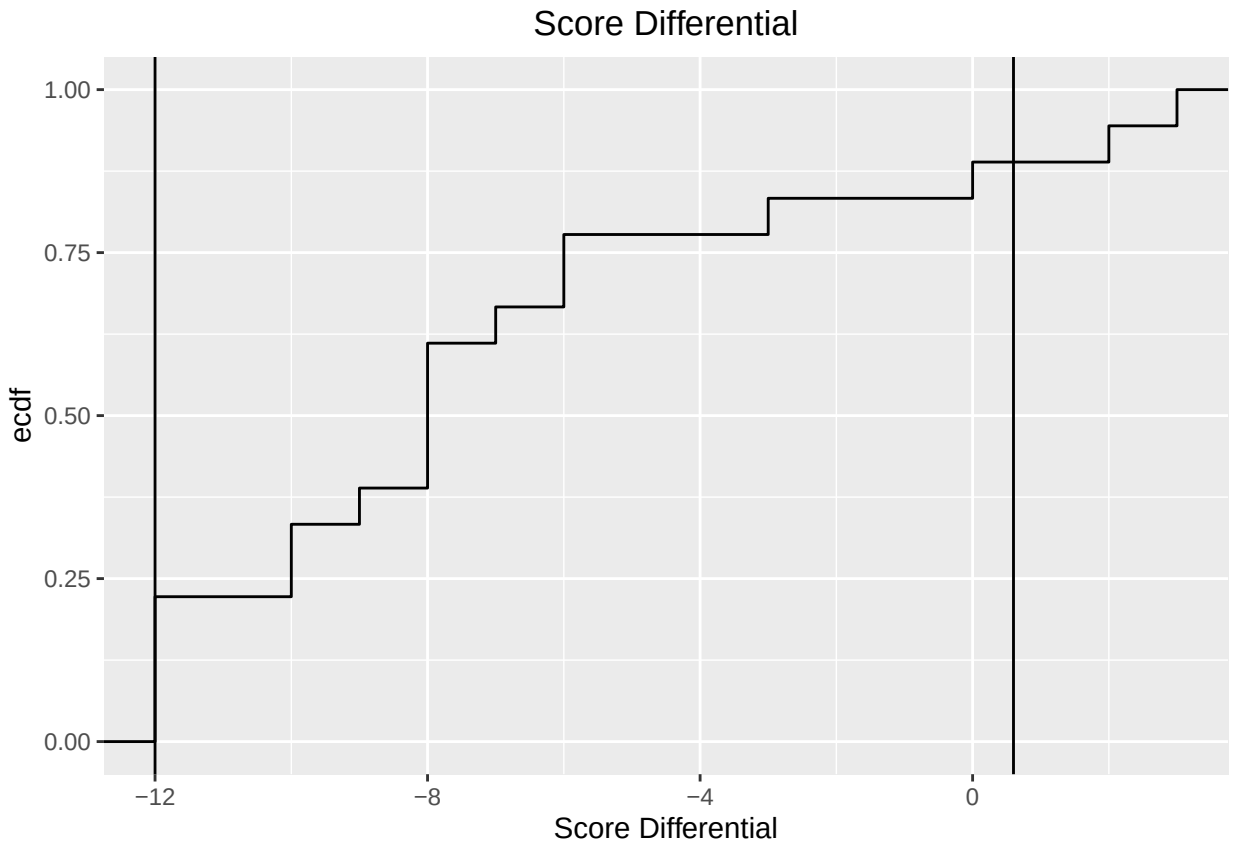
```
singular_game <- singular_game %>% rename('LINEUP SECONDS' = `LINEUP MINUTES`) %>% mutate(LINEUP_SORTED
  if (is.na(l)) return(NA)
  paste(sort(strsplit(l, ", ")[[1]]), collapse = " ")
}))
```

```
singular_game <- subset(singular_game, !((`SCORE DIFFERENTIAL WHEN ENTER` <= -11 | `SCORE DIFFERENTIAL
```

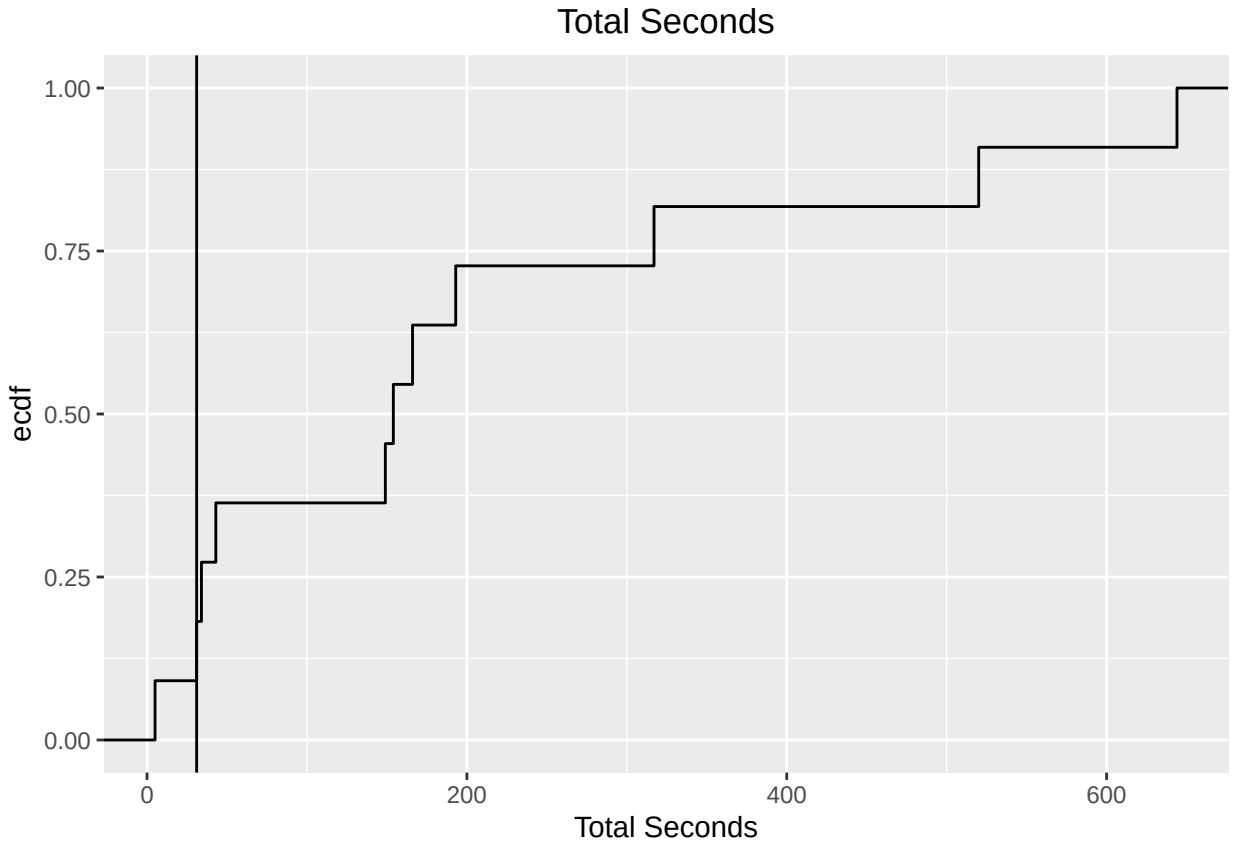
```
game <- singular_game %>% group_by(`LINEUP_SORTED`) %>% summarise(
    `NUMBER OF GUARDS` = mean(`NUMBER OF GUARDS`),
    `OPPONENT POSSESSIONS` = sum(`OPPONENT POSSESSIONS`, na.rm = TRUE),
    `CMU POSSESSIONS` = sum(`CMU POSSESSIONS`, na.rm = TRUE),
    `LINEUP SECONDS` = sum(`LINEUP SECONDS`, 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))
ggplot(singular_game, aes(x = `SCORE DIFFERENTIAL WHEN ENTER`)) + stat_ecdf() + geom_vline(xintercept =
```




```
# see where to cut time -> SHOULD DO THIS AFTER OR BEFORE CUT SCRAP MINUTES?
p <- quantile(game$`LINEUP SECONDS`,probs=c(0.1))
ggplot(game, aes(x = `LINEUP SECONDS`)) + stat_ecdf() + geom_vline(xintercept = p) + labs(title = "Tota
```




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

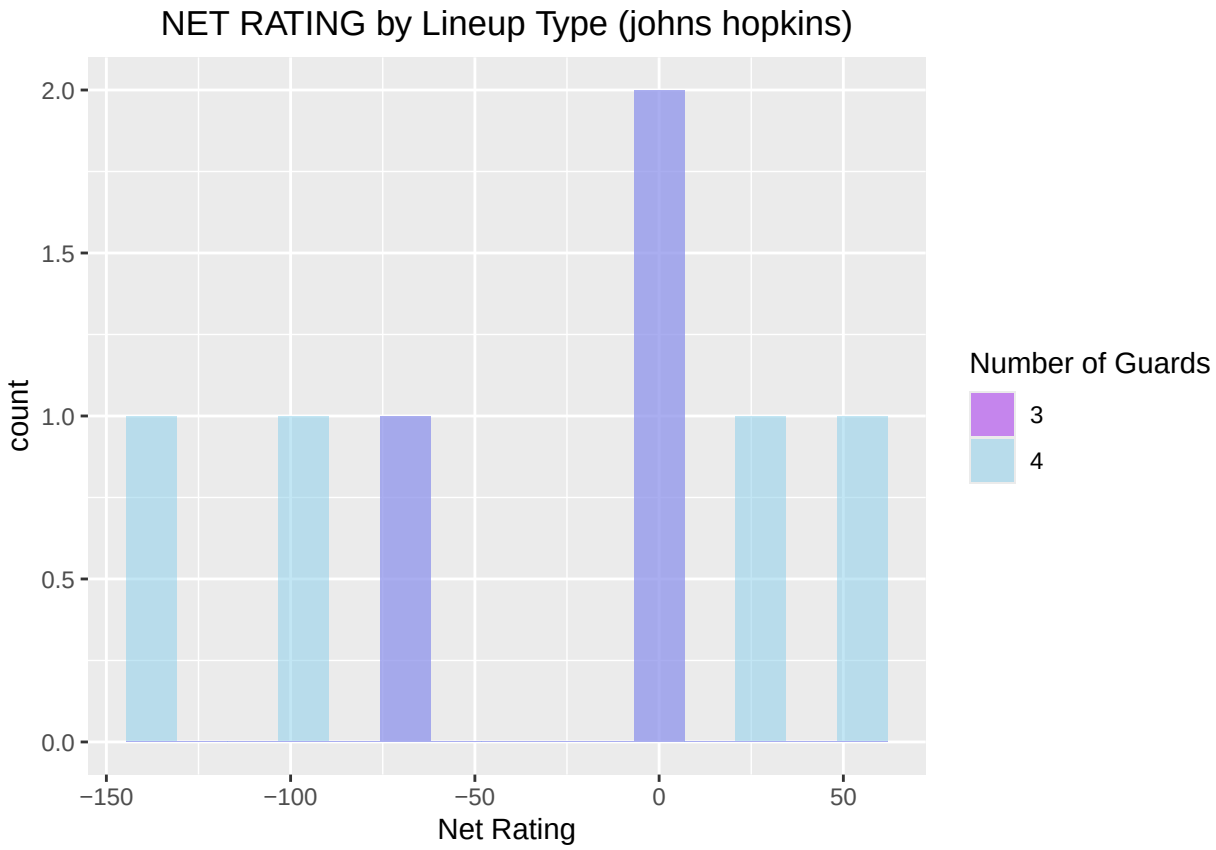
```
## 10%
##  31
```

```
#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()`).
```




```
n3 <- sum(game$`NUMBER OF GUARDS` == 3)
n4 <- sum(game$`NUMBER OF GUARDS` == 4)

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

```
## $`3`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##  -74.29  -37.14    0.00  -24.76    0.00    0.00       1
##
## $`4`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -133.33  -85.71    0.00  -30.74   15.00   59.56
```

```
nr3m <- median(game$`NET RATING`[game$`NUMBER OF GUARDS` %in% c(3)], na.rm = TRUE)
nr4m <- median(game$`NET RATING`[game$`NUMBER OF GUARDS` %in% c(4)], na.rm = TRUE)
nr3m
```

```
## [1] 0
```

```
nr4m
```

```
## [1] 0
```

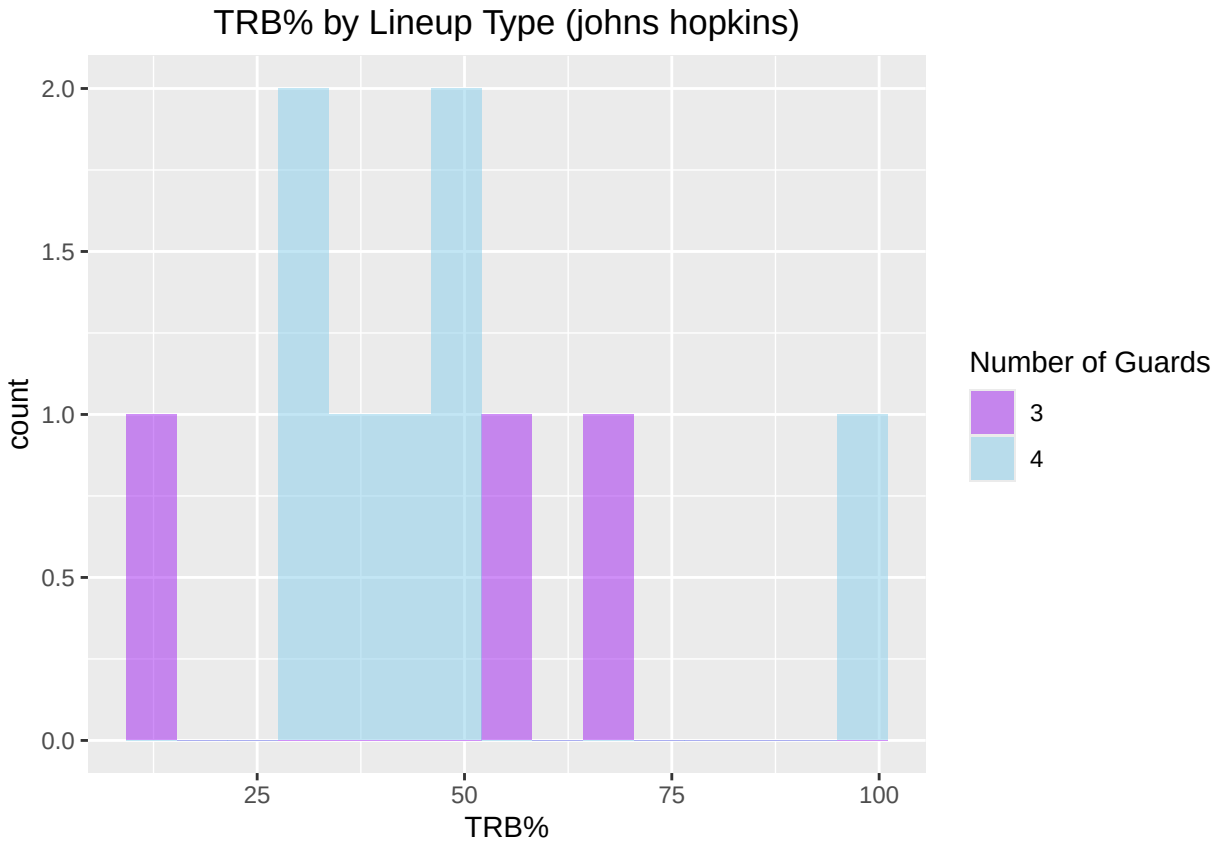
```
nr_p
```

```
## [1] 0.1876828
```

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

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

```
## Warning: Removed 1 row 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` %
```

```
## $`3`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   14.29   33.46   52.63   44.53   59.65   66.67       1
##
## $`4`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   33.33   35.42   44.44   49.80   50.00  100.00
```

```
r3m <- median(game$`TRB%`[game$`NUMBER OF GUARDS` %in% c(3)], na.rm = TRUE)
r4m <- median(game$`TRB%`[game$`NUMBER OF GUARDS` %in% c(4)], na.rm = TRUE)
r_p <- wilcox.test(`TRB%` ~ `NUMBER OF GUARDS`, data = subset(game, `NUMBER OF GUARDS` %in% t_f), exact

r3m
```

```
## [1] 52.63158
```

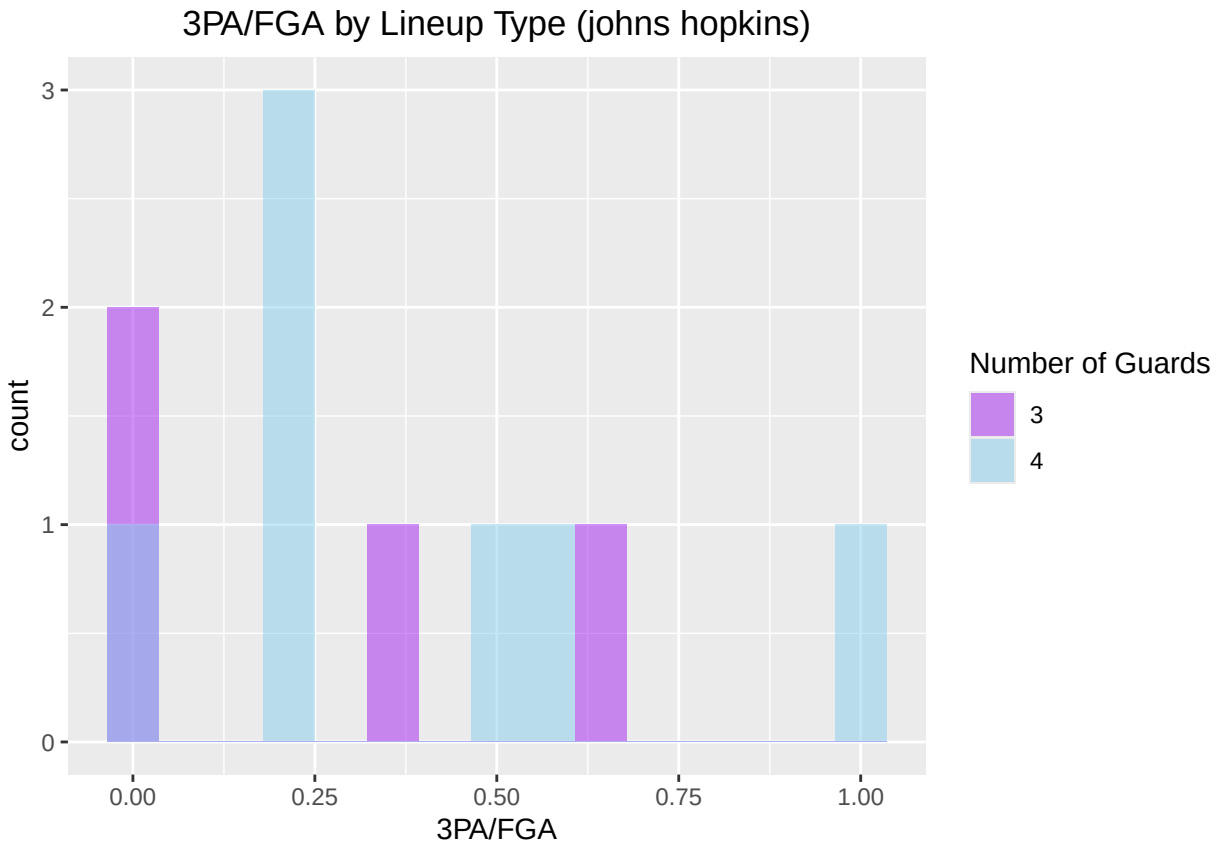
```
r4m
```

```
## [1] 44.44444
```

```
r_p
```

```
## [1] 0.818624
```

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

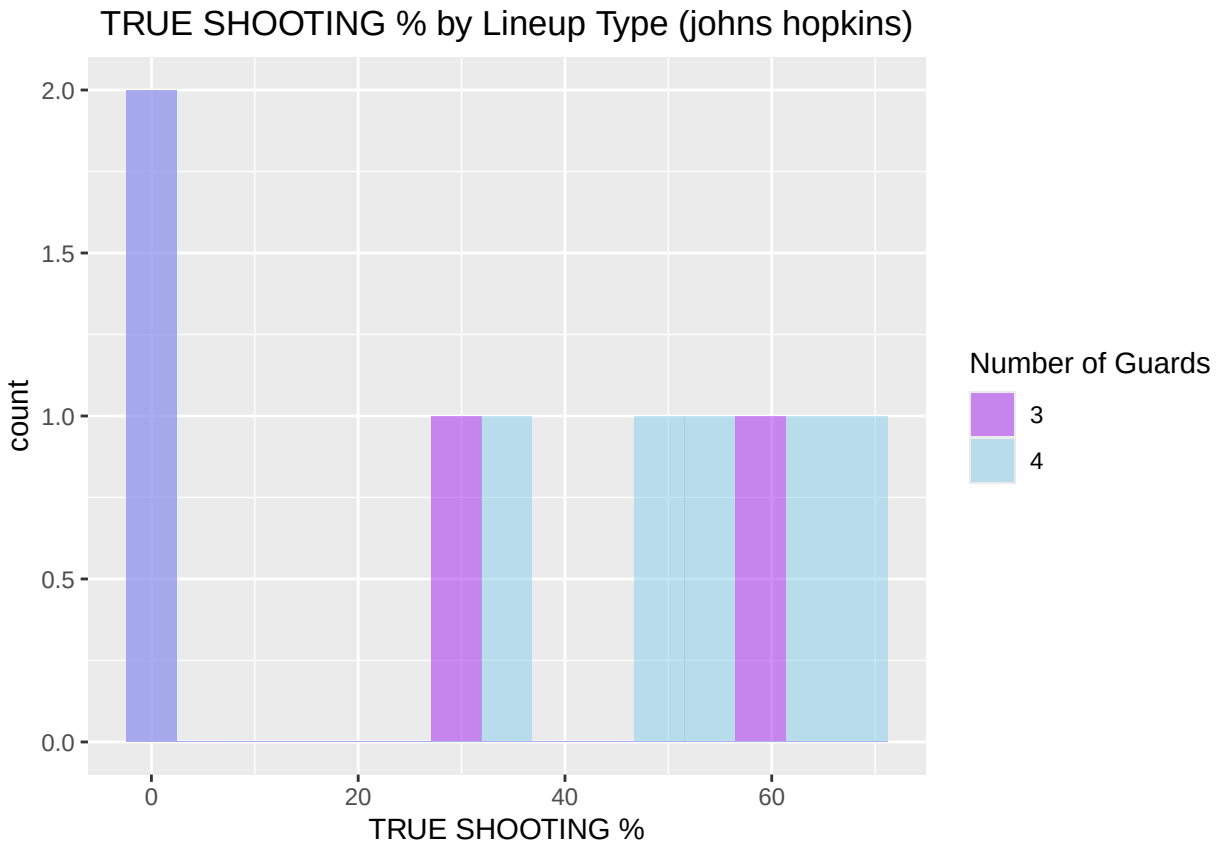



```
tapply(game$`3PA/FGA`[game$`NUMBER OF GUARDS` %in% t_f], game$`NUMBER OF GUARDS`[game$`NUMBER OF GUARDS
```

```
## $`3`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  0.1667  0.2372  0.4038  0.6154
##
## $`4`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.2250  0.2500  0.3912  0.5192  1.0000
```

```
three3m <- median(game$`3PA/FGA`[game$`NUMBER OF GUARDS` %in% c(3)], na.rm = TRUE)
three4m <- median(game$`3PA/FGA`[game$`NUMBER OF GUARDS` %in% c(4)], na.rm = TRUE)
three_p <- wilcox.test(`3PA/FGA` ~ `NUMBER OF GUARDS`, data = subset(game, `NUMBER OF GUARDS` %in% t_f)
```

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

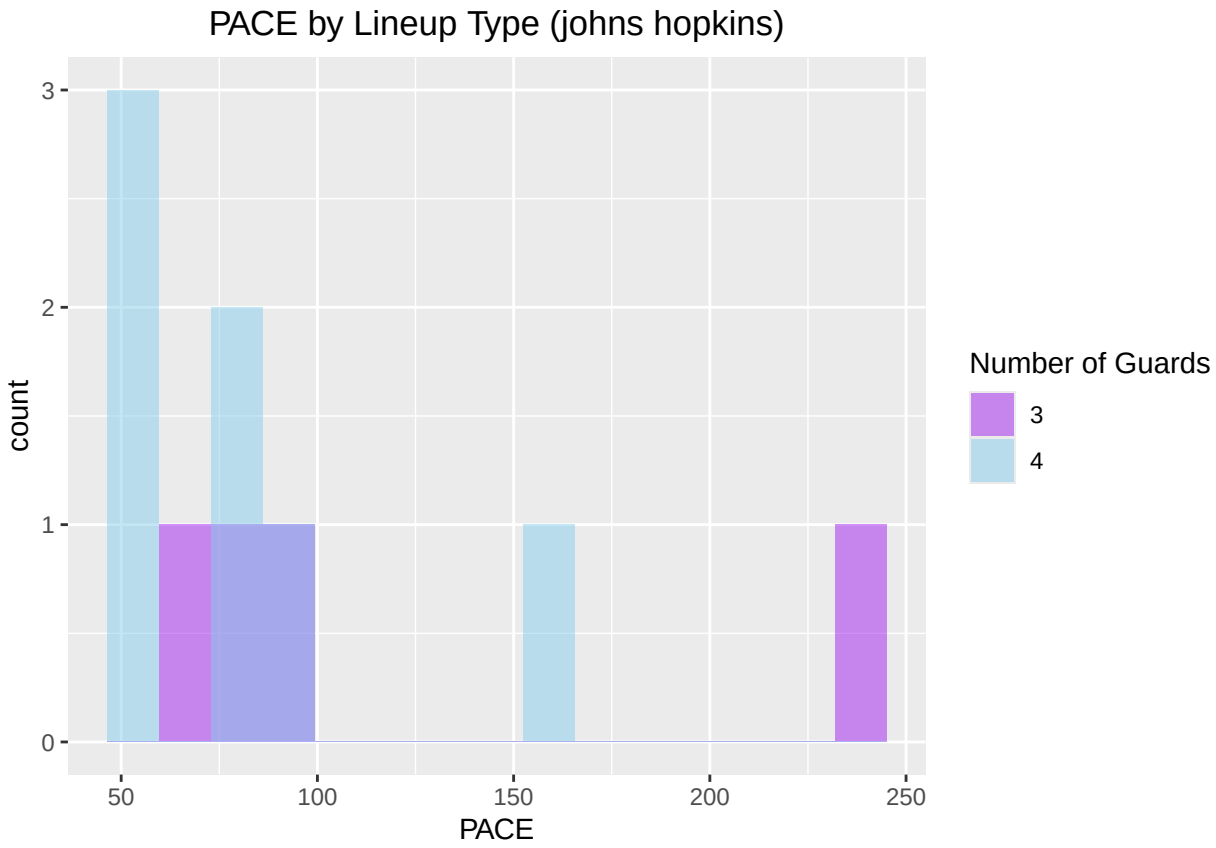



```
tapply(game$`TRUE SHOOTING %`[game$`NUMBER OF GUARDS` %in% t_f], game$`NUMBER OF GUARDS`[game$`NUMBER OF
```

```
## $`3`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    0.00   15.38   22.28   37.66   58.33
##
## $`4`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   17.01   50.00   38.69   59.02   68.75
```

```
ts3m <- median(game$`TRUE SHOOTING %`[game$`NUMBER OF GUARDS` %in% c(3)], na.rm = TRUE)
ts4m <- median(game$`TRUE SHOOTING %`[game$`NUMBER OF GUARDS` %in% c(4)], na.rm = TRUE)
ts_p <- wilcox.test(`TRUE SHOOTING %` ~ `NUMBER OF GUARDS`, data = subset(game, `NUMBER OF GUARDS` %in%
```

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



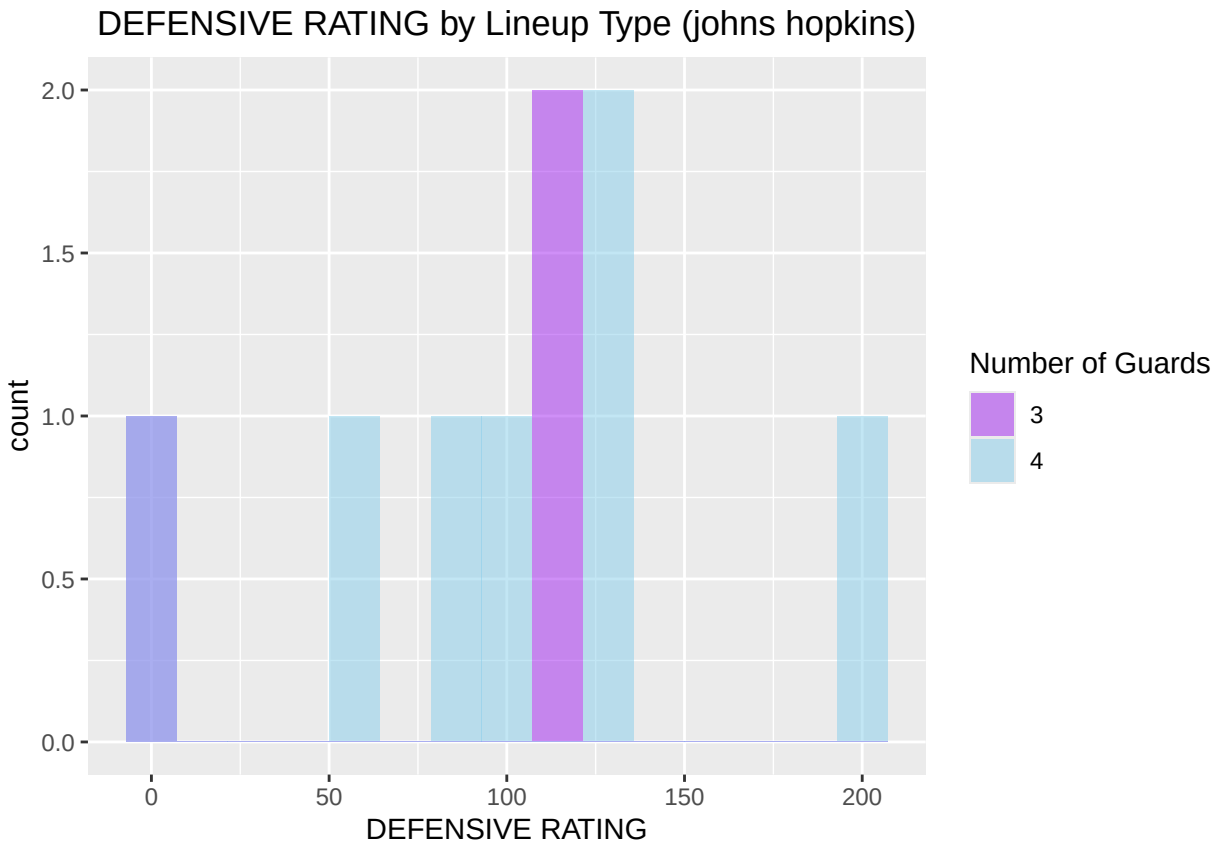

```
tapply(game$`PACE`[game$`NUMBER OF GUARDS` %in% t_f], game$`NUMBER OF GUARDS`[game$`NUMBER OF GUARDS` %
```

```
## $`3`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   70.59   74.95   81.57  118.43  125.06  240.00
##
## $`4`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   54.55   56.09   75.71   80.07   81.60  154.84
```

```
p3m <- median(game$`PACE`[game$`NUMBER OF GUARDS` %in% c(3)], na.rm = TRUE)
p4m <- median(game$`PACE`[game$`NUMBER OF GUARDS` %in% c(4)], na.rm = TRUE)
p_p <- wilcox.test(`PACE` ~ `NUMBER OF GUARDS`, data = subset(game, `NUMBER OF GUARDS` %in% t_f), exact
```

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

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

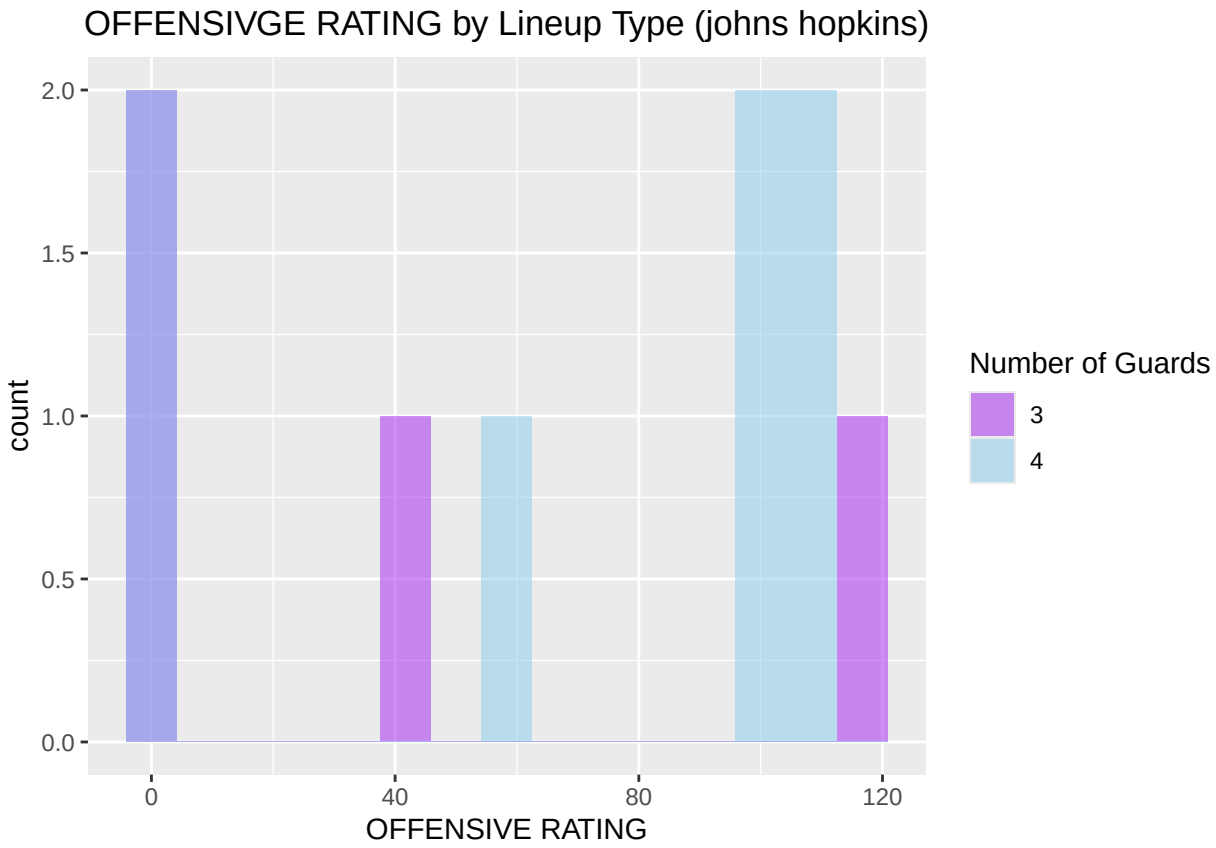



```
tapply(game$`DEFENSIVE RATING`[game$`NUMBER OF GUARDS` %in% t_f], game$`NUMBER OF GUARDS`[game$`NUMBER
```

```
## $`3`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00   57.14  114.29   76.98  115.48  116.67       1
##
## $`4`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   66.47  100.00   99.26  130.95  200.00
```

```
dr3m <- median(game$`DEFENSIVE RATING`[game$`NUMBER OF GUARDS` %in% c(3)], na.rm = TRUE)
dr4m <- median(game$`DEFENSIVE RATING`[game$`NUMBER OF GUARDS` %in% c(4)], na.rm = TRUE)
dr_p <- wilcox.test(`DEFENSIVE RATING` ~ `NUMBER OF GUARDS`, data = subset(game, `NUMBER OF GUARDS` %in
```

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




```
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.
##    0.00    0.00   20.00   39.17   59.17  116.67
##
## $`4`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   28.57  100.00   68.52  105.00  112.50
```

```
or3m
```

```
## [1] 20
```

```
or4m
```

```
## [1] 116.6667
```

```
or3m <- median(game$`OFFENSIVE RATING`[game$`NUMBER OF GUARDS` %in% c(3)], na.rm = TRUE)
or4m <- median(game$`OFFENSIVE RATING`[game$`NUMBER OF GUARDS` %in% c(4)], na.rm = TRUE)
or_p <- wilcox.test(`OFFENSIVE RATING` ~ `NUMBER OF GUARDS`, data = subset(game, `NUMBER OF GUARDS` %in
```

```
individual_games <<- individual_games %>% add_row(
  `GAME` = g,
  `SCORE` = " ",
  `3G` = n3,
  `4G` = n4,
  `3G MEDIAN NET RATING` = round(nr3m,2),
  `4G MEDIAN NET RATING` = round(nr4m,2),
  `NET RATING DIFFERENCE` = round(abs(nr3m - nr4m), 2),
```

```
  `NET RATING MANN-WHITNEY P-VALUE` = round(nr_p,2),
  `3G MEDIAN TRB%` = round(r3m,2),
  `4G MEDIAN TRB%` = round(r4m,2),
  `TRB% DIFFERENCE` = round(abs(r3m - r4m),2),
  `TRB% MANN-WHITNEY P-VALUE` = round(r_p,2),
  `3G MEDIAN 3PA/FGA` = round(three3m,2),
  `4G MEDIAN 3PA/FGA` = round(three4m,2),
  `3PA/FGA DIFFERENCE` = round(abs(three3m - three4m),2),
  `3PA/FGA MANN-WHITNEY P-VALUE` = round(three_p,2),
  `3G MEDIAN TRUE SHOOTING %` = round(ts3m,2),
  `4G MEDIAN TRUE SHOOTING %` = round(ts4m,2),
  `TRUE SHOOTING % DIFFERENCE` = round(abs(ts3m - ts4m),2),
  `TRUE SHOOTING % MANN-WHITNEY P-VALUE` = round(ts_p,2),
  `3G MEDIAN PACE` = round(p3m,2),
  `4G MEDIAN PACE` = round(p4m,2),
  `PACE DIFFERENCE` = round(abs(p3m - p4m),2),
  `PACE MANN-WHITNEY P-VALUE` = round(p_p,2),
  `3G MEDIAN DEFENSIVE RATING` = round(dr3m,2),
  `4G MEDIAN DEFENSIVE RATING` = round(dr4m,2),
  `DEFENSIVE RATING DIFFERENCE` = round(abs(dr3m - dr4m),2),
  `DEFENSIVE RATING MANN-WHITNEY P-VALUE` = round(dr_p,2),
  `3G MEDIAN OFFENSIVE RATING` = round(or3m,2),
  `4G MEDIAN OFFENSIVE RATING` = round(or4m,2),
  `OFFENSIVE RATING DIFFERENCE` = round(abs(or3m - or4m),2),
  `OFFENSIVE RATING MANN-WHITNEY P-VALUE` = round(or_p,2)
)

# hard coded -> FIX LATER
game_order <- c("allegheny", "penn state-behrend", "muskingum", "oberlin", "denison", "carlow", "wooste

individual_games <- individual_games %>% arrange(factor(`GAME`, levels = game_order))
```