Project Check in 1

2024-10-03

Below you will find the template for Project Check in 1. Each group member should submit a project check in document. Work with your team to ensure you limit duplicated work. The example code below is instructional and should not appear in your submission. Please do feel free to adapt it to your own dataset.

1. ***If your dataset has changed*** from what you submitted for Project Check in 0, please record:

- number of observations (rows)
- number of variables (columns)
- number of missing values
- names of particular columns of interest (if there are too many to print all of them!)
- data source and links to any accompanying documentation

```r
library(readr)
dline_stats <- read_csv("Data/Game_Logs_Defensive_Lineman.csv")
```

```
## Rows: 317599 Columns: 25
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (21): Player Id, Name, Position, Season, Game Date, Home or Away, Oppone...
## dbl  (4): Year, Week, Games Played, Yards Per Int
##
## 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.
```

```r
nrow(dline_stats)
```

```
## [1] 317599
```

```r
ncol(dline_stats)
```

```
## [1] 25
```

```r
colnames(dline_stats)
```

```
##  [1] "Player Id"          "Name"               "Position"
##  [4] "Year"               "Season"             "Week"
##  [7] "Game Date"          "Home or Away"       "Opponent"
## [10] "Outcome"            "Score"              "Games Played"
## [13] "Games Started"      "Total Tackles"      "Solo Tackles"
## [16] "Assisted Tackles"   "Sacks"              "Safties"
## [19] "Passes Defended"    "Ints"               "Int Yards"
## [22] "Yards Per Int"      "Longest Int Return" "Ints for TDs"
## [25] "Forced Fumbles"
```

```
dline_stats$Sacks <- as.numeric(dline_stats$Sacks)
```

```
## Warning: NAs introduced by coercion
```

```
dline_stats$`Total Tackles` <- as.numeric(dline_stats$`Total Tackles`)
```

```
## Warning: NAs introduced by coercion
```

Fisher, R. A. (1936) The use of multiple measurements in taxonomic problems. Annals of Eugenics, 7, Part II, 179–188. doi:10.1111/j.1469-1809.1936.tb02137.x.

2. Show summary statistics for the five variables of the most interest.

```
summary(dline_stats$Outcome)
```

```
##    Length     Class      Mode 
##    317599 character character
```

```
summary(dline_stats$Sacks)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##   0.000   0.000   0.000   0.165   0.000   7.000  157463
```

```
summary(dline_stats$Ints)
```

```
##    Length     Class      Mode 
##    317599 character character
```

```
summary(dline_stats$`Total Tackles`)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##   0.000   0.000   1.000   2.267   4.000  24.000  148167
```

```
summary(dline_stats$`Forced Fumbles`)
```

```
##    Length     Class      Mode 
##    317599 character character
```

```
#range(dline_stats$`Home or Away`)
```

3. Show another set of summary statistics by filtering on a column of interest. This could be years, teams, genres. Dig a little deeper here! Think about what might have a different distribution!

```
home_df <- dline_stats[dline_stats$`Home or Away` =="Home",]
away_df <- dline_stats[dline_stats$`Home or Away`=="Away",]

home_wins <- home_df[home_df$Outcome == 'W',]
home_losses <- home_df[home_df$Outcome == 'L',]
home_win_pct <- nrow(home_wins) / (nrow(home_wins) + nrow(home_losses)) * 100

away_wins <- away_df[away_df$Outcome == 'W',]
away_losses <- away_df[away_df$Outcome == 'L',]
away_win_pct <- nrow(away_wins) / (nrow(away_wins) + nrow(away_losses)) * 100

win_comp <- data.frame(
  Location = c("Home", "Away"),
  win_percentage <- c(home_win_pct, away_win_pct)
)

print(win_comp)
```

```
##   Location win_percentage....c.home_win_pct..away_win_pct.
## 1     Home                                        57.83871
## 2     Away                                        42.45274
```

4. Visualize the distribution of at least three variables. For example, below I've made a histogram to investigate the distribution of Petal Length. I've colored them by species. This is the base R way. You'll notice that we've had to be clever about how we set the x-axis here, it has to encompass the full range of Petal.Length.

I expect plots to have sensible, nice-looking labels.

```
dline_stats$'Solo Tackles' <- as.numeric(dline_stats$'Solo Tackles')
```

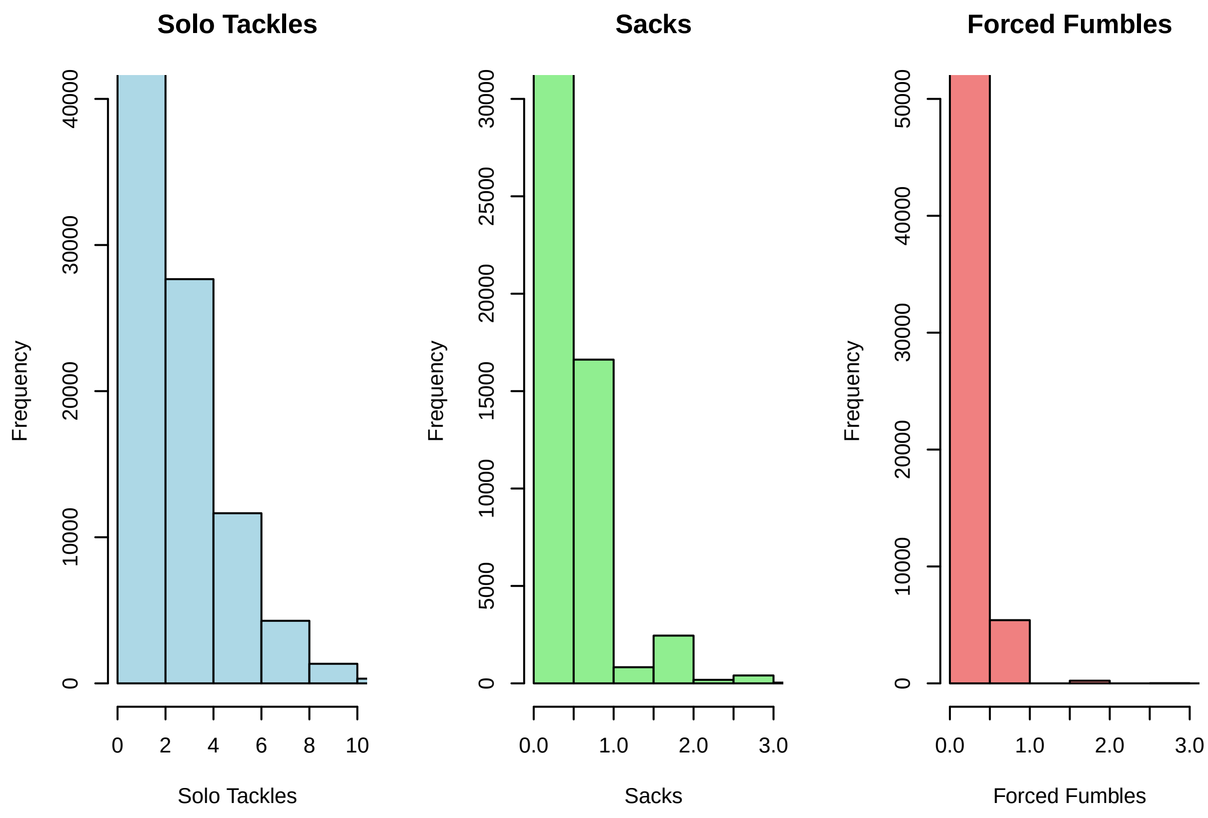
```
## Warning: NAs introduced by coercion
```

```
dline_stats$'Sacks' <- as.numeric(dline_stats$'Sacks')
dline_stats$'Forced Fumbles' <- as.numeric(dline_stats$'Forced Fumbles')
```

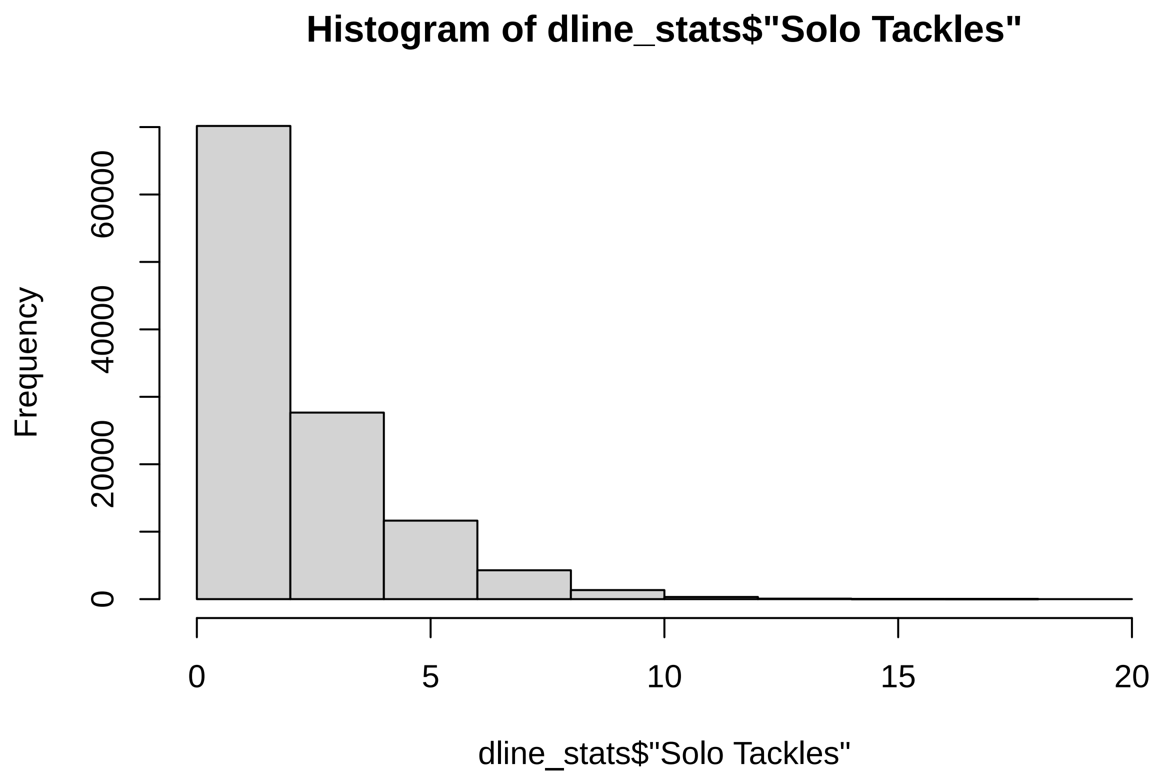
```
## Warning: NAs introduced by coercion
```

```
par(mfrow = c(1, 3))

h1 <- hist(dline_stats$'Solo Tackles',
           breaks = 10,
           main = "Solo Tackles",
           xlab = "Solo Tackles",
           col = "lightblue",
           border = "black",
           xlim = c(0, 10),
           ylim = c(0, 40000))

h2 <- hist(dline_stats$'Sacks',
           breaks = 10,
```

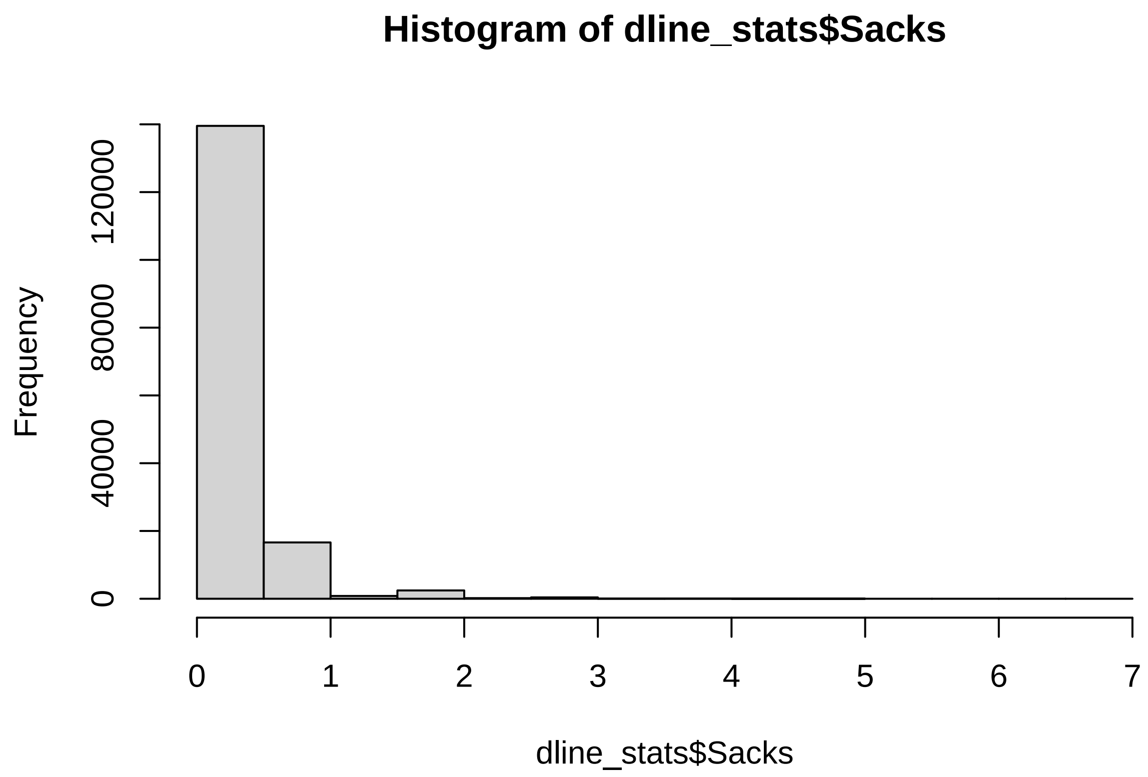
```r
           main = "Sacks",
           xlab = "Sacks",
           col = "lightgreen",
           border = "black",
           xlim = c(0, 3),
           ylim = c(0, 30000))

h3 <- hist(dline_stats$'Forced Fumbles',
           breaks = 10,
           main = "Forced Fumbles",
           xlab = "Forced Fumbles",
           col = "lightcoral",
           border = "black",
           xlim = c(0, 3),
           ylim = c(0, 50000))
```



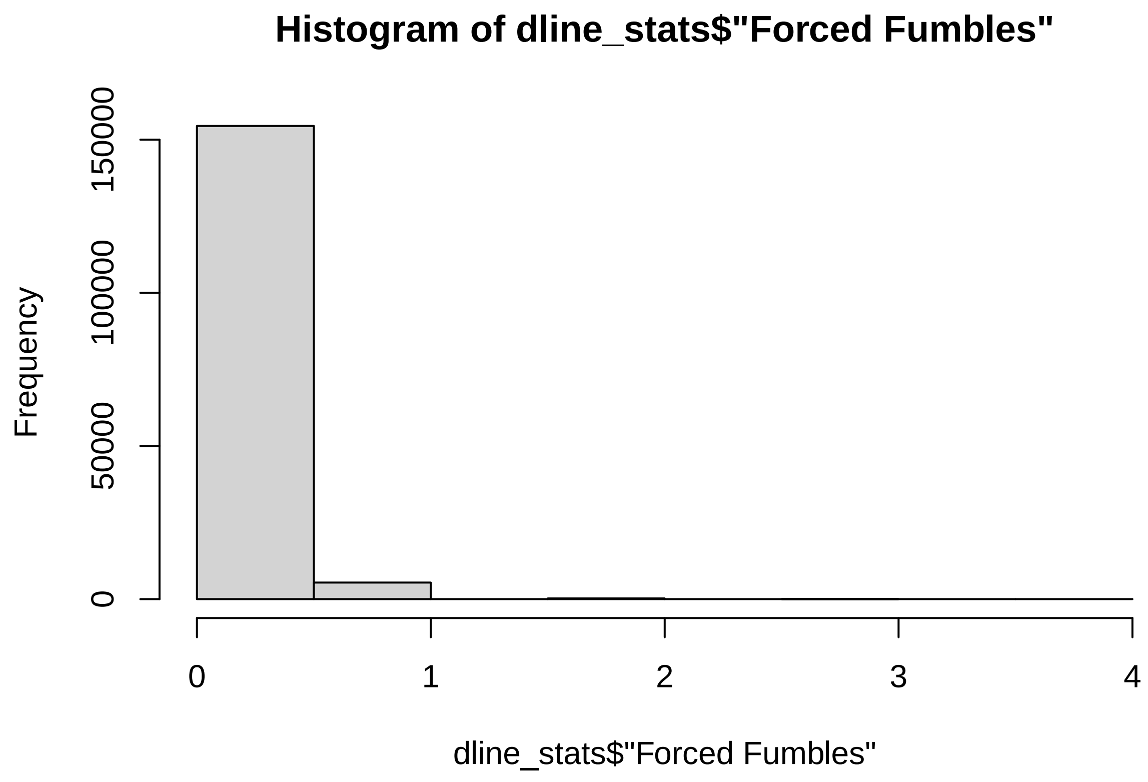
```r
par(mfrow = c(1, 1))

plot(h1)
```


Histogram of dline_stats$"Solo Tackles"
Frequency
0
20000
40000
60000
0
5
10
15
20
dline_stats$"Solo Tackles"

```r
plot(h2)
```

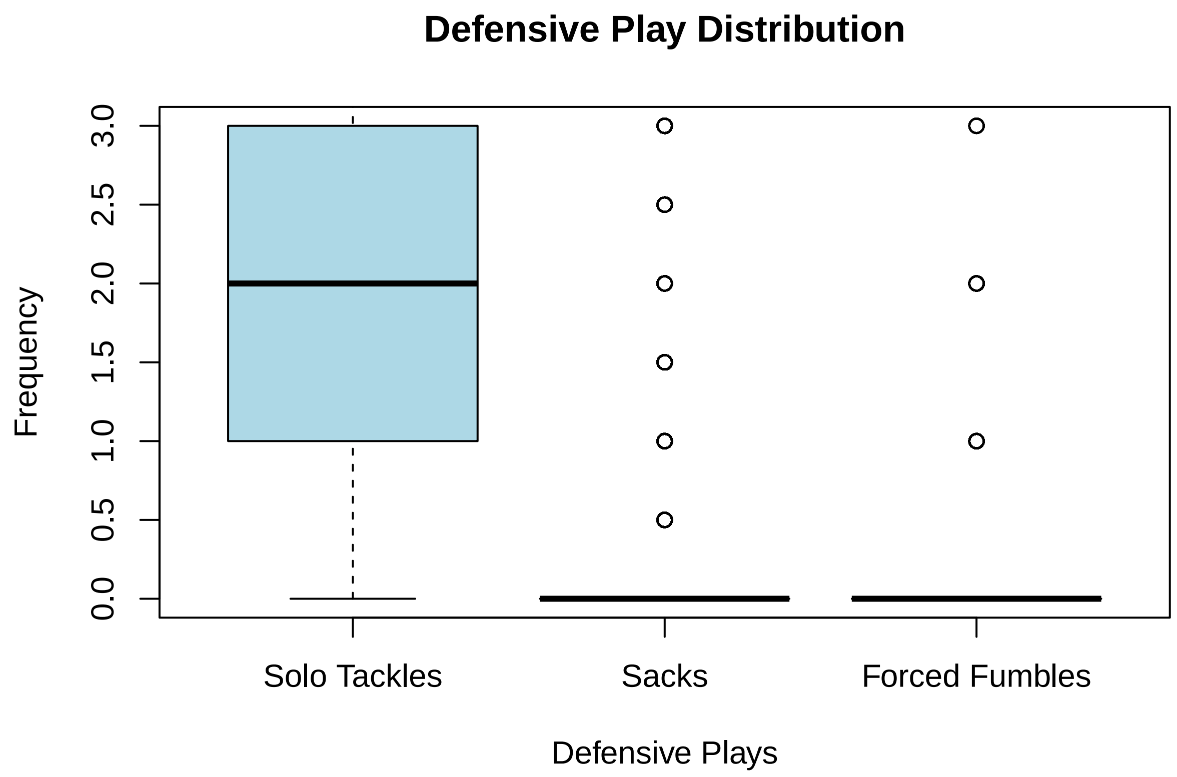

Histogram of dline_stats$Sacks
Frequency
0
40000
80000
120000
0
1
2
3
4
5
6
7
dline_stats$Sacks

```r
plot(h3)
```



Where appropriate, this can be a series of boxplots, for ex:

```r
combined_data <- list(
  "Solo Tackles" = dline_stats$'Solo Tackles',
  "Sacks" = dline_stats$'Sacks',
  "Forced Fumbles" = dline_stats$'Forced Fumbles'
)

# Plot the boxplot for the combined data
boxplot(combined_data,
        main = "Defensive Play Distribution",
        xlab = "Defensive Plays",
        ylab = "Frequency",
        col = c("lightblue", "lightgreen", "lightcoral"),
        border = "black",
        ylim = c(0,3))
```



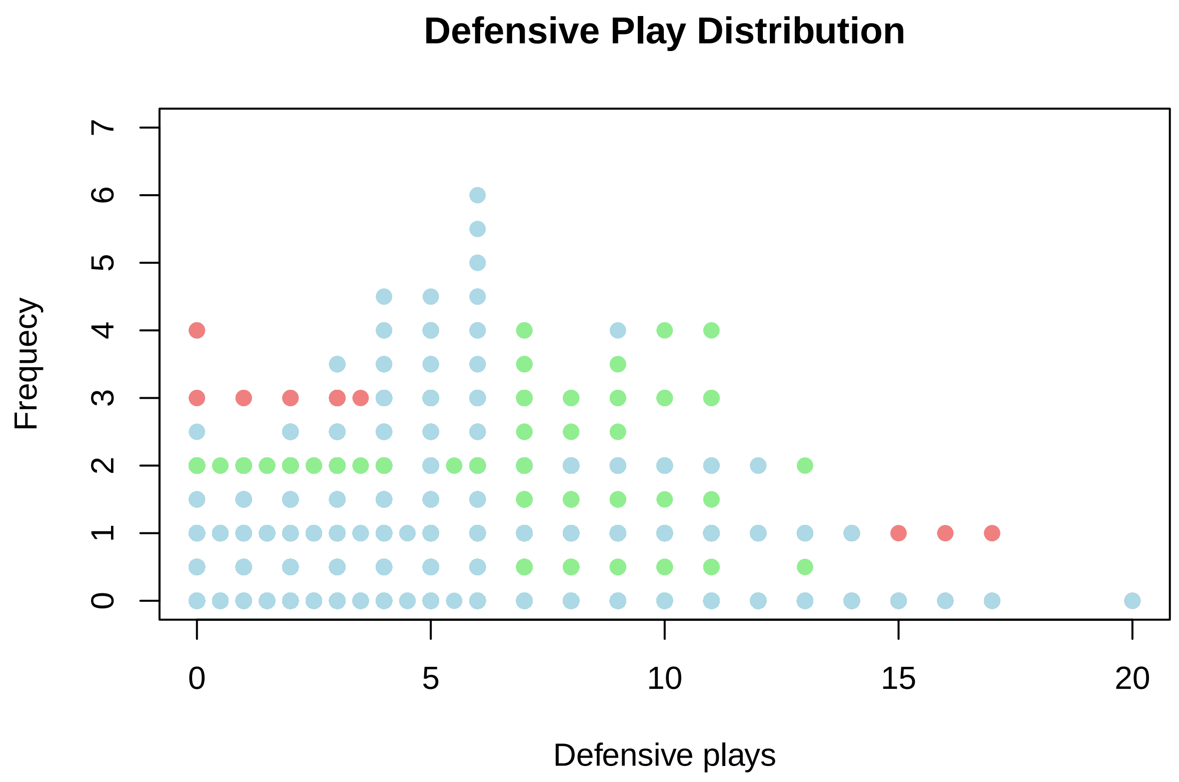
3. Show three scatterplots that show the relationship between variables. Coloring data is a useful dimension to add here! There are many different ways to generate color palettes in R, but the general process here is the same:

- define color palette, here my_colors (or pick your own hex codes or a fancy color brewer package)
- map the variable to the variables. We've done this using fields::color.scale() in class. Or get fancy with for loops and overplotting
- generate plot!

Extremely optional: You can turn your points into emojis with the emojifont package

```
my_colors <- c("lightblue", "lightgreen", "lightcoral")

solo_tackles_colors <- my_colors[cut(dline_stats$'Solo Tackles', breaks = length(my_colors), labels = F
sacks_colors <- my_colors[cut(dline_stats$'Sacks', breaks = length(my_colors), labels = FALSE)]
forced_fumbles_colors <- my_colors[cut(dline_stats$'Forced Fumbles', breaks = length(my_colors), labels

plot(dline_stats$'Solo Tackles', dline_stats$'Sacks',
     col = solo_tackles_colors, pch = 19,
     xlab = "Defensive plays", ylab = "Frequecy",
     main = "Defensive Play Distribution",
     xlim = c(0, max(dline_stats$'Solo Tackles', na.rm = TRUE)),
     ylim = c(0, max(dline_stats$'Sacks', na.rm = TRUE)))
points(dline_stats$'Solo Tackles', dline_stats$'Forced Fumbles',
       col = sacks_colors, pch = 19)
```

```r
points(dline_stats$'Sacks', dline_stats$'Forced Fumbles',
       col = forced_fumbles_colors, pch = 19)
```



4. Write a few sentences about the observations you see in the above plots. Provide any context where necessary.