NBA Homegrown Project

William Rice

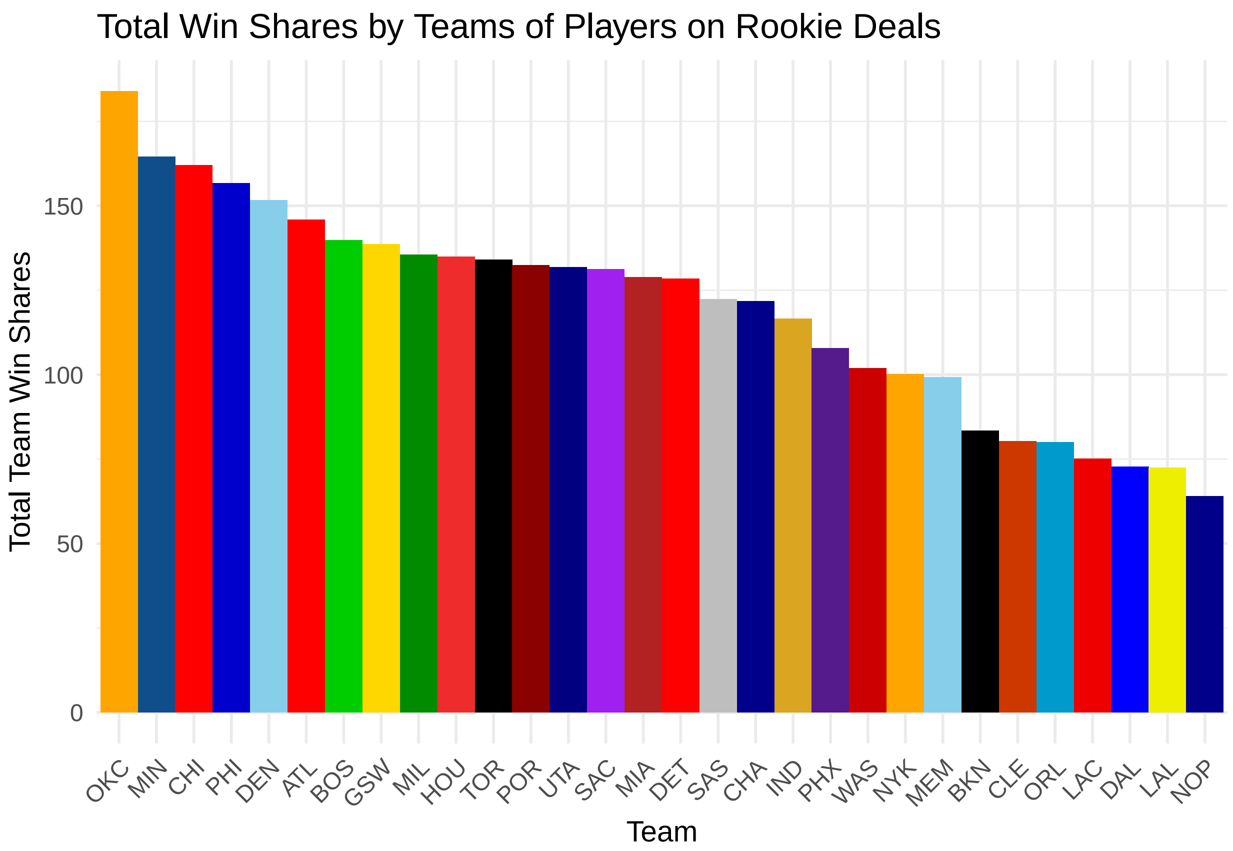
2023-09-18

```r
library(dplyr)
```

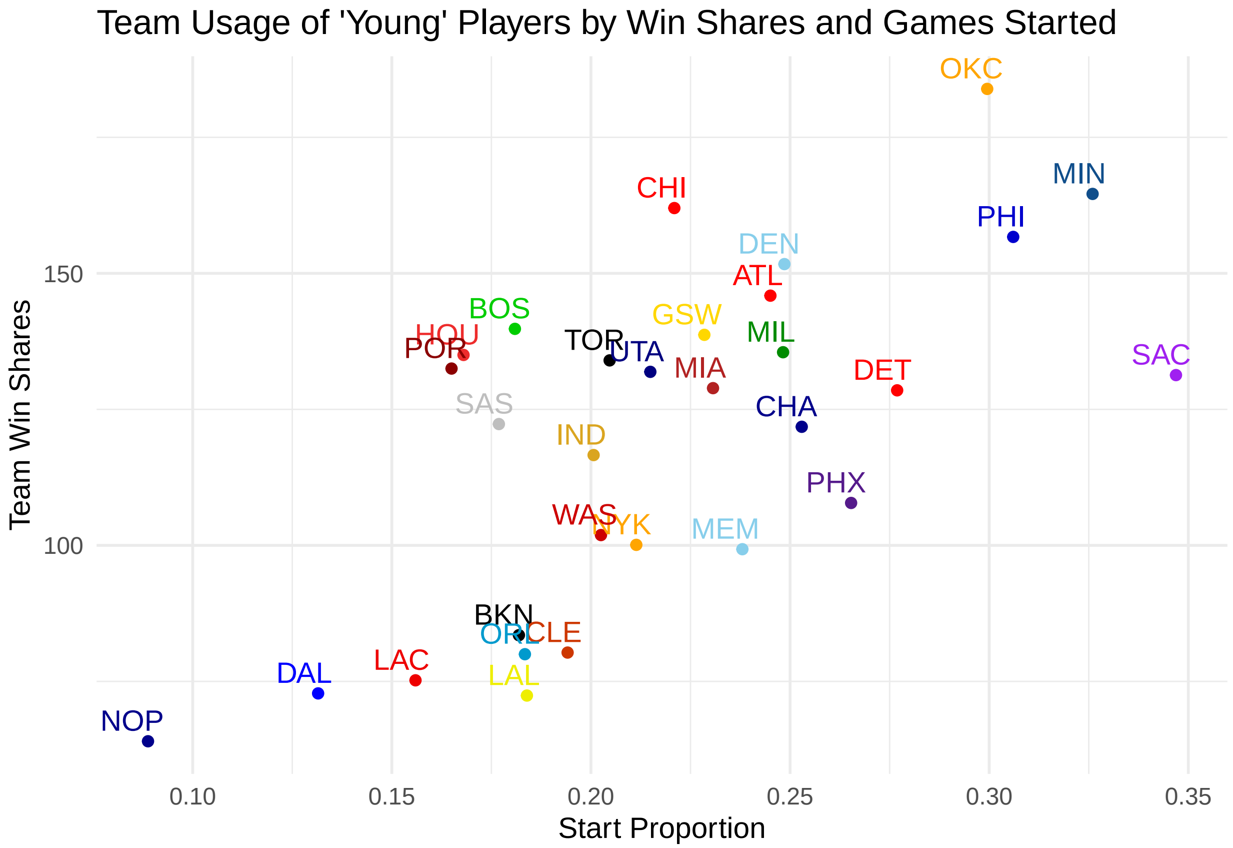
```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(ggplot2)
library(readr)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v stringr   1.5.0
## v lubridate 1.9.2     v tibble    3.2.1
## v purrr     1.0.1     v tidyr     1.3.0

## -- 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 become errors
```

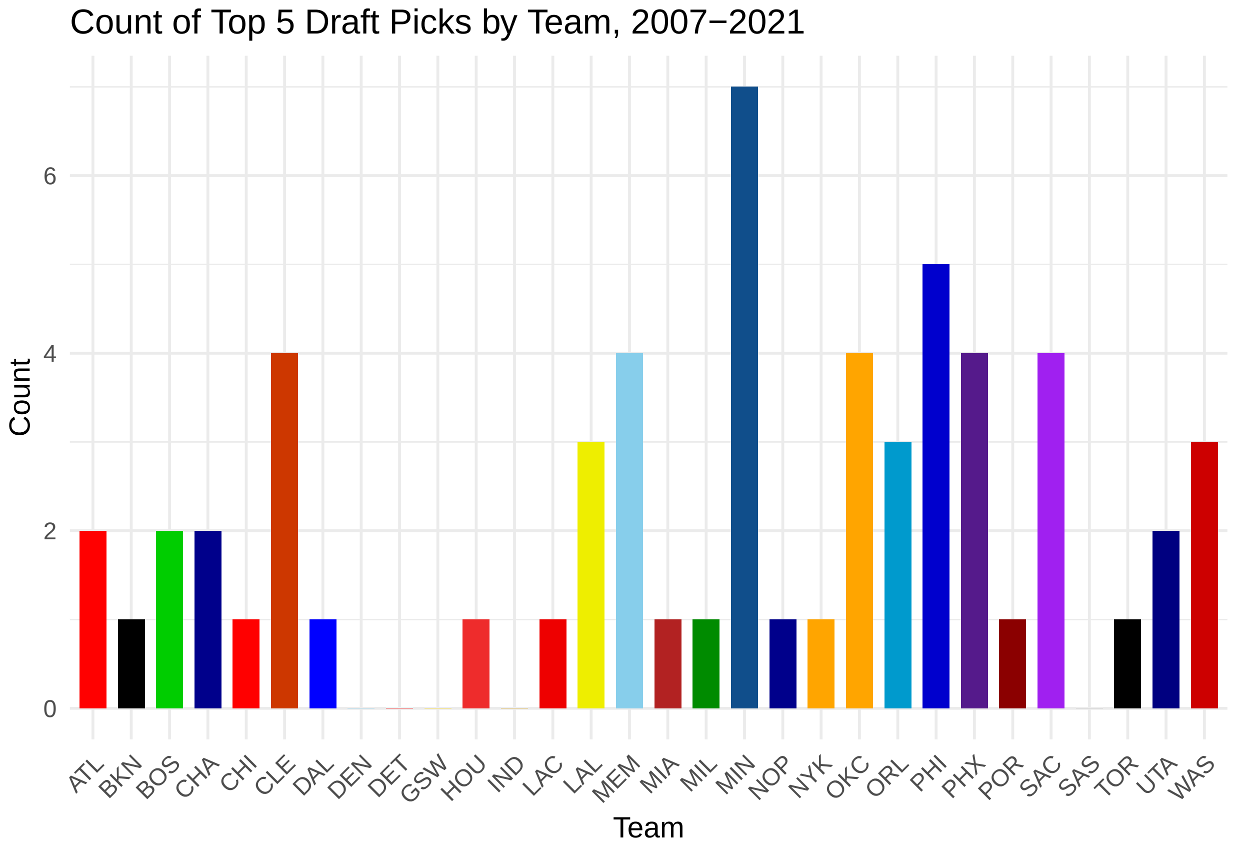

Total Win Shares by Teams of Players on Rookie Deals
Total Team Win Shares
0
50
100
150
Team
OKC
MIN
CHI
PHI
DEN
ATL
BOS
GSW
MIL
HOU
TOR
POR
UTA
SAC
MIA
DET
SAS
CHA
IND
PHX
WAS
NYK
MEM
BKN
CLE
ORL
LAC
DAL
LAL
NOP

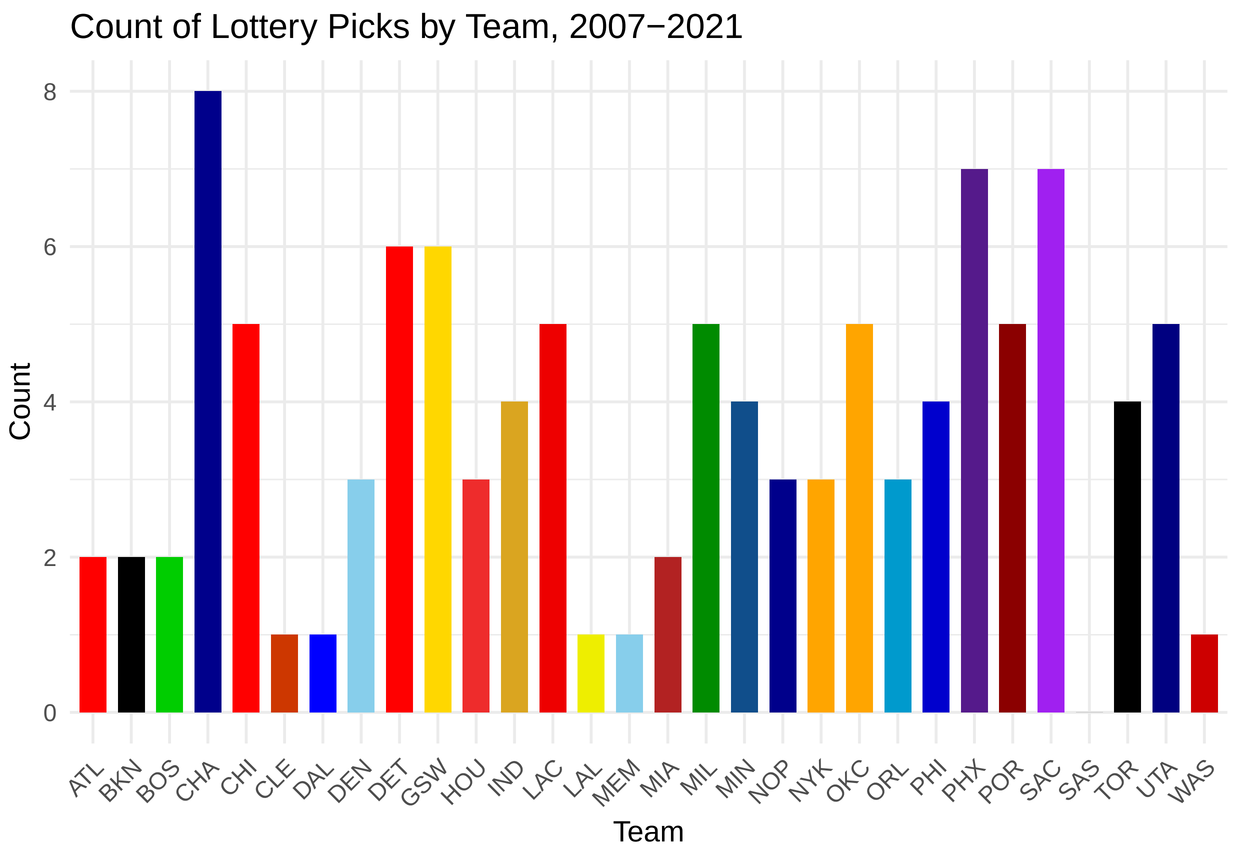
Including Plots



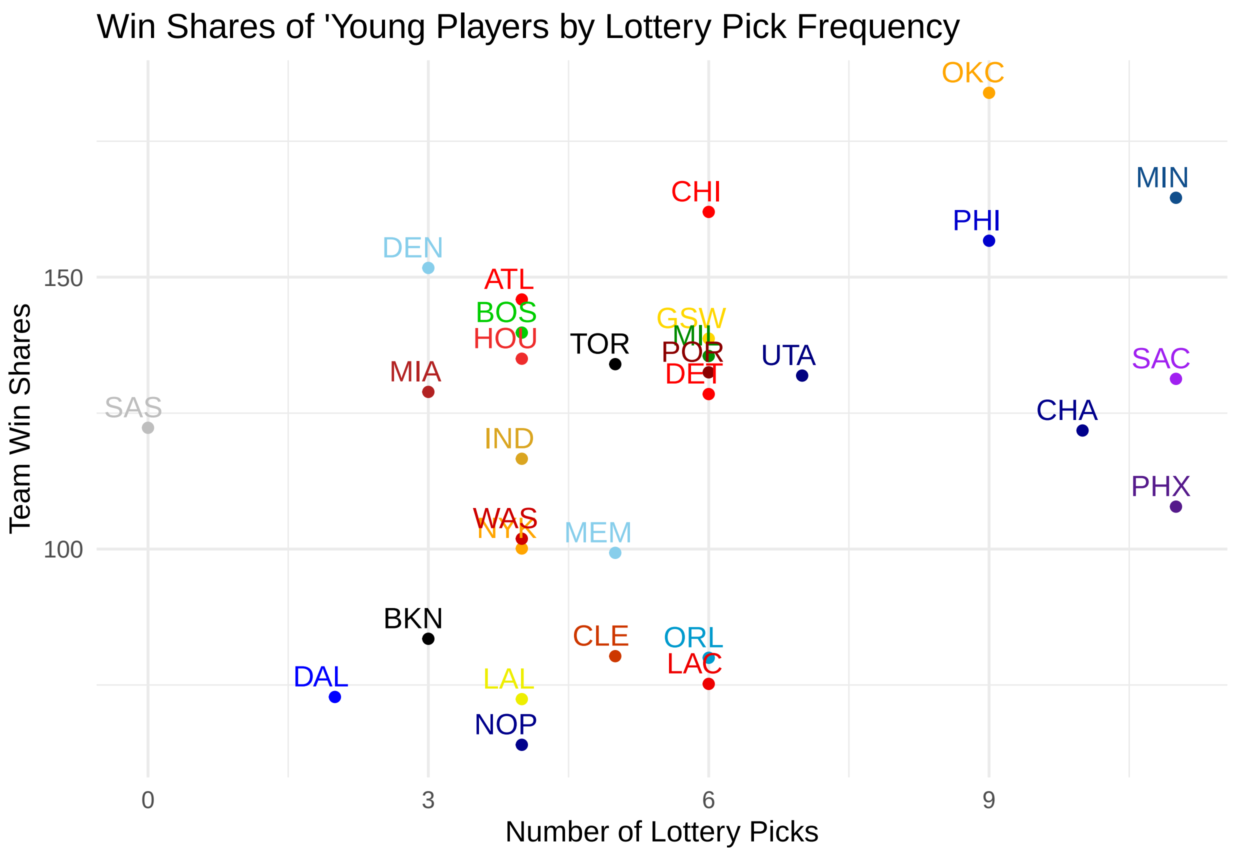
Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

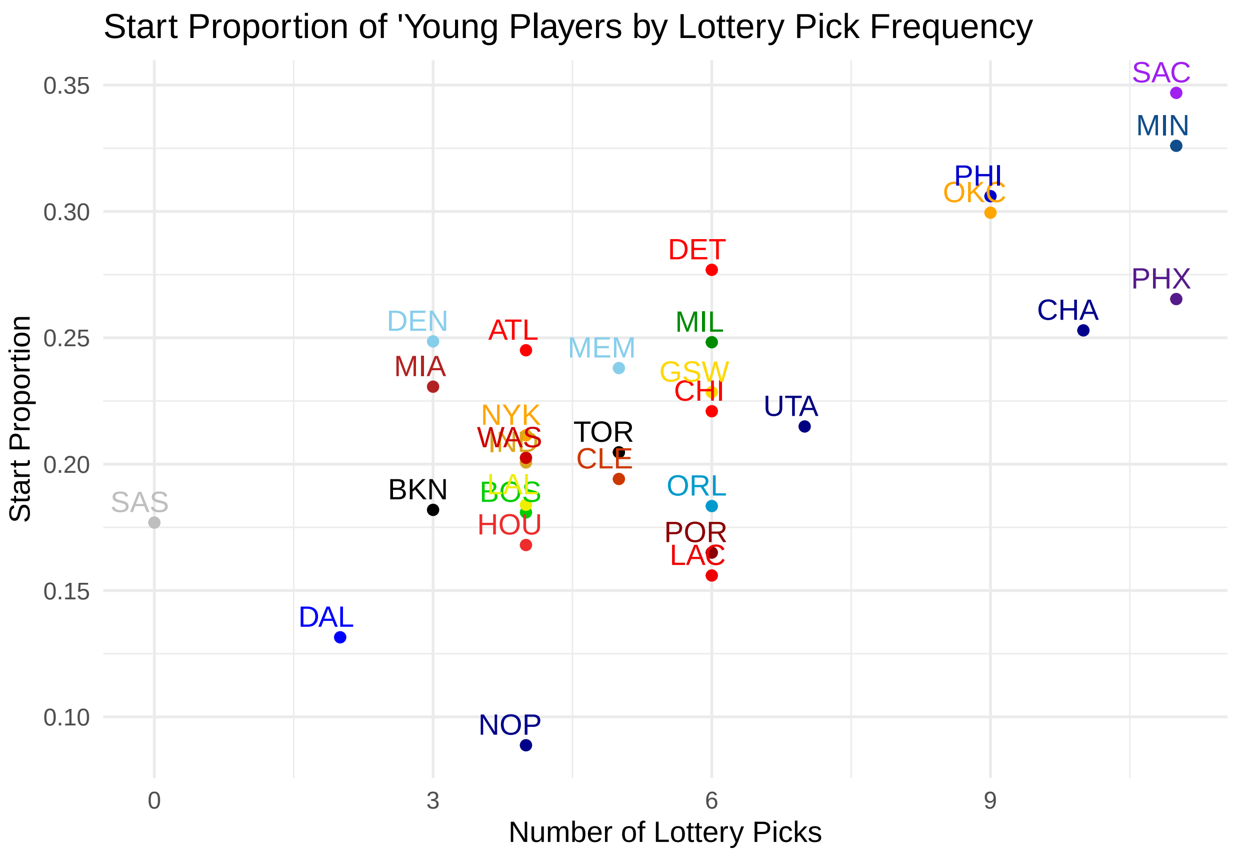
```
## ‘summarise()‘ has grouped output by ’team’. You can override using the
## ‘.groups‘ argument.
## ‘summarise()‘ has grouped output by ’team’. You can override using the
## ‘.groups‘ argument.
```


Count of Top 5 Draft Picks by Team, 2007−2021
Count
0
2
4
6
Team
ATL
BKN
BOS
CHA
CHI
CLE
DAL
DEN
DET
GSW
HOU
IND
LAC
LAL
MEM
MIA
MIL
MIN
NOP
NYK
OKC
ORL
PHI
PHX
POR
SAC
SAS
TOR
UTA
WAS

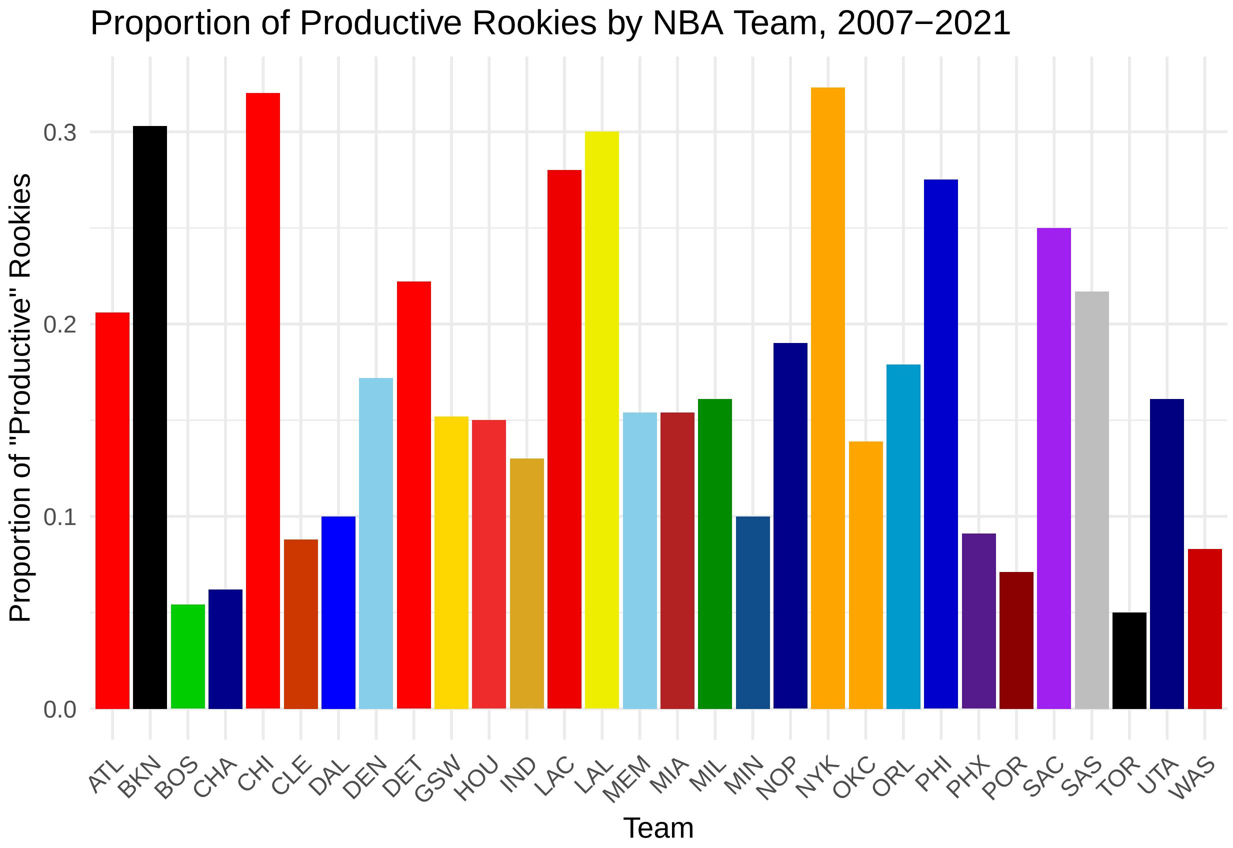


```
## ‘summarise()‘ has grouped output by ’team’. You can override using the
## ‘.groups‘ argument.
```


Win Shares of 'Young Players by Lottery Pick Frequency
Team Win Shares
Number of Lottery Picks
150
100
0
3
6
9
OKC
MIN
CHI
PHI
DEN
ATL
BOS
HOU
TOR
GSW
MIL
POR
DET
UTA
SAC
MIA
SAS
CHA
IND
PHX
WAS
NYK
MEM
BKN
CLE
ORL
LAC
DAL
LAL
NOP



```r
project_dataset <- project_dataset %>%
  mutate(ppg = points/games) %>%
  mutate(apg = ast/games) %>%
  mutate(rpg = tot_reb/games) %>%
  mutate(mpg = mins/games) %>%
  mutate_at(vars(apg, mpg, ppg, rpg), ~ round(., 1))
view(project_dataset)


statistics_summary <- project_dataset %>%
  group_by(team) %>%
  summarize(
    average_ppg = mean(ppg),
    average_apg = mean(apg),
    average_rpg = mean(rpg),
    average_mpg = mean(mpg)
  ) %>%
  mutate_at(vars(average_ppg, average_apg, average_rpg, average_mpg), ~ round(., 1))
view(statistics_summary)


filtered_statistics_dataset <- project_dataset %>%
  filter(mpg > mean(mpg)) %>%
  filter(WS > mean(WS)) %>%
  filter(efg > mean(efg)) %>%
  filter(BPM > mean(BPM)) %>%
  filter(PER > mean(PER)) %>%
  filter(VORP > mean(VORP))
view(filtered_statistics_dataset)


rookies_filtered <- draft_pick %>%
  mutate(ppg = points/games) %>%
  mutate(apg = ast/games) %>%
  mutate(rpg = tot_reb/games) %>%
  mutate(mpg = mins/games) %>%
  mutate_at(vars(apg, mpg, ppg, rpg), ~ round(., 1)) %>%
  mutate(total_possible_games = ifelse(season == 2011 | season == 2020 | season == 2021, ifelse(season =
  filter(games >= 0.70*total_possible_games & mpg > 12) %>%
  filter(WS > 0) %>%
  filter(ppg > 8) %>%
  group_by(team) %>%
  summarize(count = n())

view(rookies_filtered)

grouped_statistics_means <- draft_pick %>%
  mutate(ppg = points/games) %>%
  mutate(apg = ast/games) %>%
  mutate(rpg = tot_reb/games) %>%
  mutate(mpg = mins/games) %>%
  mutate_at(vars(apg, mpg, ppg, rpg), ~ round(., 1)) %>%
  group_by(pick_status) %>%
  summarize(
    average_ppg = mean(ppg),
    average_apg = mean(apg),
    average_rpg = mean(rpg),
    average_mpg = mean(mpg),
    average_WS = mean(WS),
    average_PER = mean(PER)
  ) %>%
  mutate_at(vars(average_ppg, average_apg, average_rpg, average_mpg, average_WS), ~ round(., 1))
view(grouped_statistics_means)

rookie_successes <- left_join(draft_pick, grouped_statistics_means, by = "pick_status") %>%
  mutate(ppg = points/games) %>%
  mutate(apg = ast/games) %>%
  mutate(rpg = tot_reb/games) %>%
  mutate(mpg = mins/games) %>%
  mutate_at(vars(apg, mpg, ppg, rpg), ~ round(., 1)) %>%
  filter(WS > average_WS) %>%
  filter(mpg > average_mpg) %>%
  filter(PER > average_PER) %>%
  filter(ppg > average_ppg + 2| apg > average_apg + 1 | rpg > average_rpg + 1.5) %>%
  group_by(team) %>%
  summarize(count = n())
view(rookie_successes)

team_total_rookies <- draft_pick %>%
  group_by(team) %>%
  summarize(count = n())
view(team_total_rookies)

success_proportion <- left_join(rookie_successes, team_total_rookies, by = "team") %>%
  mutate(rookie_hits = count.x/count.y) %>%
  mutate_at(vars(rookie_hits), ~ round(., 3)) %>%
  select(-count.x, -count.y)
view(success_proportion)


ggplot(success_proportion, aes(x = team, y = rookie_hits, fill = team)) +
  geom_bar(stat = 'identity', width = 0.9) +
  labs(x = 'Team', y = 'Proportion of "Productive" Rookies', title = "Proportion of Productive Rookies 
  scale_fill_manual(values = team_colors) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  theme(legend.position = "none")
```

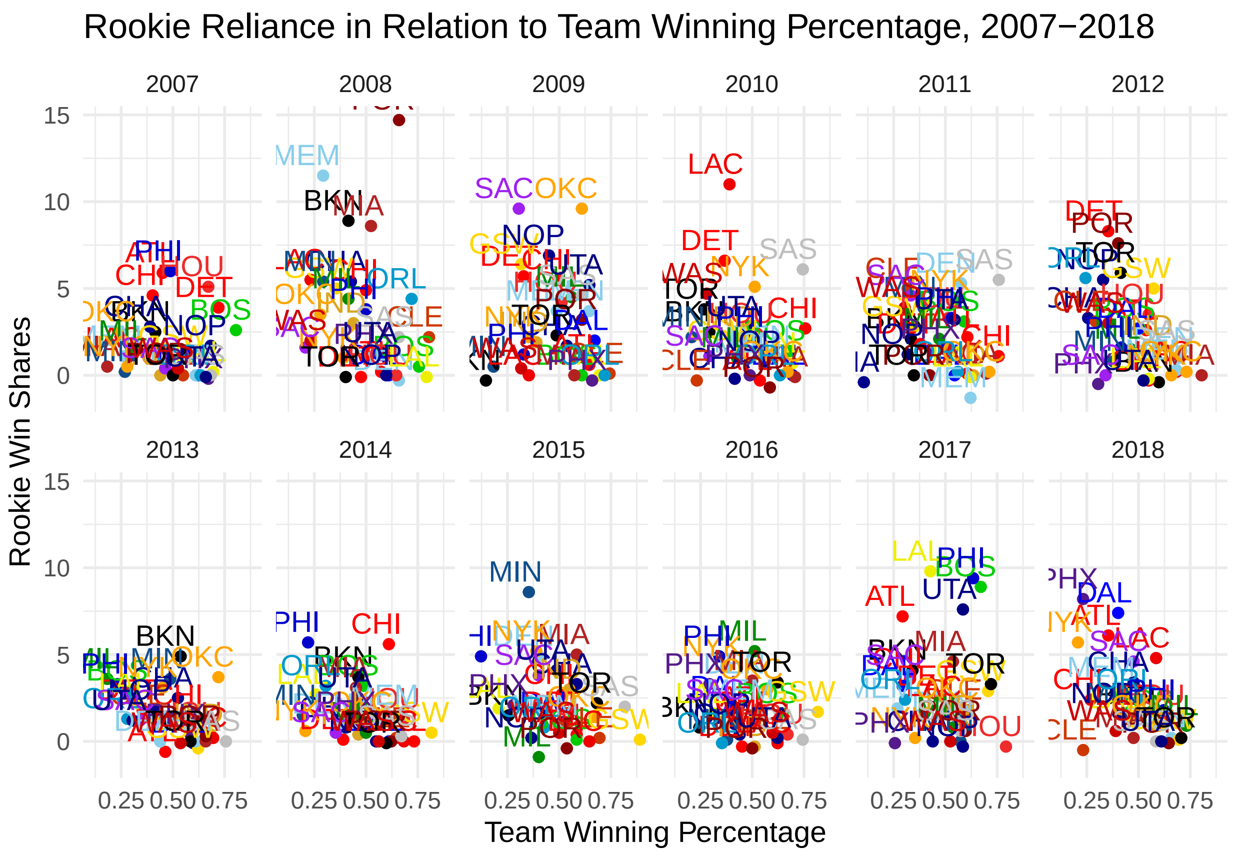


```
team_data <- team_data %>%
  mutate(win_percentage = W/games) %>%
  mutate_at(vars(win_percentage), ~ round(., 3))

team_and_player_data <- left_join(draft_pick, team_data, by = c("nbateamid", "season")) %>%
  select(-games.y, -team.y) %>%
  rename(team_off_rtg = off_rtg,
  team_def_rtg = def_rtg,
  team_net_rtg = net_rtg,
  team_win_percentage = win_percentage,
  games = games.x,
  team = team.x)
view(team_and_player_data)

rookie_team_data <- team_and_player_data %>%
  group_by(team, season) %>%
  summarize(rookie_win_shares = sum(WS))
```

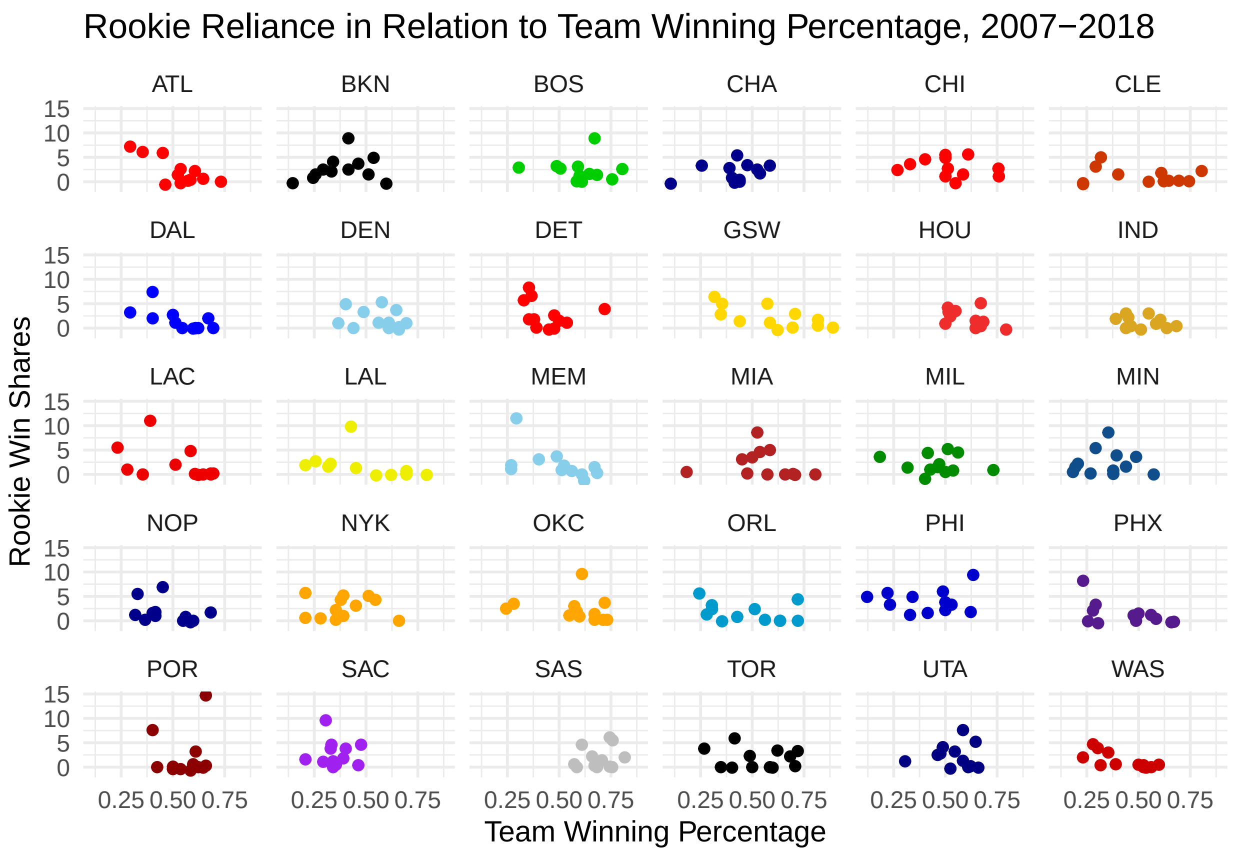
```
## ‘summarise()‘ has grouped output by ’team’. You can override using the
## ‘.groups‘ argument.
```

```
view(rookie_team_data)

rookie_team_data <- left_join(team_data, rookie_team_data, by = c('team', 'season')) %>%
  mutate_all(~ifelse(is.na(.), 0, .))


rookie_team_data <- rookie_team_data %>%
  filter(season <= 2018)

view(rookie_team_data)

ggplot(rookie_team_data, aes(x = win_percentage, y = rookie_win_shares, color = team, label = team, grou
  geom_jitter() +
  geom_smooth(method = "lm", se = FALSE, color = "black") +
  geom_text(vjust = -0.50, hjust = .75) +
  labs(x = "Team Winning Percentage", y = "Rookie Win Shares", title = "Rookie Reliance in Relation to T
  theme_minimal() +
  scale_color_manual(values = team_colors) +
  facet_wrap(~season, ncol = 6) +
  theme(legend.position = "none")
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```



```
ggplot(rookie_team_data, aes(x = win_percentage, y = rookie_win_shares, group = team, color = team)) +
  geom_jitter() +
  geom_text(vjust = -0.50, hjust = .75, label = "") +
  labs(x = "Team Winning Percentage", y = "Rookie Win Shares", title = "Rookie Reliance in Relation to
  theme_minimal() +
  facet_wrap(~team, ncol = 6) +
  scale_color_manual(values = team_colors) +
  theme(legend.position = "none")
```



```
rookie_regression <- lm(win_percentage ~ rookie_win_shares, data = rookie_team_data)
cor(rookie_team_data$rookie_win_shares, rookie_team_data$win_percentage)
```

```
## [1] -0.2318268
```

```
print(rookie_regression)
```

```
##
## Call:
## lm(formula = win_percentage ~ rookie_win_shares, data = rookie_team_data)
##
## Coefficients:
##       (Intercept)  rookie_win_shares
##           0.53113           -0.01488
```

```
improvement_dataset <- project_dataset %>%
  filter(years_in_league == 1 | years_in_league == 4) %>%
  group_by(nbapersonid) %>%
  filter(n_distinct(years_in_league) == 2) %>%
  ungroup()
view(improvement_dataset)

improvement_dataset[299, "player"] <- 'Luka Doncic'
```

```
difference_dataset <- improvement_dataset %>%
  select(nbapersonid, player, WS, mpg, ppg, efg, PER, apg, rpg, VORP, BPM, usg, ftp, fgp3) %>%
  group_by(nbapersonid, player) %>%
  summarize(
    WS_diff = last(WS) - first(WS),
    mpg_diff = last(mpg) - first(mpg),
    ppg_diff = last(ppg) - first(ppg),
    efg_diff = last(efg) - first(efg),
    PER_diff = last(PER) - first(PER),
    apg_diff = last(apg) - first(apg),
    VORP_diff = last(VORP) - first(VORP),
    BPM_diff = last(BPM) - first(BPM),
    usg_diff = last(usg) - first(usg),
    ftp_diff = last(ftp) - first(ftp),
    fgp3_diff = last(fgp3) - first(fgp3),
    rpg_diff = last(rpg) - first(rpg)
  ) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'nbapersonid'. You can override using the
## `.groups` argument.
```

```
view(difference_dataset)


draft_pick[801, "player"] <- 'Luka Doncic'


draft_pick_filtered <- draft_pick %>%
  semi_join(difference_dataset, by = "nbapersonid")

difference_dataset <- left_join(difference_dataset, draft_pick_filtered, by = "nbapersonid")

difference_dataset <- difference_dataset %>%
  select(nbapersonid, player.x, WS_diff, rpg_diff, fgp3_diff, ftp_diff, usg_diff, BPM_diff, VORP_diff, a
  rename(player = player.x)


team_improvement <- difference_dataset %>%
  group_by(team) %>%
  summarize(
    count = n(),
    average_win_share_improvement = mean(WS_diff),
    minutes_improvement = mean(mpg_diff),
    PER_improvement = mean(PER_diff),
    VORP_improvement = mean(VORP_diff),
    points_improvement = mean(ppg_diff)
    ) %>%
  mutate_at(vars(average_win_share_improvement, PER_improvement, minutes_improvement, VORP_improvement,

view(team_improvement)
```

```
moderate_improvement_teams <- team_improvement %>%
  filter(minutes_improvement >= 6 & points_improvement >= 4 & count >= 4)
view(moderate_improvement_teams)
#On very small minimums, half the league saw marginal success from their drafted rookies from year 1 to

high_improvement_teams <- team_improvement %>%
  filter(minutes_improvement >= 8 & points_improvement >= 6 & count >= 6 & PER_improvement > 0)
view(high_improvement_teams)
#On more filters and higher thresholds, 7 teams saw high improvement from their drafted players#

highest_improving_teams <- team_improvement %>%
  filter(minutes_improvement >= 10 & points_improvement >= 7.5 & count >= 8 & PER_improvement > 2)
view(highest_improving_teams)
#On the filters that got towards the most extreme, Toronto and Boston saw the most improvement in quali
```