Final Project EDA

Jackson Gasperack & Sawan Pandita

2025-11-26

Table of contents

Front Matter

```r
# Add libraries
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.2
v ggplot2   4.0.0     v tibble    3.3.0
v lubridate 1.9.4     v tidyr     1.3.1
v purrr     1.1.0
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
```

```r
library(ggplot2)
library(stringr)
```

```r
# Load in data files
zip_raw <- "https://codeload.github.com/jgasperackpsu/stat380_finalproject/zip/refs/heads/mai

temp <- tempfile(fileext = ".zip")
temp_dir = tempdir()
download.file(zip_raw, temp)
unzip(temp, exdir = temp_dir)

csv_files <- list.files(temp_dir, pattern = "\\.csv$", full.names = TRUE,
                        recursive = TRUE)

# Convert files to datasets
datasets <- lapply(csv_files, read_csv)
```

```
Rows: 3885 Columns: 24
-- Column specification --------------------------------------------------------
Delimiter: ","
chr  (4): TEAM, CONF, POSTSEASON, SEED
dbl (20): G, W, ADJOE, ADJDE, BARTHAG, EFG_O, EFG_D, TOR, TORD, ORB, DRB, FT...

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.
Rows: 347 Columns: 23
-- Column specification --------------------------------------------------------
Delimiter: ","
chr  (3): TEAM, CONF, POSTSEASON
dbl (20): G, W, ADJOE, ADJDE, BARTHAG, EFG_O, EFG_D, TOR, TORD, ORB, DRB, FT...

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.
Rows: 351 Columns: 23
-- Column specification --------------------------------------------------------
Delimiter: ","
chr  (3): TEAM, CONF, POSTSEASON
dbl (20): G, W, ADJOE, ADJDE, BARTHAG, EFG_O, EFG_D, TOR, TORD, ORB, DRB, FT...

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.
Rows: 351 Columns: 23
-- Column specification --------------------------------------------------------
Delimiter: ","
chr  (3): TEAM, CONF, POSTSEASON
```

```
dbl (20): G, W, ADJOE, ADJDE, BARTHAG, EFG_O, EFG_D, TOR, TORD, ORB, DRB, FT...

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.
Rows: 351 Columns: 23
-- Column specification --------------------------------------------------------
Delimiter: ","
chr  (3): TEAM, CONF, POSTSEASON
dbl (20): G, W, ADJOE, ADJDE, BARTHAG, EFG_O, EFG_D, TOR, TORD, ORB, DRB, FT...

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.
Rows: 351 Columns: 23
-- Column specification --------------------------------------------------------
Delimiter: ","
chr  (3): TEAM, CONF, POSTSEASON
dbl (20): G, W, ADJOE, ADJDE, BARTHAG, EFG_O, EFG_D, TOR, TORD, ORB, DRB, FT...

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.
Rows: 351 Columns: 23
-- Column specification --------------------------------------------------------
Delimiter: ","
chr  (3): TEAM, CONF, POSTSEASON
dbl (20): G, W, ADJOE, ADJDE, BARTHAG, EFG_O, EFG_D, TOR, TORD, ORB, DRB, FT...

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.
Rows: 353 Columns: 23
-- Column specification --------------------------------------------------------
Delimiter: ","
chr  (3): TEAM, CONF, POSTSEASON
dbl (20): G, W, ADJOE, ADJDE, BARTHAG, EFG_O, EFG_D, TOR, TORD, ORB, DRB, FT...

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.
Rows: 353 Columns: 22
-- Column specification --------------------------------------------------------
Delimiter: ","
chr  (2): TEAM, CONF
dbl (20): RK, G, W, ADJOE, ADJDE, BARTHAG, EFG_O, EFG_D, TOR, TORD, ORB, DRB...

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.
Rows: 347 Columns: 23
-- Column specification --------------------------------------------------------
Delimiter: ","
chr  (3): TEAM, CONF, POSTSEASON
dbl (20): G, W, ADJOE, ADJDE, BARTHAG, EFG_O, EFG_D, TOR, TORD, ORB, DRB, FT...

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.
Rows: 358 Columns: 23
-- Column specification --------------------------------------------------------
Delimiter: ","
chr  (4): TEAM, CONF, POSTSEASON, SEED
dbl (19): G, W, ADJOE, ADJDE, BARTHAG, EFG_O, EFGD_D, TOR, TORD, ORB, DRB, F...

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.
Rows: 363 Columns: 23
-- Column specification --------------------------------------------------------
Delimiter: ","
chr  (4): TEAM, CONF, POSTSEASON, SEED
dbl (19): G, W, ADJOE, ADJDE, BARTHAG, EFG_O, EFG_D, TOR, TORD, ORB, DRB, FT...

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.
Rows: 362 Columns: 23
-- Column specification --------------------------------------------------------
Delimiter: ","
chr  (4): TEAM, CONF, POSTSEASON, SEED
dbl (19): G, W, ADJOE, ADJDE, BARTHAG, EFG%, EFGD%, TOR, TORD, ORB, DRB, FTR...

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.
Rows: 364 Columns: 25
-- Column specification --------------------------------------------------------
Delimiter: ","
chr  (3): Team, CONF, SEED
dbl (22): RK, G, W, ADJOE, ADJDE, BARTHAG, EFG_O, EFG_D, TOR, TORD, ORB, DRB...

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
names(datasets) <- tools::file_path_sans_ext(basename(csv_files))

names <- rep(NA, length(datasets))
for(i in seq_along(datasets)){
  names[i] <- names(datasets[i])
}

for(name in names){
  assign(name, datasets[[name]])
}
```

Cleaning

```r
# Some datasets have character integers
cleanSEEDS <- function(seed){
  ifelse(seed == "N/A", NA, as.numeric(seed))
}

cbb22$SEED <- sapply(cbb22$SEED, cleanSEEDS)
cbb23$SEED <- sapply(cbb23$SEED, cleanSEEDS)
cbb24$SEED <- sapply(cbb24$SEED, cleanSEEDS)
cbb25$SEED <- sapply(cbb25$SEED, cleanSEEDS)
cbb$SEED <- sapply(cbb$SEED, cleanSEEDS)

print(str(cbb25))
```

```
spc_tbl_ [364 x 25] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
 $ RK     : num [1:364] 1 2 3 4 5 6 7 8 9 10 ...
 $ Team   : chr [1:364] "Houston" "Duke" "Auburn" "Florida" ...
 $ CONF   : chr [1:364] "B12" "ACC" "SEC" "SEC" ...
 $ G      : num [1:364] 34 34 33 34 33 34 33 33 34 33 ...
 $ W      : num [1:364] 30 31 28 30 25 27 25 24 22 25 ...
 $ ADJOE  : num [1:364] 125 128 129 128 128 ...
 $ ADJDE  : num [1:364] 88 91.3 93.7 94 96.4 91.1 96.7 92.3 96.5 92 ...
 $ BARTHAG: num [1:364] 0.982 0.981 0.976 0.971 0.962 ...
 $ EFG_O  : num [1:364] 52.7 57.4 55.7 55 56.3 52.8 55.5 54.6 52.9 53.8 ...
 $ EFG_D  : num [1:364] 44.9 44.5 46 45.3 47.9 44.4 48 48.4 48.6 47 ...
 $ TOR    : num [1:364] 14.6 14.4 13.4 15 16.7 16 14.8 16.9 16.2 14.5 ...
 $ TORD   : num [1:364] 21.7 17.7 17.4 17 13.5 17.8 17.4 21.8 16.6 20.2 ...
 $ ORB    : num [1:364] 36.1 35.2 34.3 38.1 34.7 35.4 34.3 32.2 35.9 31 ...
 $ DRB    : num [1:364] 29.3 26.5 30.3 28.8 29.2 30.1 28.5 28.4 28.5 27.2 ...
 $ FTR    : num [1:364] 28.2 32.1 33.5 32.6 40.1 35.4 29.7 38.5 36.6 32.9 ...
 $ FTRD   : num [1:364] 34.1 25.4 39.2 33 33.9 29.4 33.5 28.9 29.2 25.9 ...
 $ 2P_O   : num [1:364] 49 58 56.1 56.4 59.7 54.1 54.3 55.1 55.1 52.7 ...
 $ 2P_D   : num [1:364] 43.9 43.4 47.2 45.9 48.8 46.7 48.2 47.5 47.2 47.6 ...
 $ 3P_O   : num [1:364] 39.8 37.7 36.8 35.5 35 34 37.9 35.8 32.4 37.2 ...
 $ 3P_D   : num [1:364] 30.9 30.9 29.2 29.6 30.8 27.8 31.7 33.2 33.6 30.7 ...
 $ 3PR    : num [1:364] 34.5 45.4 40.6 43.6 46.2 43 44.6 36.9 34.6 35.5 ...
 $ 3PRD   : num [1:364] 43.1 37.9 34.8 37.3 35.1 45.1 34.1 40.7 41.9 35.8 ...
 $ ADJ_T  : num [1:364] 61.4 65.7 67.8 69.5 74.6 63.8 65.6 68.5 69.9 69.8 ...
 $ WAB    : num [1:364] 11.6 9.6 12.5 11.1 9.8 9.3 5.8 5.4 5.4 4.6 ...
 $ SEED   : Named num [1:364] 1 1 1 1 2 2 3 3 4 4 ...
  ..- attr(*, "names")= chr [1:364] "1" "1" "1" "1" ...
 - attr(*, "spec")=
  .. cols(
  ..   RK = col_double(),
  ..   Team = col_character(),
  ..   CONF = col_character(),
  ..   G = col_double(),
  ..   W = col_double(),
  ..   ADJOE = col_double(),
  ..   ADJDE = col_double(),
  ..   BARTHAG = col_double(),
  ..   EFG_O = col_double(),
  ..   EFG_D = col_double(),
  ..   TOR = col_double(),
  ..   TORD = col_double(),
  ..   ORB = col_double(),
  ..   DRB = col_double(),
  ..   FTR = col_double(),
  ..   FTRD = col_double(),
  ..   `2P_O` = col_double(),
  ..   `2P_D` = col_double(),
  ..   `3P_O` = col_double(),
  ..   `3P_D` = col_double(),
  ..   `3PR` = col_double(),
  ..   `3PRD` = col_double(),
  ..   ADJ_T = col_double(),
  ..   WAB = col_double(),
  ..   SEED = col_character()
  .. )
 - attr(*, "problems")=<externalptr>
NULL
```

```
unique_vals <- sapply(cbb25, function(x) length(unique(x)))
print(unique_vals)
```

```
     RK     Team     CONF        G        W    ADJOE    ADJDE  BARTHAG    EFG_O    EFG_D 
    364      364       31       10       29      207      200      356      121      111 
    TOR     TORD      ORB      DRB      FTR     FTRD     2P_O     2P_D     3P_O     3P_D 
     88       98      148      113      155      174      136      123      104       96 
    3PR     3PRD    ADJ_T      WAB     SEED 
    174      146      100      207       17 
```

```
# 2020 no postseason -> COVID
# 2025 no postseason -> Season just began
# All other datasets should have same number variables and '20,'25 have 1 less

cbb25 <- cbb25 %>% select(-c("RK", "3PR", "3PRD"))
# Remove stats that only appear in '25
# Remove rank as there's a seed column
# Keep rank in 2020 since there's no Seed column

# Make col names the same
cbb24 <- cbb24 %>% rename(EFG_O = `EFG%`,  EFG_D = `EFGD%`)
cbb22 <- cbb22 %>% rename(EFG_D = EFGD_D)
```

Relationships

Distributions

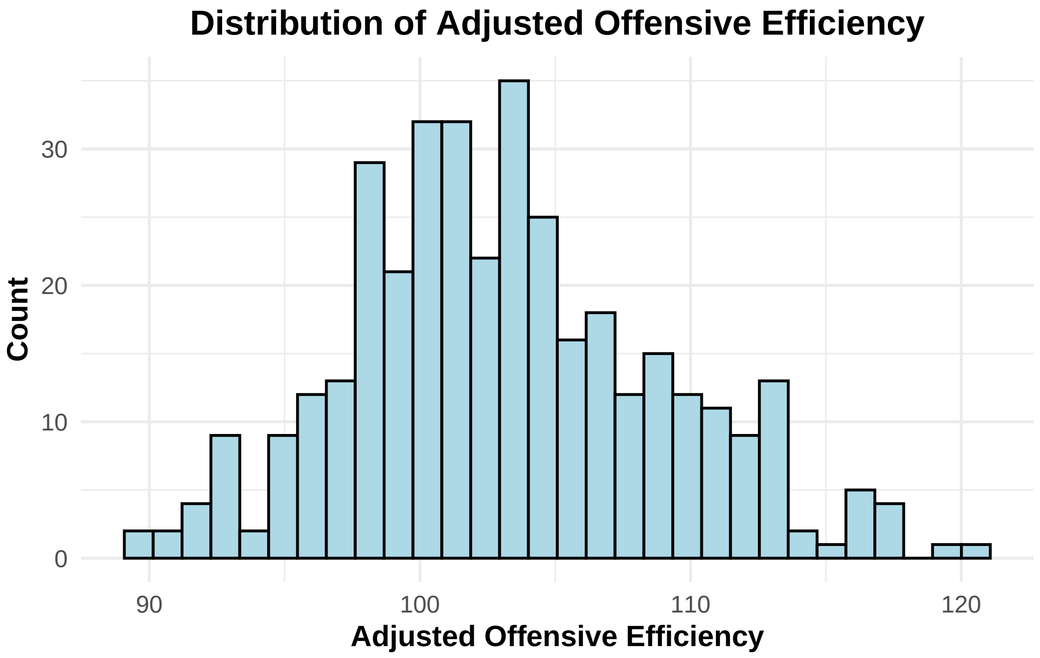
```
# Finding the distributions of our offensive stats

off_stats <- c("ADJOE","EFG_O","TOR","ORB","FTR","2P_O","3P_O")
nice_names <- c(
      "ADJOE"="Adjusted Offensive Efficiency",
      "EFG_O"="Effective Field Goal Rate",
      "TOR"="Turnover Rate",
      "ORB"="Offensive Rebound Rate",
      "FTR"="Free Throw Rate",
      "2P_O"="2-Point FG Rate",
      "3P_O"="3-Point FG Rate"
      )
```

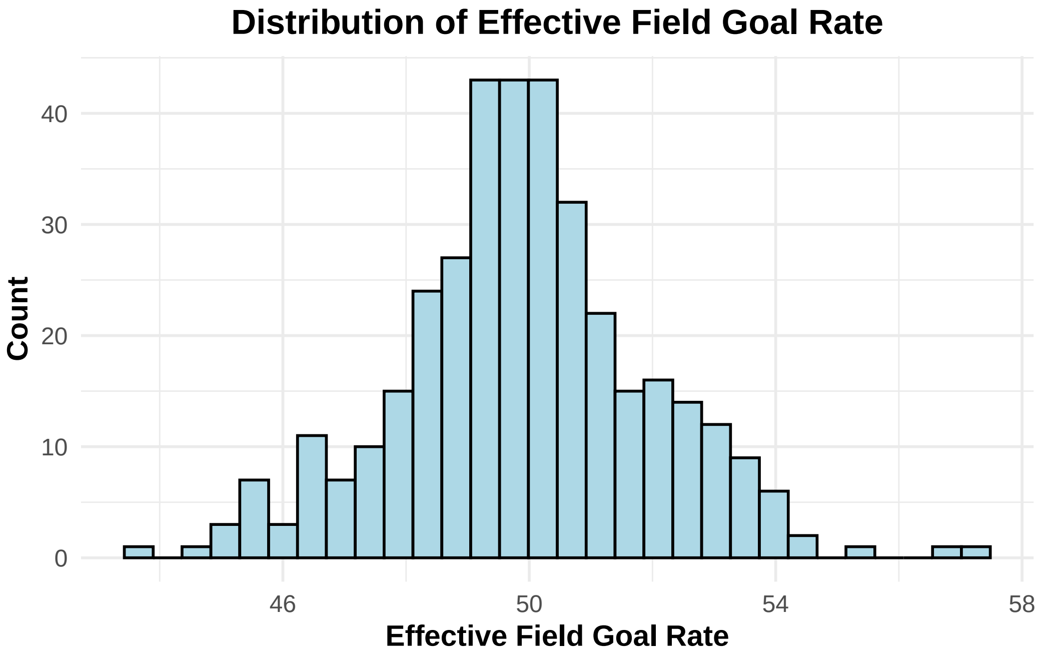
```r
stats_since2013 <- cbb %>%
  group_by(TEAM) %>%
  summarize(
    ADJOE = mean(ADJOE),
    ADJDE = mean(ADJDE),
    EFG_O = mean(EFG_O),
    EFG_D = mean(EFG_D),
    TOR = mean(TOR),
    TORD = mean(TORD),
    ORB = mean(ORB),
    DRB = mean (DRB),
    FTR = mean(FTR),
    FTRD = mean(FTRD),
    `2P_O` = mean(`2P_O`),
    `2P_D` = mean(`2P_D`),
    `3P_O` = mean(`3P_O`),
    `3P_D` = mean(`3P_D`)
  ) %>%
  mutate(across(where(is.numeric), ~round(.x,2)))

for(stat in off_stats){
  print(
  ggplot(data = stats_since2013, mapping = aes(x = .data[[stat]])) +
    geom_histogram(fill = "lightblue", color = "black") +
    labs(title = paste("Distribution of",nice_names[stat]),
         x = nice_names[stat],
         y = "Count") +
    theme_minimal() +
    theme(plot.title = element_text(face = "bold", hjust = 0.5),
          axis.title = element_text(face = "bold"))
  )
}
```

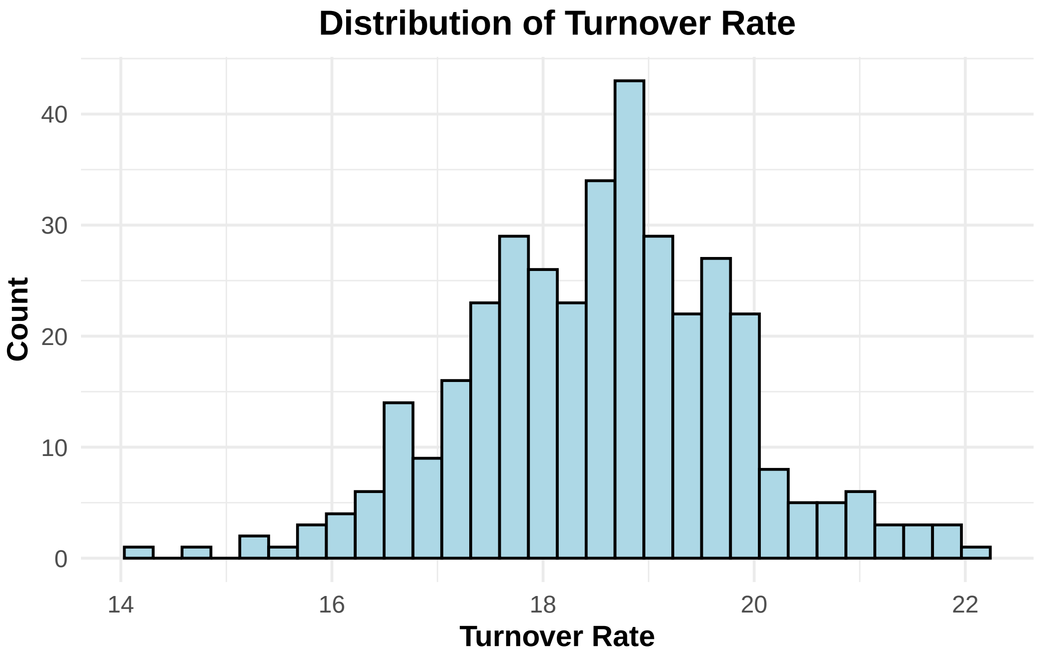
```
`stat_bin()` using `bins = 30`. Pick better value `binwidth`.
```



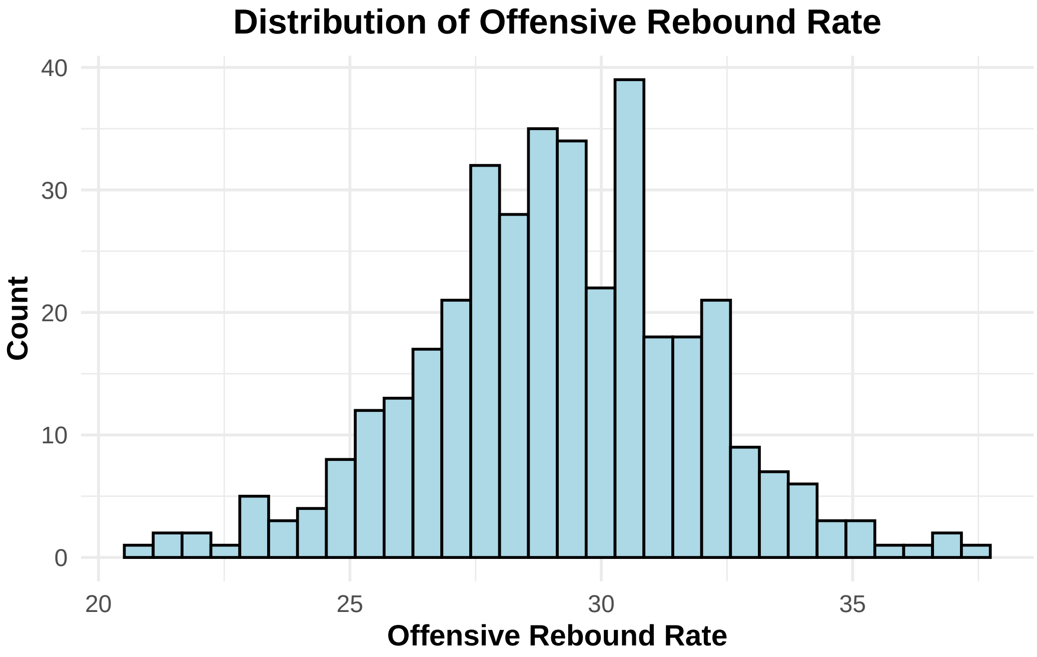
```
`stat_bin()` using `bins = 30`. Pick better value `binwidth`.
```



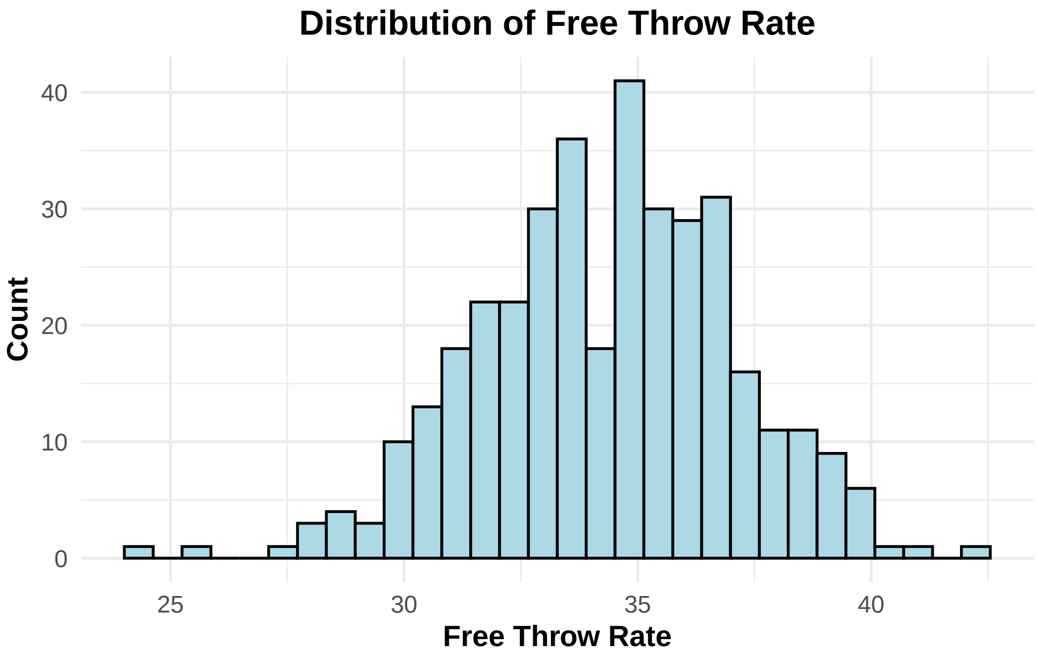
```
`stat_bin()` using `bins = 30`. Pick better value `binwidth`.
```



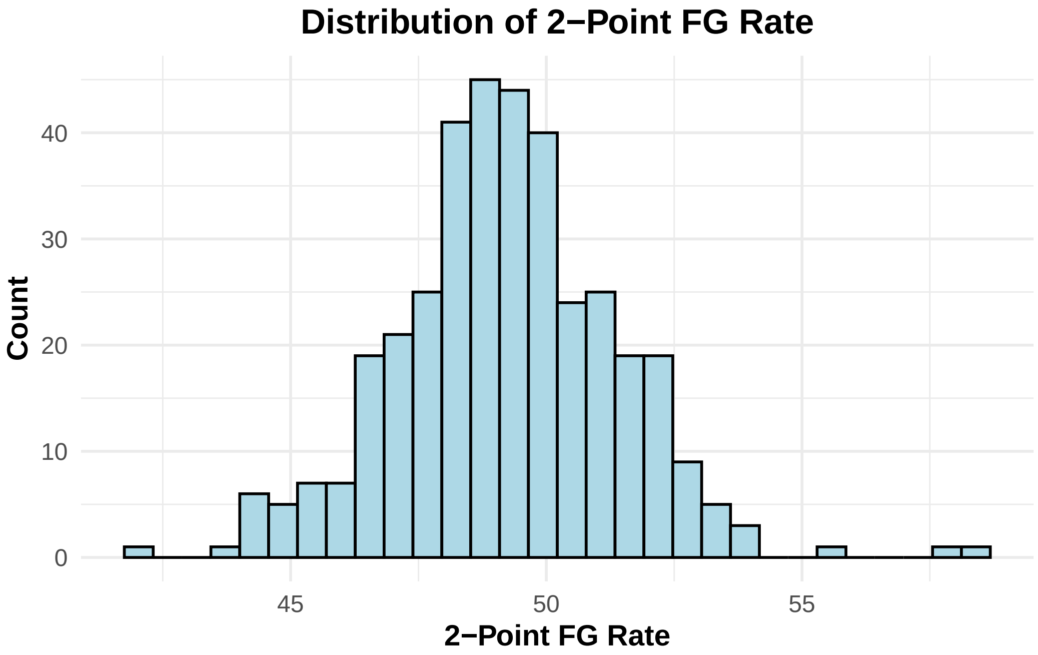
```
`stat_bin()` using `bins = 30`. Pick better value `binwidth`.
```



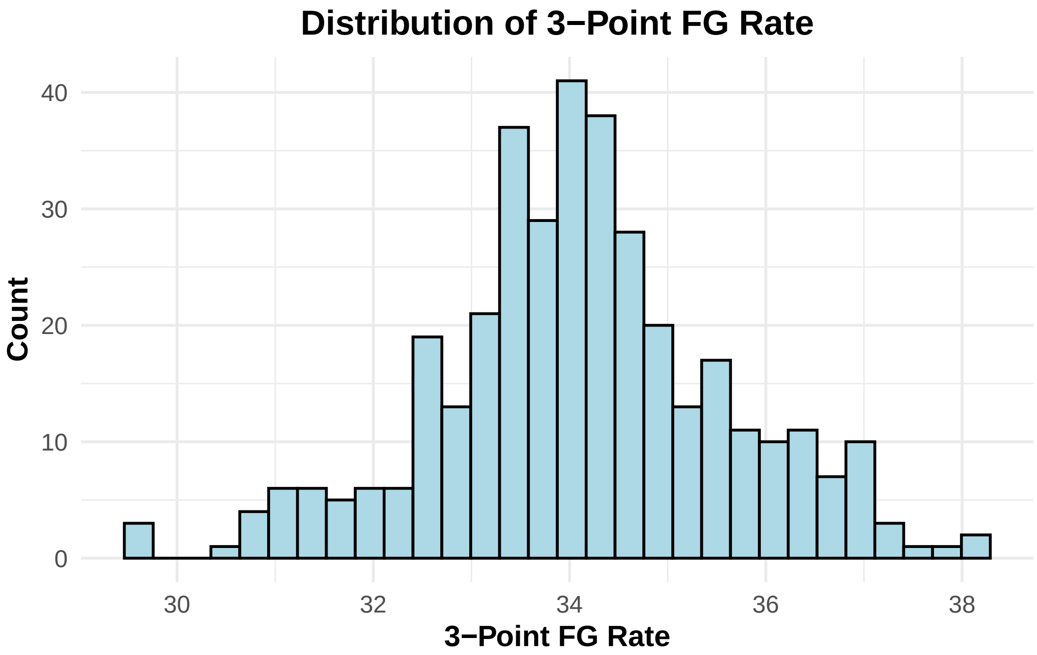
```
`stat_bin()` using `bins = 30`. Pick better value `binwidth`.
```



```
`stat_bin()` using `bins = 30`. Pick better value `binwidth`.
```



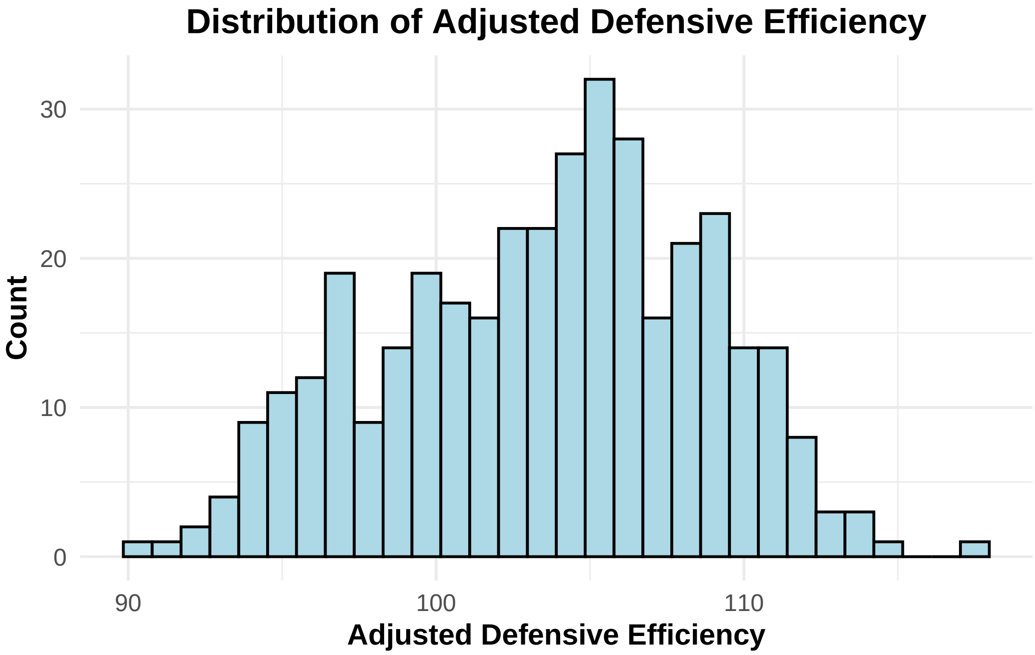
```
`stat_bin()` using `bins = 30`. Pick better value `binwidth`.
```



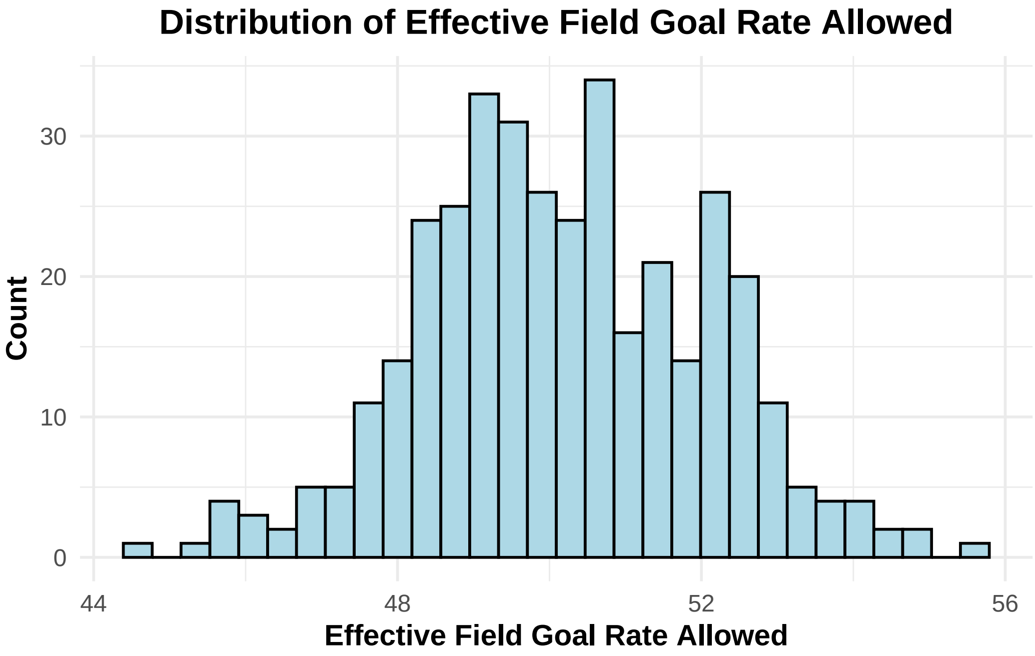
```
def_stats <- c("ADJDE","EFG_D","TORD","DRB","FTRD","2P_D","3P_D")
nice_names_def <- c(
      "ADJDE"="Adjusted Defensive Efficiency",
      "EFG_D"="Effective Field Goal Rate Allowed",
      "TORD"="Turnover Rate Allowed",
      "DRB"="Offensive Rebound Rate Allowed",
      "FTRD"="Free Throw Rate Allowed",
      "2P_D"="2-Point FG Rate Allowed",
      "3P_D"="3-Point FG Rate Allowed"
      )

for(stat in def_stats){
  print(
  ggplot(data = stats_since2013, mapping = aes(x = .data[[stat]])) +
    geom_histogram(fill = "lightblue", color = "black") +
    labs(title = paste("Distribution of",nice_names_def[stat]),
         x = nice_names_def[stat],
         y = "Count") +
    theme_minimal() +
    theme(plot.title = element_text(face = "bold", hjust = 0.5),
          axis.title = element_text(face = "bold"))
  )
}
```

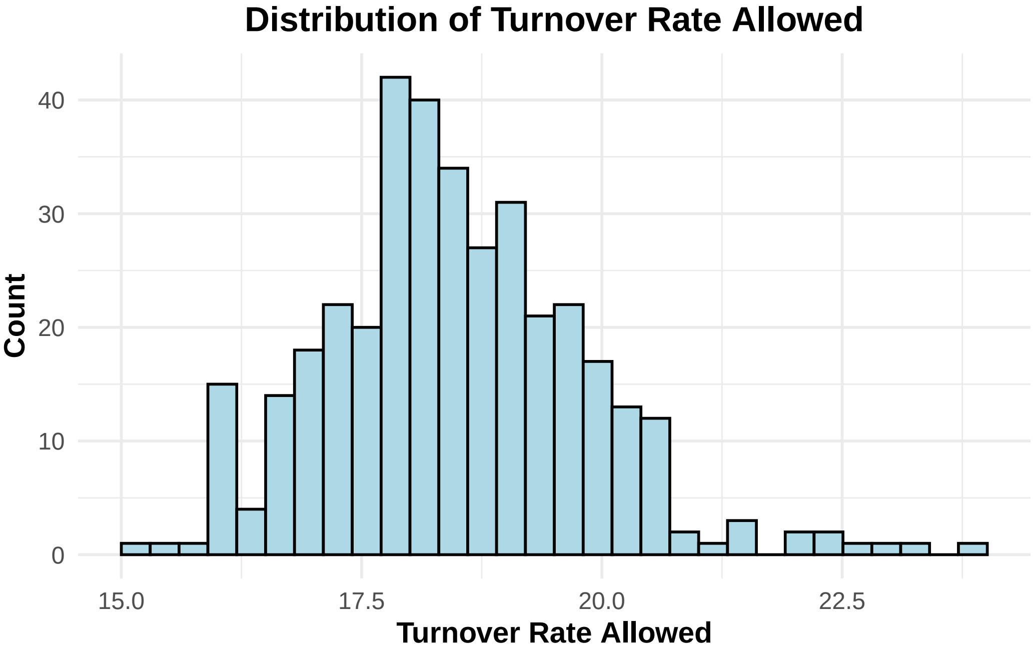
```
`stat_bin()` using `bins = 30`. Pick better value `binwidth`.
```



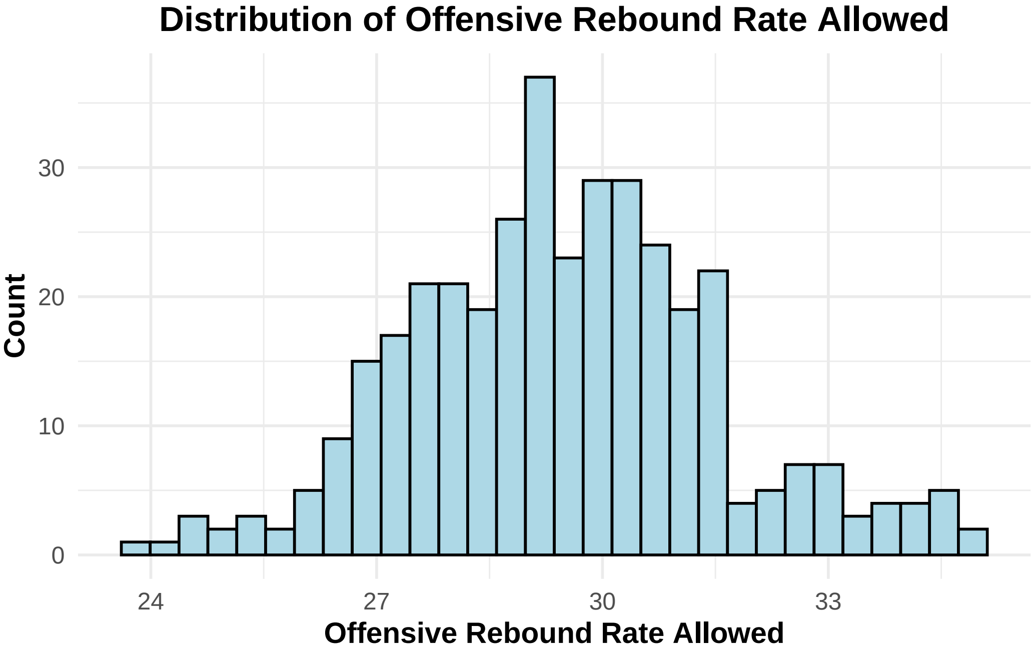
```
`stat_bin()` using `bins = 30`. Pick better value `binwidth`.
```



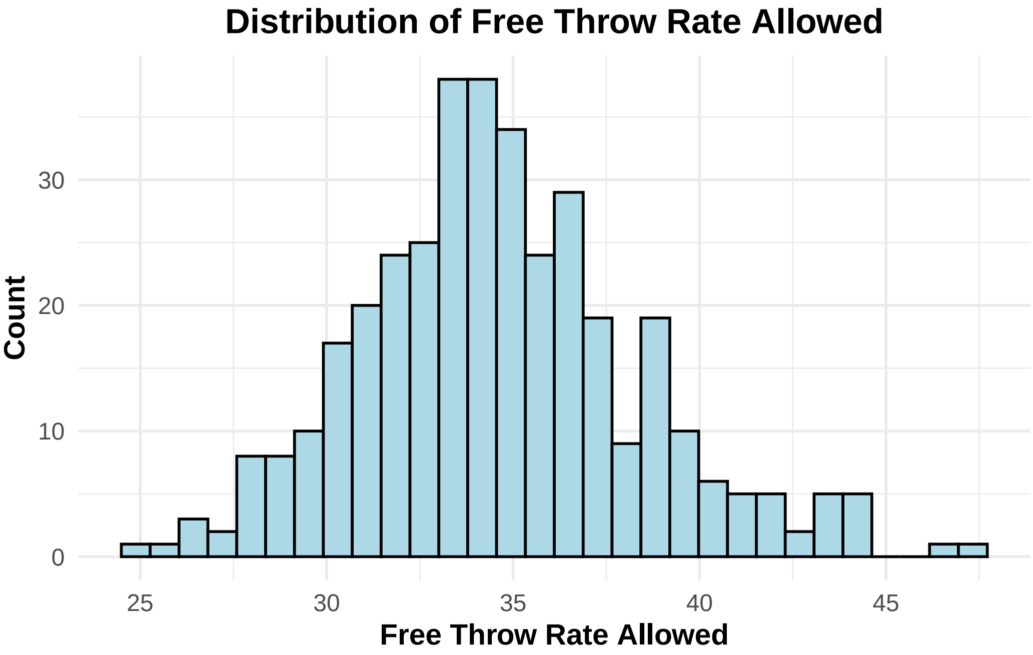
```
`stat_bin()` using `bins = 30`. Pick better value `binwidth`.
```



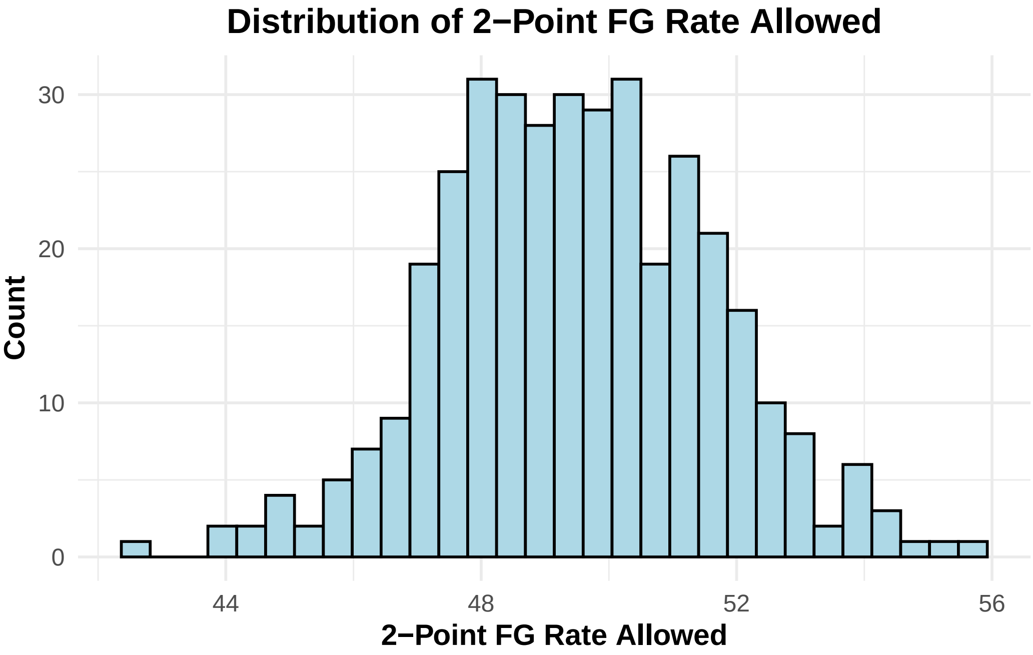
```
`stat_bin()` using `bins = 30`. Pick better value `binwidth`.
```



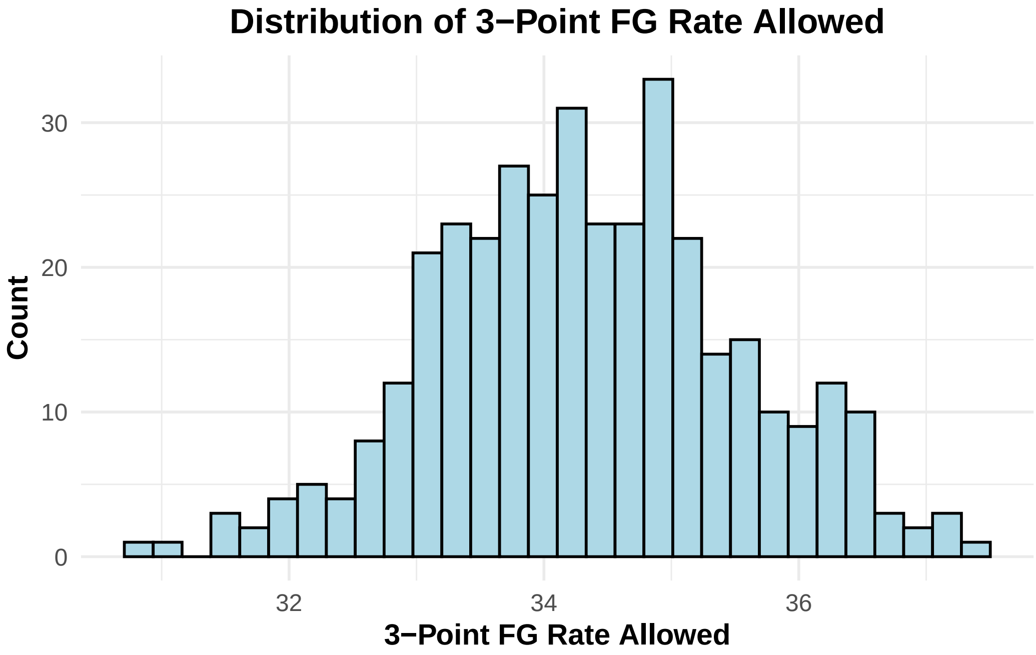
```
`stat_bin()` using `bins = 30`. Pick better value `binwidth`.
```



```
`stat_bin()` using `bins = 30`. Pick better value `binwidth`.
```



```
`stat_bin()` using `bins = 30`. Pick better value `binwidth`.
```



Correlations

```r
seed_game_production <- cbb %>%
  filter(!is.na(SEED)) %>%
  group_by(TEAM, YEAR, SEED) %>%
  summarize(
    ADJOE = mean(ADJOE),
    ADJDE = mean(ADJDE)
  )
```

```
`summarise()` has grouped output by 'TEAM', 'YEAR'. You can override using the
`.groups` argument.
```

```r
nrow(seed_game_production)
```

```
[1] 748
```

```r
sums <- rep(NA,16) # Check that there are 4 of each seed per year
for(i in seq_along(sums)){
  sums[i] <- sum(seed_game_production==i)
}

sumsperyear <- rep(NA,16) # There isn't so we have to fix that
for(i in seq_along(sumsperyear)){
  sumsperyear[i] <- list(table(filter(seed_game_production, SEED == i)$YEAR))
}
```

```r
cbb <- cbb %>%
  mutate(SEED = case_when(
    (TEAM == "Iowa St.") & (YEAR == 2016) ~ 4L,
    (TEAM == "George Washington") & (YEAR == 2014) ~ 9L,
    TRUE ~ SEED
  ),
    POSTSEASON = case_when(
      POSTSEASON == "R68" ~ "R64",
      POSTSEASON == "N/A" ~ NA,
      TRUE ~ POSTSEASON
    )
  )

# There are 6 16 seeds for every year and I found there are 2 other teams of low
# seedings as well, those teams are the first four teams that got beat in the games
# leading up to the march madness so the winners secure their spot in the actual
# tournament
# So instead of removing those teams I'll be naming them a new integer -1

first4_2013 <- c("Middle Tennessee", "Boise St.", "LIU Brooklyn", "Liberty")
first4_2014 <- c("Iowa", "Xavier", "Texas Southern", "Mount St. Mary's")
first4_2015 <- c("Boise St.","BYU","North Florida", "Manhattan")
first4_2016 <- c("Vanderbilt", "Tulsa","Fairleigh Dickinson", "Southern")
first4_2017 <-c("Wake Forest","North Carolina Central","Providence","New Orleans")
first4_2018 <- c("UCLA","Arizona St.","North Carolina Central","LIU Brooklyn")
first4_2019 <- c("Temple","St. John's","Prairie View A&M","North Carolina Central")
first4_2021 <- c("Michigan St.","Wichita St.","Mount St. Mary's","Appalachian St.")
first4_2022 <- c("Rutgers","Wyoming","Texas A&M Corpus Chris","Bryant")
first4_2023 <- c("Nevada","Mississippi St.","Texas Southern","Southeast Missouri St.")
first4_2024 <- c("Virginia","Boise St.","Howard","Montana St.")

cbb <- cbb %>%
  mutate(SEED = case_when(
    (TEAM %in% first4_2013) & (YEAR == 2013) ~ -1,
    (TEAM %in% first4_2014) & (YEAR == 2014) ~ -1,
    (TEAM %in% first4_2015) & (YEAR == 2015) ~ -1,
    (TEAM %in% first4_2016) & (YEAR == 2016) ~ -1,
    (TEAM %in% first4_2017) & (YEAR == 2017) ~ -1,
    (TEAM %in% first4_2018) & (YEAR == 2018) ~ -1,
    (TEAM %in% first4_2019) & (YEAR == 2019) ~ -1,
    (TEAM %in% first4_2021) & (YEAR == 2021) ~ -1,
    (TEAM %in% first4_2022) & (YEAR == 2022) ~ -1,
    (TEAM %in% first4_2023) & (YEAR == 2023) ~ -1,
    (TEAM %in% first4_2024) & (YEAR == 2024) ~ -1,
    TRUE ~ SEED
  ))

new_seed_game_production <- cbb %>%
  filter(!is.na(SEED)) %>%
  group_by(TEAM, YEAR, SEED) %>%
  summarize(
    ADJOE = mean(ADJOE),
    ADJDE = mean(ADJDE)
  )
```

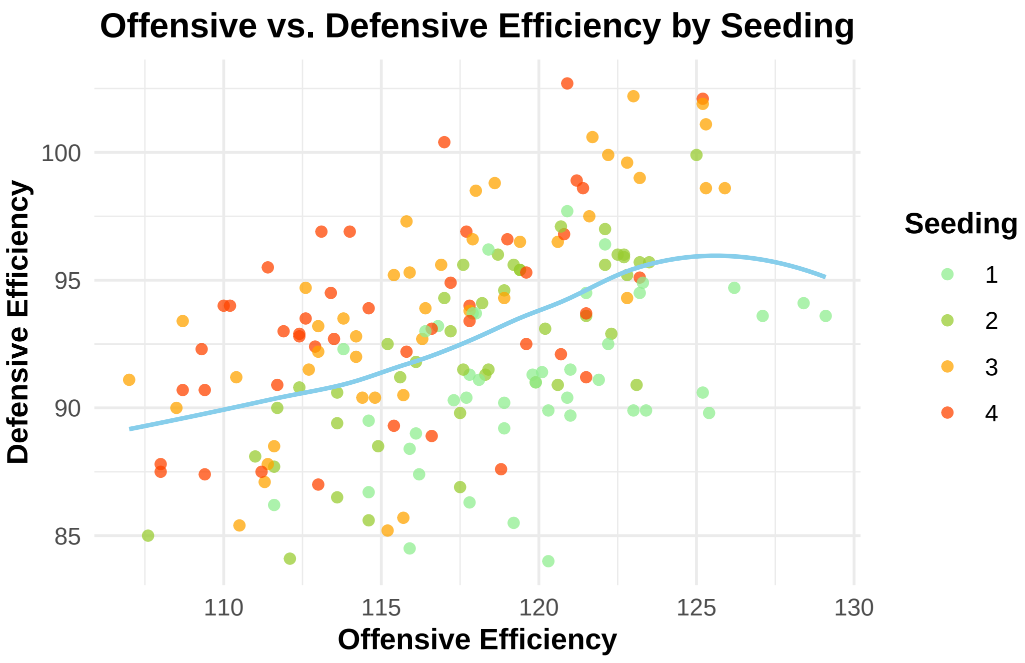
```
`summarise()` has grouped output by 'TEAM', 'YEAR'. You can override using the
`.groups` argument.
```

```r
new_sumsperyear <- rep(NA,17) # New check
for(i in seq_along(new_sumsperyear)){
  new_sumsperyear[i] <- list(table(filter(new_seed_game_production, SEED ==
                                      ifelse(i==17,-1,i))$YEAR))
}
```

```r
top4 <- new_seed_game_production %>%
  filter(SEED <= 4 & SEED > 0)
bottom4 <- new_seed_game_production %>%
  filter(SEED > 12)
best_and_worst <- bind_rows(top4, bottom4)
# Using best group and worst group since the difference between the one seed and the
# 16th seed is smaller than most people think, and since showing all data points is
# too visually crowding, we compare the "worst" teams to the "best" teams
```

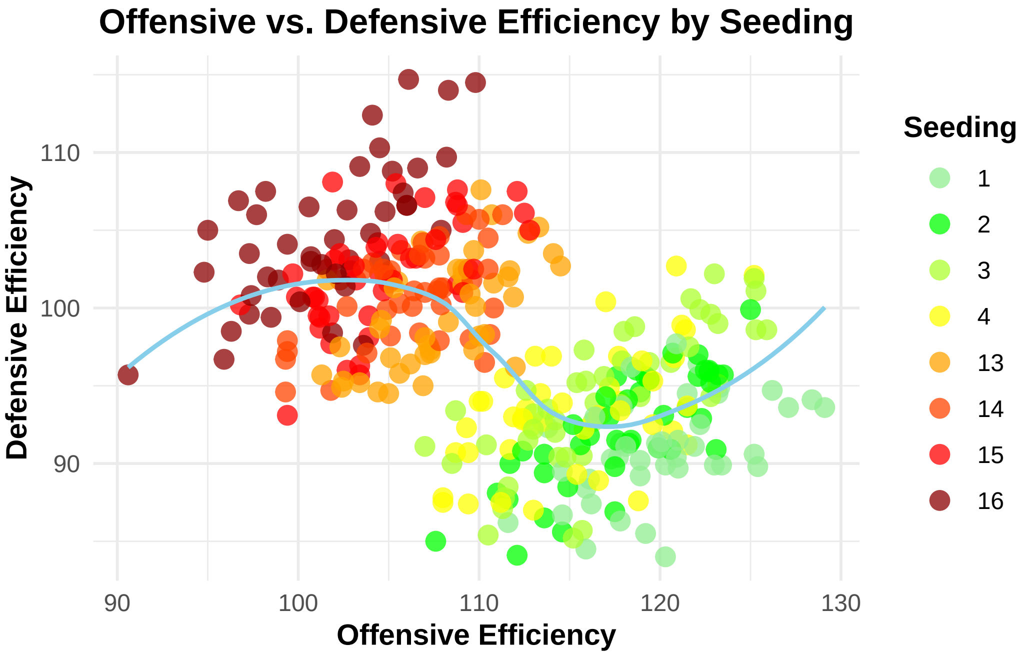
```
ggplot(
  data = top4,
  mapping = aes(x = ADJOE, y = ADJDE, color = factor(SEED))
) +
  geom_point(alpha = 0.75, size = 1.5) +
  geom_smooth(linewidth = .8, color = "skyblue", se = FALSE) +
  labs(title = "Offensive vs. Defensive Efficiency by Seeding",
       x = "Offensive Efficiency",
       y = "Defensive Efficiency",
       color = "Seeding") +
  scale_color_manual(values = c("1" = "lightgreen", "2" = "yellowgreen", "3" =
                                "orange", "4" = "orangered")) +
  theme_minimal() +
  theme(plot.title = element_text(face = "bold", hjust = 0.5),
        axis.title = element_text(face = "bold"),
        legend.title = element_text(face = "bold", hjust = 0.5))
```

```
`geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



```
ggplot(
  data = best_and_worst,
  mapping = aes(x = ADJOE, y = ADJDE, color = factor(SEED))
) +
  geom_point(alpha = 0.75, size = 3) +
  geom_smooth(linewidth = .8, color = "skyblue", se = FALSE) +
  labs(title = "Offensive vs. Defensive Efficiency by Seeding",
       x = "Offensive Efficiency",
       y = "Defensive Efficiency",
       color = "Seeding") +
  scale_color_manual(values = c("1" = "lightgreen", "2" = "green", "3" = "greenyellow",
                        "4" = "yellow", "13" = "orange", "14" = "orangered", "15" = "red",
                        "16" = "darkred")) +
  theme_minimal() +
  theme(plot.title = element_text(face = "bold", hjust = 0.5),
        axis.title = element_text(face = "bold"),
        legend.title = element_text(face = "bold", hjust = 0.5))
```

```
`geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



```
off_contrib <- c("EFG_O","TOR","ORB","FTR","2P_O","3P_O")
def_contrib <- c("EFG_D","TORD","DRB","FTRD","2P_D","3P_D")
off_corrs <- rep(NA, length(off_contrib))
def_corrs <- rep(NA, length(def_contrib))
```

```r
for(i in seq_along(off_contrib)){
  stat <- off_contrib[i]
  off_corrs[i] <- cor(cbb$ADJOE, cbb[[stat]], use = "complete.obs")
}
off_corrs
```

```
[1]  0.7284253 -0.5961689  0.2554623  0.0873290  0.6467673  0.5590129
```

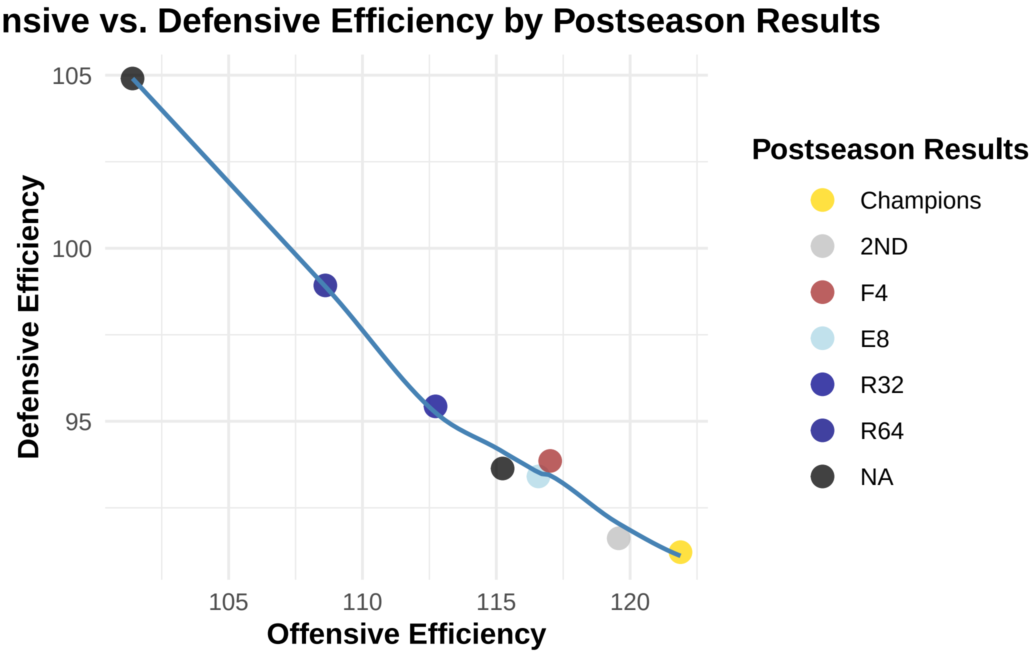
```r
for(i in seq_along(def_contrib)){
  stat <- def_contrib[i]
  def_corrs[i] <- cor(cbb$ADJDE, cbb[[stat]], use = "complete.obs")
}
def_corrs
```

```
[1]  0.7829400 -0.2814940  0.3240309  0.1636061  0.7113540  0.5601543
```

```r
postseasonRanks <- cbb %>%
  group_by(POSTSEASON) %>%
  summarize(
    ADJOE = mean(ADJOE),
    ADJDE = mean(ADJDE),
  ) %>%
  mutate(POSTSEASON =
  factor(POSTSEASON, levels = c("Champions", "2ND", "F4", "E8", "S16", "R32", "R64"))
  )

ggplot(
  data = postseasonRanks,
  mapping = aes(x = ADJOE, y = ADJDE, color = POSTSEASON)
) +
  geom_point(alpha = 0.75, size = 3.5) +
  geom_smooth(linewidth = .8, color = "steelblue", se = FALSE) +
  labs(title = "Offensive vs. Defensive Efficiency by Postseason Results",
       x = "Offensive Efficiency",
       y = "Defensive Efficiency",
       color = "Postseason Results") +
  scale_color_manual(values = c("Champions" = "gold", "2ND" = "gray", "F4" ="brown",
                                "E8" = "lightblue", "R16" = "blue","R32" = "darkblue",
                                "R64"= "navy"), na.value = "black") +
  theme_minimal() +
  theme(plot.title = element_text(face = "bold", hjust = 0.5),
        axis.title = element_text(face = "bold"),
        legend.title = element_text(face = "bold", hjust = 0.5))
```

```
`geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```


nsive vs. Defensive Efficiency by Postseason Results
Defensive Efficiency
Offensive Efficiency
Postseason Results
Champions
2ND
F4
E8
R32
R64
NA