Derik Queen Analysis

Code ▾

Hide

```
library(reticulate)
library(ggplot2)
library(tidyr)
library(dplyr)
library(stringr)
```

Hide

```
import pandas as pd
import numpy as np
import requests
import json
import seaborn as sns
import matplotlib.pyplot as plt
```

Hide

```
shotsurl = 'https://api.cbbanalytics.com/api/gs/pbp-shots?competitionId=38409&playerId=2313627'
statsurl = 'https://api.cbbanalytics.com/api/gs/player-game-stats?competitionId=38409&playerId=231
3627&pass=false'
```

Hide

```
response1 = requests.get(shotsurl)
data1 = response1.json()
shotsdf = pd.DataFrame(data1)
```

Hide

```
response2 = requests.get(statsurl)
data2 = response2.json()
statsdf = pd.DataFrame(data2)
```

Hide

```
# pctileurl = 'https://api.cbbanalytics.com/api/gs/pctiles-player-agg-pbp-stats/competition/38409/
division/1/'
# response3 = requests.get(pctileurl)
# data3 = response3.json()
# pctiledf = pd.DataFrame(data3)
```

Hide

```
head(py$shotsdf)
```

	_id	updated	competitionId	hom…	awa…	homeConferenceId	awayConferenc
	<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<d
1	2538717-223	2025-02-17T00:05:03Z	38409	103871	103734	73	
2	2546544-28	2025-01-13T21:38:20Z	38409	103871	104360	73	

	_id	updated	competitionId	hom…	awa…	homeConferenceId	awayConferenc
	<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<d
3	2546519-91	2024-12-08T18:18:26Z	38409	103871	104041	73	
4	2546507-24	2024-11-27T23:33:33Z	38409	103871	103382	73	
5	2546564-348	2025-02-07T19:59:52Z	38409	104041	103871	73	
6	2546477-278	2024-11-12T02:00:39Z	38409	103871	103615	73	

6 rows | 1-8 of 68 columns

Hide

```
head(py$statsdf)
```

	_id	updated	leagueId	competitionId	gen…	gameId	period	tea…
	<chr>	<chr>	<dbl>	<dbl>	<chr>	<dbl>	<chr>	<dbl>
1	2538022-2313627-game	2025-01-05T23:04:40Z	456	38409	MALE	2538022	game	10387
2	2539293-2313627-game	2025-01-30T02:10:07Z	456	38409	MALE	2539293	game	10387
3	2539401-2313627-game	2025-03-06T01:25:28Z	456	38409	MALE	2539401	game	10387
4	2539534-2313627-game	2025-01-26T19:07:00Z	456	38409	MALE	2539534	game	10387
5	2540180-2313627-game	2025-02-13T15:13:41Z	456	38409	MALE	2540180	game	10387
6	2542667-2313627-game	2025-03-03T20:08:33Z	456	38409	MALE	2542667	game	10387

6 rows | 1-9 of 128 columns

Hide

```
colnames(py$statsdf)
```

```
  [1] "_id"              "updated"          "leagueId"
  [4] "competitionId"    "gender"           "gameId"
  [7] "period"           "teamId"           "teamMarket"
 [10] "teamIdAgst"       "teamMarketAgst"   "conferenceId"
 [13] "conferenceIdAgst" "divisionId"       "divisionIdAgst"
 [16] "playerId"         "fullName"         "jerseyNum"
 [19] "hasImage"         "classYr"          "isQualified"
 [22] "position"         "positionGame"     "isLiveStream"
 [25] "gameDate"         "gameTime"         "isNeutral"
 [28] "isHome"           "isWin"            "isExhib"
 [31] "inDivision"       "isPostSeason"     "inConferenceReg"
 [34] "inConferenceAll"  "isStarter"        "tournamentId"
 [37] "netRankAgst"      "quadAgst"         "teamGameRecency"
 [40] "overallWins"      "overallLosses"    "confWins"
 [43] "confLosses"       "gp"               "gs"
 [46] "mins"             "poss"             "ptsScored"
 [49] "scp"              "drb"              "pitp"
 [52] "fbpts"            "orb"              "fgm2"
 [55] "fga2"             "tov"              "fgm3"
 [58] "stl"              "fga3"             "fga"
 [61] "fgm"              "blkd"             "blk"
 [64] "ast"              "fta"              "ftm"
 [67] "tf"               "pf"               "opf"
 [70] "dpf"              "pfd"              "reb"
 [73] "plusMinus"        "minsTm"           "ptsScoredTm"
 [76] "possTm"           "fgmTm"            "fgaTm"
 [79] "ftaTm"            "orbTm"            "drbTm"
 [82] "rebTm"            "tovTm"            "ptsAgst"
 [85] "dPossTm"          "drbAgst"          "orbAgst"
 [88] "rebAgst"          "fgaAgst"          "fga3Agst"
 [91] "ftaAgst"          "fgmAgst"          "tovAgst"
 [94] "minsTmBy5"        "tsa"              "fgPct"
 [97] "fg2Pct"           "fg3Pct"           "efgPct"
[100] "tsPct"            "fga3Rate"         "ftaRate"
[103] "ftmRate"          "ftPct"            "blkPct"
[106] "astPct"           "astRatio"         "astTov"
[109] "orbPct"           "drbPct"           "rebPct"
[112] "stlPct"           "tovPct"           "stlTov"
[115] "usagePct"         "pfEff"            "stlPerPf"
[118] "blkPerPf"         "pfdP100"          "scpPctPts"
[121] "fbptsPctPts"      "pitpPctPts"       "ftmPctPts"
[124] "fgm2PctPts"       "fgm3PctPts"       "vps"
[127] "hkmPct"           "astUsage"
```

Hide

```
colnames(py$shotsdf)
```

```
 [1] "_id"              "updated"          "competitionId"    
 [4] "homeId"           "awayId"           "homeConferenceId" 
 [7] "awayConferenceId" "gameId"           "isLiveStream"     
[10] "isNeutral"        "inDivision"       "gameDate"         
[13] "tournamentId"     "conferenceId"     "didHomeWin"       
[16] "isExhib"          "isExhibAgst"      "periodNumber"     
[19] "periodType"       "periodIdx"        "period"           
[22] "teamIdAgst"       "teamId"           "playerId"         
[25] "actionNumber"     "minsIntoGame"     "secsIntoGame"     
[28] "clock"            "shotClock"        "actionDuration"   
[31] "actionType"       "subType"          "success"          
[34] "homePts"          "awayPts"          "homeScore"        
[37] "awayScore"        "side"             "anGs"             
[40] "paGs"             "xRaw"             "yRaw"             
[43] "x"                "y"                "hexX"             
[46] "hexY"             "shotDist"         "zonesGenius"      
[49] "zones6"           "zones13"          "zones17"          
[52] "dists7"           "xGs"              "yGs"              
[55] "awayLineupId"     "homeLineupId"     "prevActionType"   
[58] "assisterId"       "possTeamId"       "possNum"          
[61] "chncNum"          "chncStartType"    "possStartType"    
[64] "possStart"        "possEnd"          "chncStart"        
[67] "chncEnd"          "shotContext"      
```

Hide

```
mean(py$statsdf$usagePct)
```

```
[1] 0.2642064
```

Hide

```
py$statsdf %>%
  group_by(isWin) %>%
  summarize(avg_pts = mean(ptsScored), avg_assists = mean(ast), avg_rebounds = mean(reb))
```

isWin <lgl>	avg_pts <dbl>	avg_assists <dbl>	avg_rebounds <dbl>
FALSE	16.75	1.25	7.50
TRUE	16.16	2.08	9.48

2 rows

Hide

```
py$statsdf %>%
  group_by(quadAgst) %>%
  summarize(avg_pts = mean(ptsScored), avg_assists = mean(ast), avg_rebounds = mean(reb),games = n
())
```

quadAgst <chr>	avg_pts <dbl>	avg_assists <dbl>	avg_rebounds <dbl>	games <int>
quad1	17.66667	1.600000	8.800	15

quadAgst	avg_pts	avg_assists	avg_rebounds	games
<chr>	<dbl>	<dbl>	<dbl>	<int>
quad2	12.42857	1.857143	10.000	7
quad3	22.33333	3.333333	10.000	3
quad4	14.87500	1.875000	8.125	8

4 rows

Hide

```
py$statsdf %>% group_by(quadAgst) %>% summarise(avgPM = mean(plusMinus,na.rm=T))
```

quadAgst	avgPM
<chr>	<dbl>
quad1	6.533333
quad2	10.000000
quad3	13.666667
quad4	24.500000

4 rows

Hide

```
mean(py$statsdf$netRankAgst)
```

```
[1] 109.5455
```

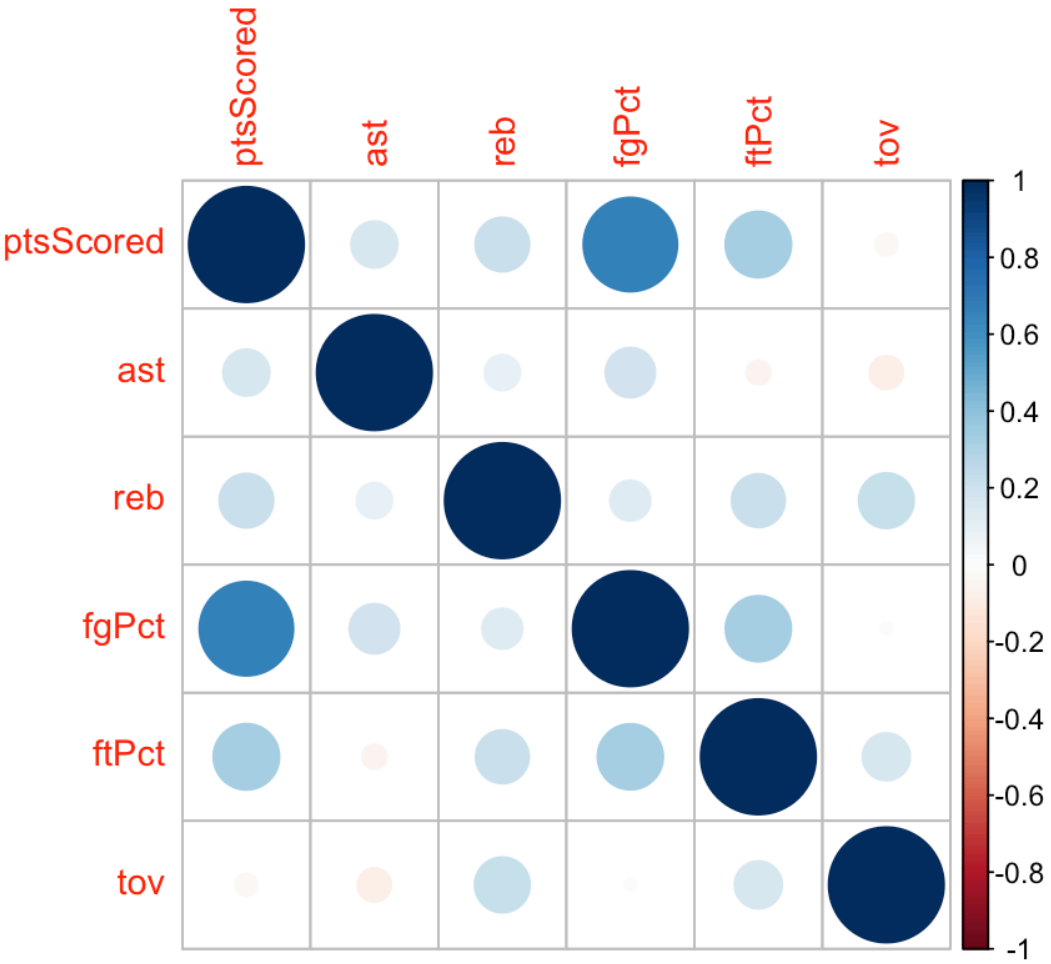
Hide

```
library(corrplot)
```

```
corrplot 0.95 loaded
```

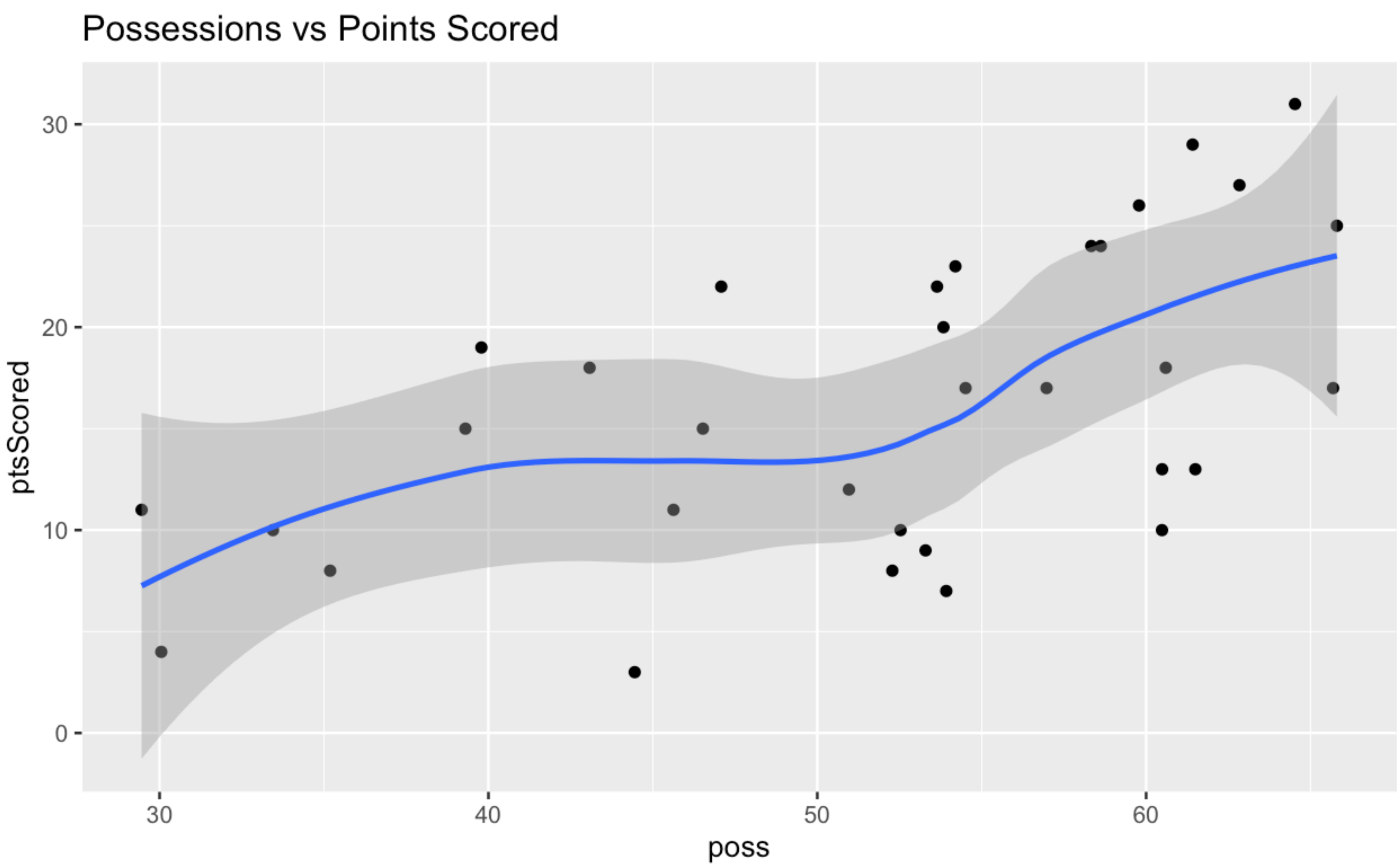
Hide

```
corr_data <- py$statsdf[, c("ptsScored", "ast", "reb", "fgPct", "ftPct", "tov")]
corr_matrix <- cor(corr_data)
corrplot(corr_matrix, method = "circle")
```



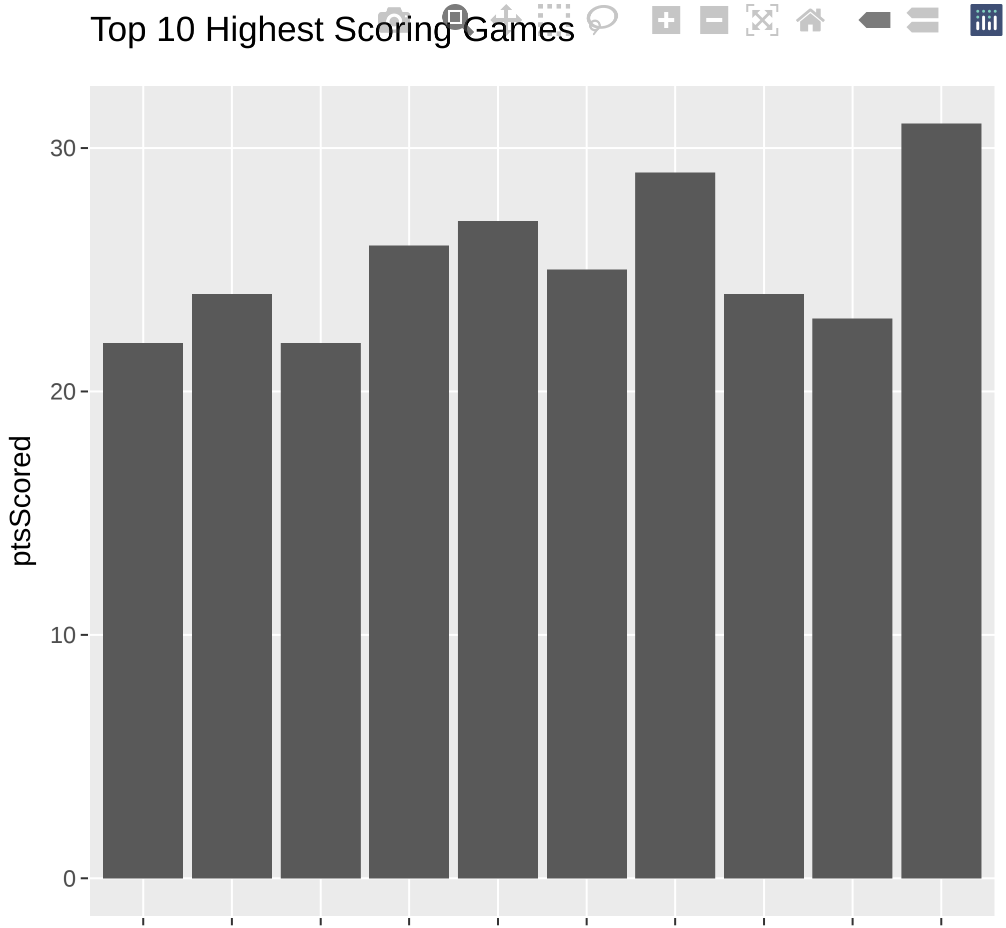
Hide

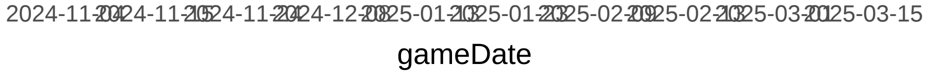
```
ggplot(py$statsdf, aes(x=poss,y=ptsScored)) + geom_point() + geom_smooth() + ggtitle('Possessions
vs Points Scored')
```



Hide

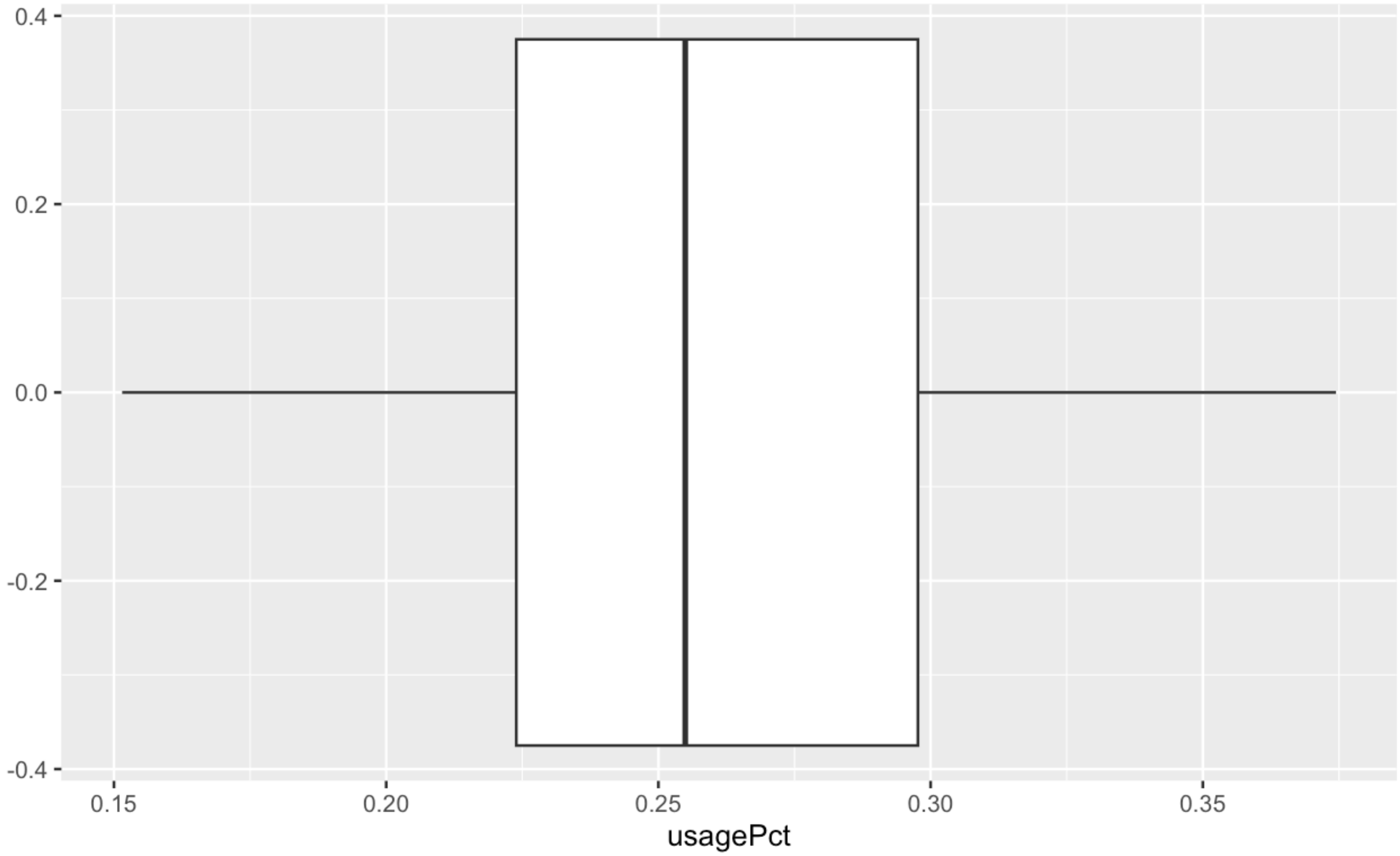
```
ggplotly(ggplot(py$statsdf %>% arrange(desc(ptsScored)) %>% head(10),aes(x=gameDate,y=ptsScored))
+ geom_bar(stat='identity') + ggtitle('Top 10 Highest Scoring Games'))
```





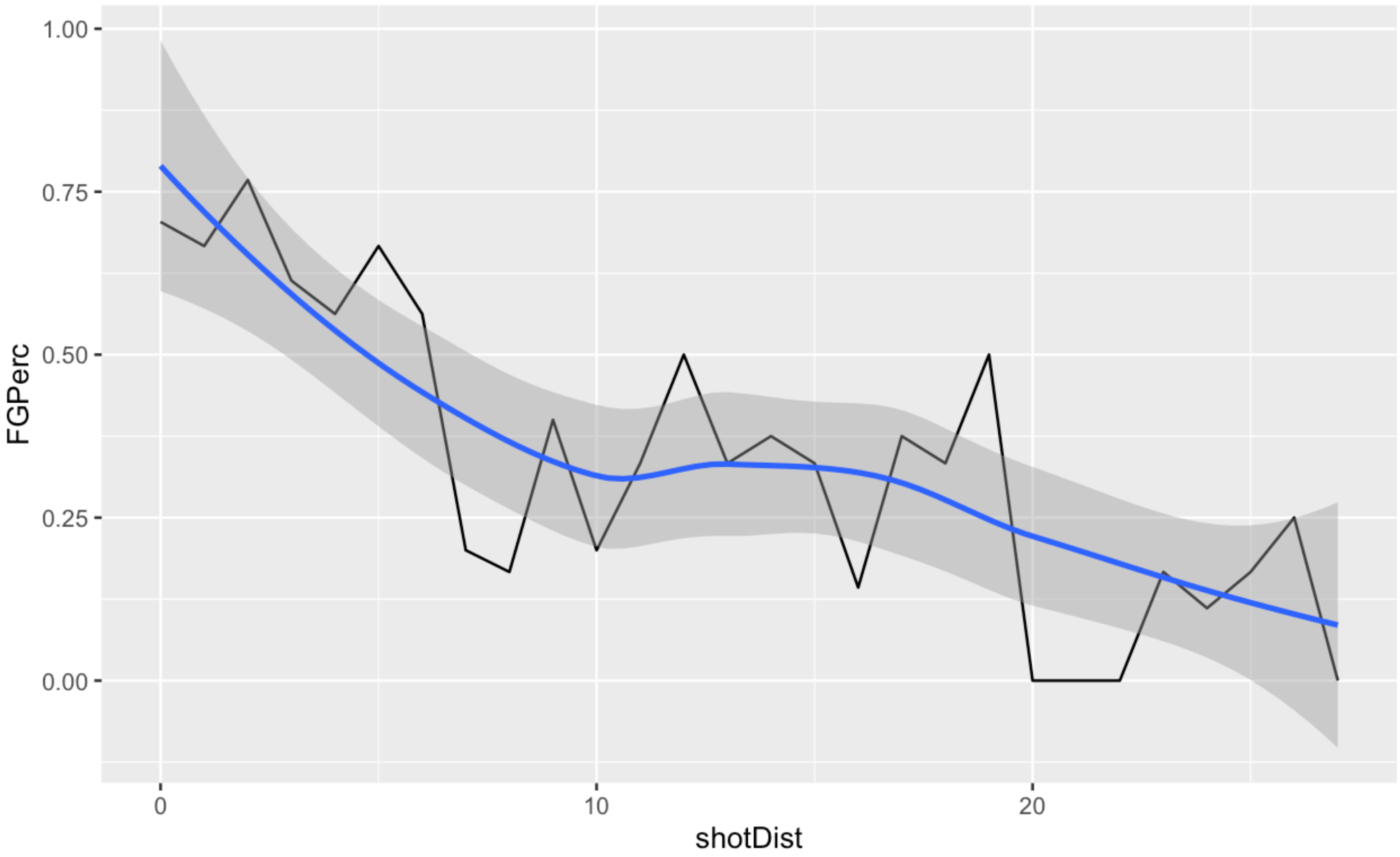
Hide

```
ggplot(py$statsdf, aes(x=usagePct)) + geom_boxplot()
```



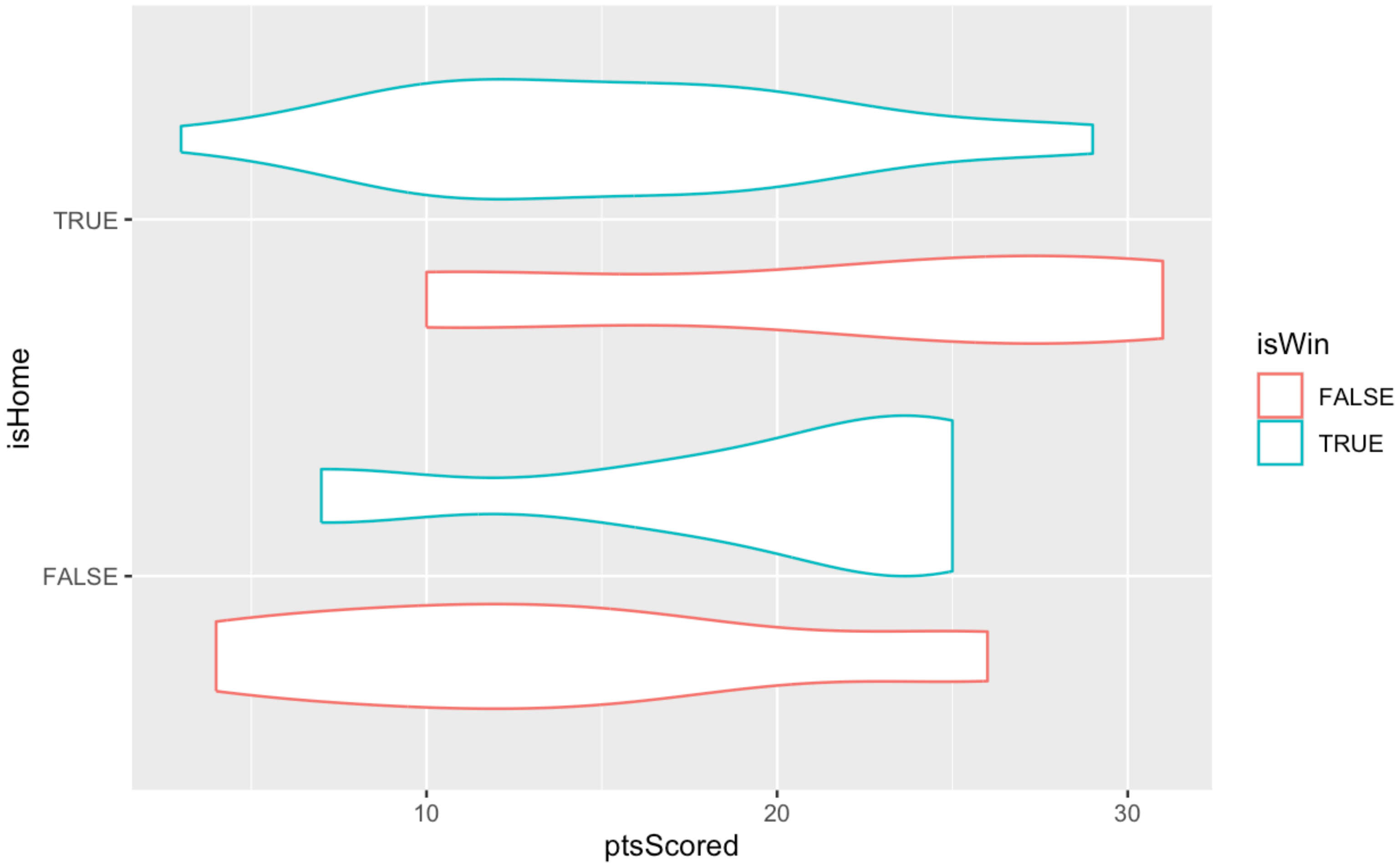
Hide

```
ggplot(py$shotsdf %>% group_by(shotDist = as.integer(shotDist)) %>% summarise(shots =n(),FGPerc =
mean(success)),aes(x=shotDist,y=FGPerc)) + geom_line() + geom_smooth()
```



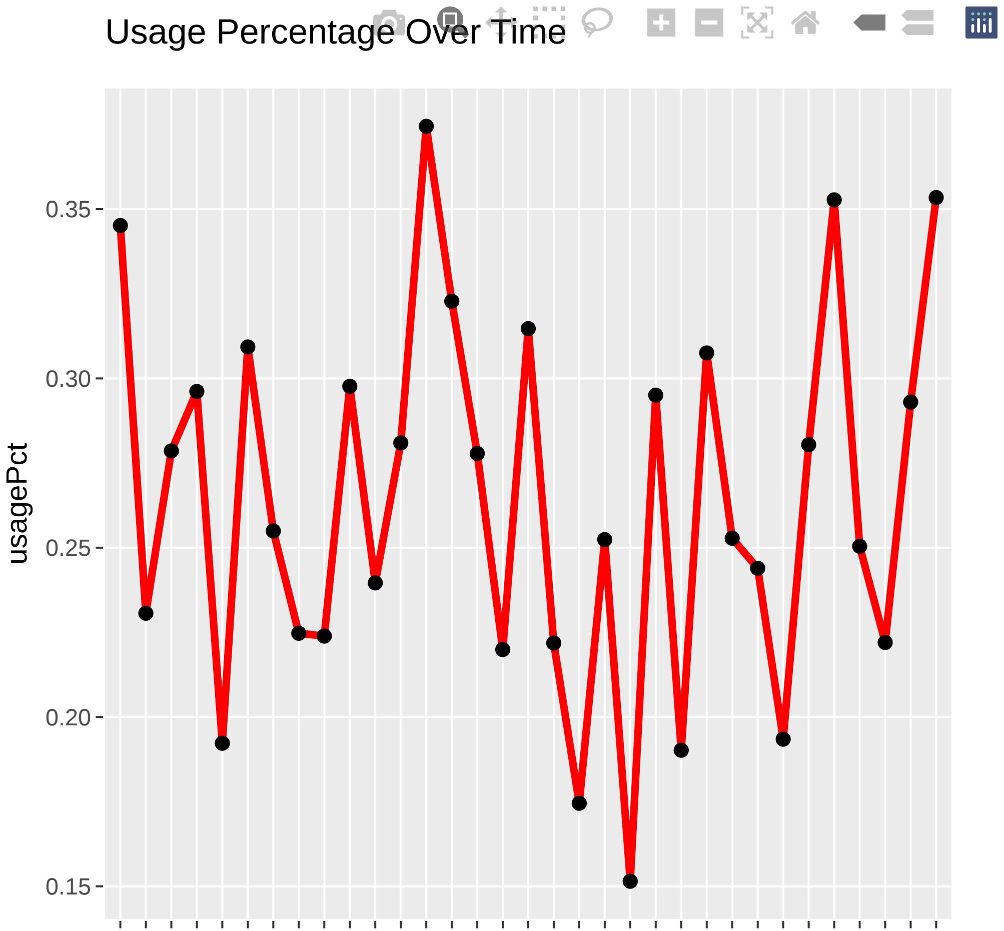
Hide

```
ggplot(py$statsdf,aes(x=ptsScored,y=isHome)) + geom_violin(aes(color=isWin))
```



Hide

```
ggplotly(ggplot(py$statsdf[order(py$statsdf$gameDate), ],aes(x=gameDate,y=usagePct, group = teamI
d)) + geom_line(color='red',linewidth=1) + geom_point() + ggtitle('Usage Percentage Over Time'))
```



Hide

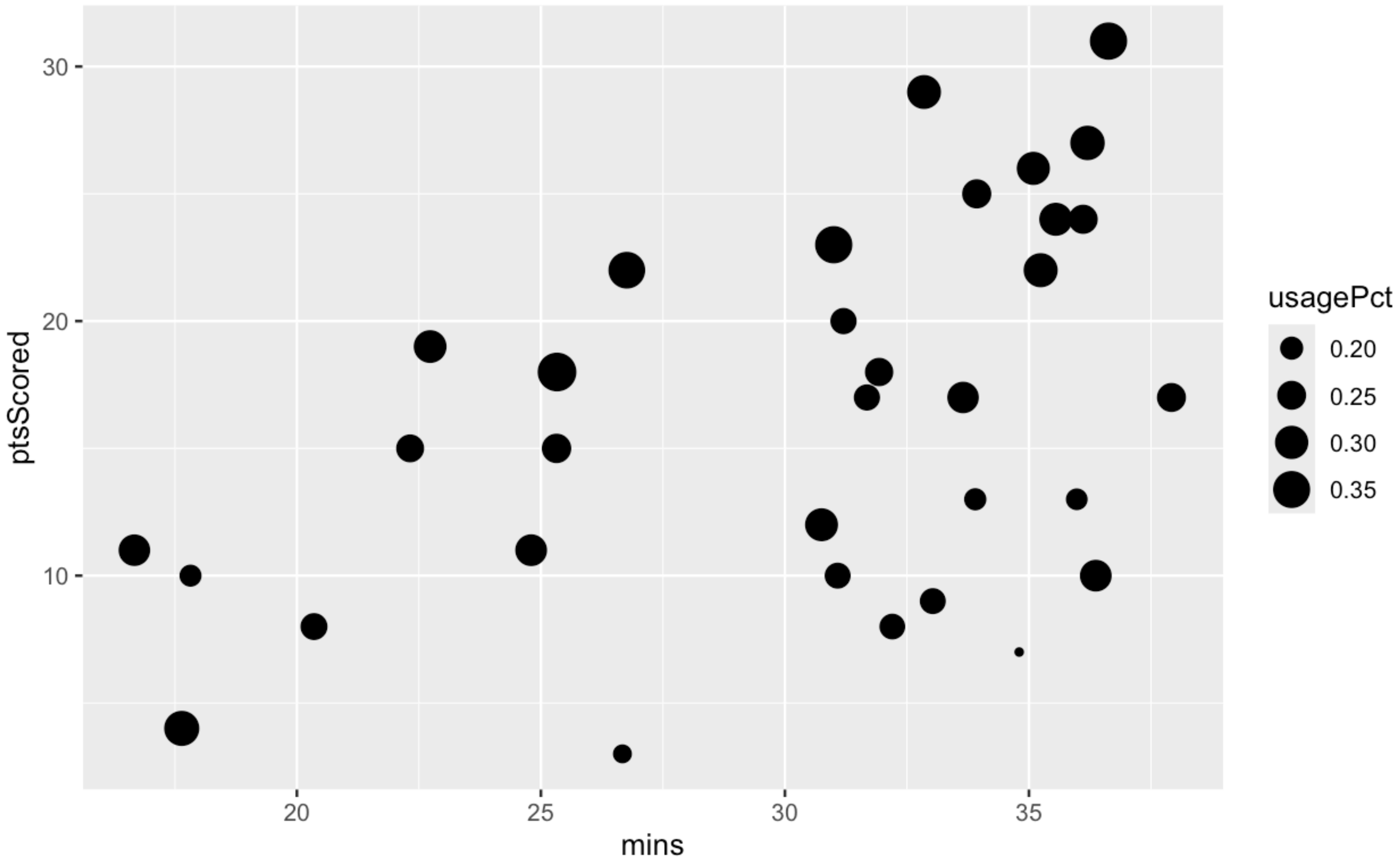
```
py$shotsdf %>% group_by(subType) %>% summarise(avgDuration = mean(actionDuration,na.rm=T))
```

subType	avgDuration
<chr>	<dbl>
drivinglayup	13.623077
dunk	8.626829
floatingjumpshot	15.888889
hookshot	17.400000
jumpshot	16.015556
layup	10.085833
pullupjumpshot	11.933333
stepbackjumpshot	18.125000
tipindunk	0.000000
tipinlayup	0.000000

1-10 of 11 rows Previous **1** 2 Next

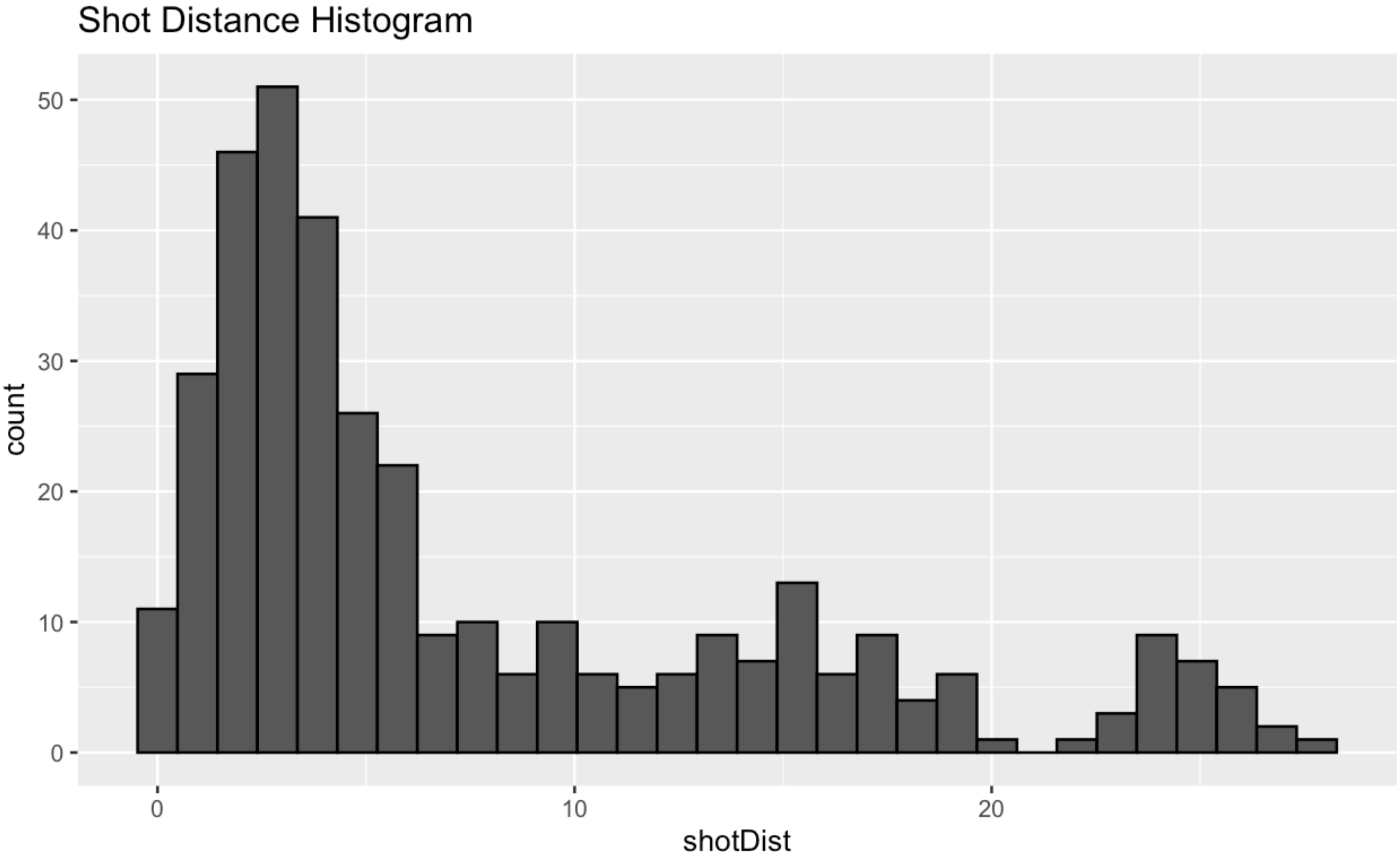
Hide

```
ggplot(py$statsdf,aes(x=mins,y=ptsScored)) + geom_point(aes(size=usagePct))
```



Hide

```
ggplot(py$shotsdf,aes(x=shotDist)) + geom_histogram(color='black') + ggtitle('Shot Distance Histog
ram')
```

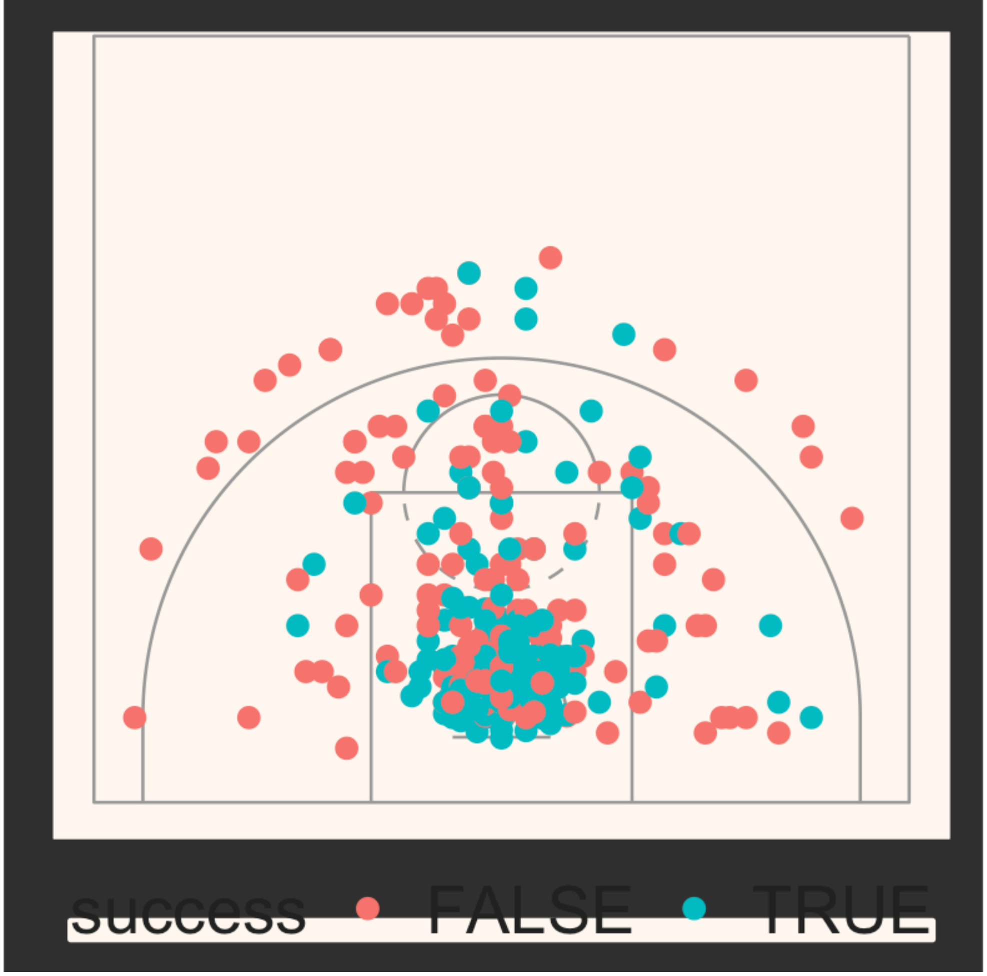

Shot Distance Histogram
count
50
40
30
20
10
0
0
10
20
shotDist

Hide

```
circle_points = function(center = c(0, 0), radius = 1, npoints = 360) {
  angles = seq(0, 2 * pi, length.out = npoints)
  return(data_frame(x = center[1] + radius * cos(angles),
                    y = center[2] + radius * sin(angles)))
}

# Court Dimenons & lines
width = 50
height = 94 / 2
key_height = 19
inner_key_width = 12
outer_key_width = 16
backboard_width = 6
backboard_offset = 4
neck_length = 0.5
hoop_radius = 0.75
hoop_center_y = backboard_offset + neck_length + hoop_radius
three_point_radius = 23.75
three_point_side_radius = 22
three_point_side_height = 14

# Court themes
court_themes = list(
  light = list(
    court = 'floralwhite',
    lines = '#999999',
    text = '#222222',
    made = '#00bfc4',
    missed = '#f8766d',
    hex_border_size = 1,
    hex_border_color = "#000000"
  ),
  dark = list(
    court = '#000004',
    lines = '#999999',
    text = '#f0f0f0',
    made = '#00bfc4',
    missed = '#f8766d',
    hex_border_size = 0,
    hex_border_color = "#000000"
  ),
  ppt = list(
    court = 'gray20',
    lines = 'white',
    text = '#f0f0f0',
    made = '#00bfc4',
    missed = '#f8766d',
    hex_border_size = 0,
    hex_border_color = "gray20"
)
)

# Function to create court based on given dimensions
plot_court = function(court_theme = court_themes$light, use_short_three = FALSE) {
  if (use_short_three) {
    three_point_radius = 22
    three_point_side_height = 0
  }

  court_points = data_frame(
    x = c(width / 2, width / 2, -width / 2, -width / 2, width / 2),
    y = c(height, 0, 0, height, height),
    desc = "perimeter"
  )

  court_points = bind_rows(court_points , data_frame(
    x = c(outer_key_width / 2, outer_key_width / 2, -outer_key_width / 2, -outer_key_width / 2),
    y = c(0, key_height, key_height, 0),
    desc = "outer_key"
  ))

  court_points = bind_rows(court_points , data_frame(
    x = c(-backboard_width / 2, backboard_width / 2),
    y = c(backboard_offset, backboard_offset),
    desc = "backboard"
  ))

  court_points = bind_rows(court_points , data_frame(
    x = c(0, 0), y = c(backboard_offset, backboard_offset + neck_length), desc = "neck"
  ))

  foul_circle = circle_points(center = c(0, key_height), radius = inner_key_width / 2)

  foul_circle_top = filter(foul_circle, y > key_height) %>%
    mutate(desc = "foul_circle_top")

  foul_circle_bottom = filter(foul_circle, y < key_height) %>%
    mutate(
      angle = atan((y - key_height) / x) * 180 / pi,
      angle_group = floor((angle - 5.625) / 11.25),
      desc = paste0("foul_circle_bottom_", angle_group)
    ) %>%
    filter(angle_group %% 2 == 0) %>%
    select(x, y, desc)

  hoop = circle_points(center = c(0, hoop_center_y), radius = hoop_radius) %>%
    mutate(desc = "hoop")

  restricted = circle_points(center = c(0, hoop_center_y), radius = 4) %>%
    filter(y >= hoop_center_y) %>%
    mutate(desc = "restricted")

  three_point_circle = circle_points(center = c(0, hoop_center_y), radius = three_point_radius) %
>%
    filter(y >= three_point_side_height, y >= hoop_center_y)

  three_point_line = data_frame(
    x = c(three_point_side_radius, three_point_side_radius, three_point_circle$x, -three_point_sid
e_radius, -three_point_side_radius),
    y = c(0, three_point_side_height, three_point_circle$y, three_point_side_height, 0),
    desc = "three_point_line"
  )

  court_points = bind_rows(
    court_points,
    foul_circle_top,
    foul_circle_bottom,
    hoop,
    restricted,
    three_point_line
  )


  court_points <- court_points

  # Final plot creation
  ggplot() +
    geom_path(
      data = court_points,
      aes(x = x, y = y, group = desc),
      color = court_theme$lines
    ) +
    coord_fixed(ylim = c(0, 45), xlim = c(-25, 25)) +
    theme_minimal(base_size = 22) +
    theme(
      text = element_text(color = court_theme$text),
      plot.background = element_rect(fill = 'gray20', color = 'gray20'),
      panel.background = element_rect(fill = court_theme$court, color = court_theme$court),
      panel.grid = element_blank(),
      panel.border = element_blank(),
      axis.text = element_blank(),
      axis.title = element_blank(),
      axis.ticks = element_blank(),
      legend.background = element_rect(fill = court_theme$court, color = court_theme$court),
      legend.margin = margin(-1, 0, 0, 0, unit = "lines"),
      legend.position = "bottom",
      legend.key = element_blank(),
      legend.text = element_text(size = rel(1.0))
    )
```

Hide

```
plot_court(use_short_three = T) + geom_point(aes(x=x-25,y=-y+47.5,color=success),size=3,data=py$shotsdf)
```

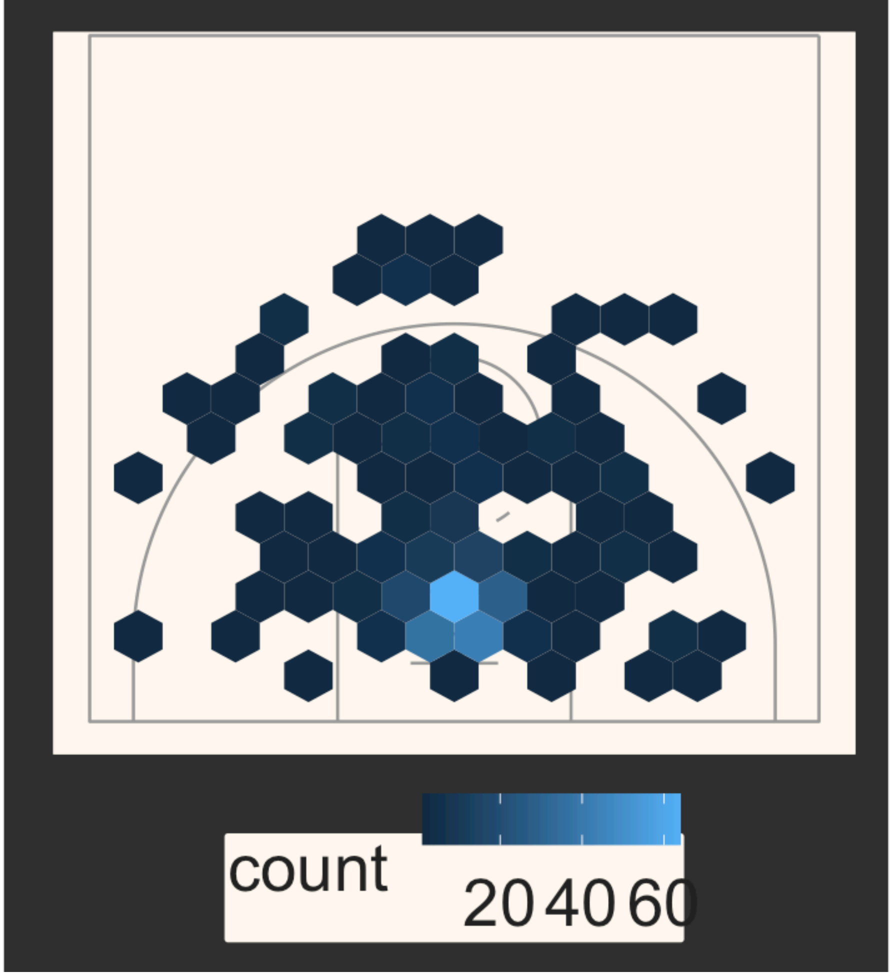


Hide

```
# geom_text(aes(x=x-25,y=-y+47.5,label=shotDist),data=py$shotsdf)
```

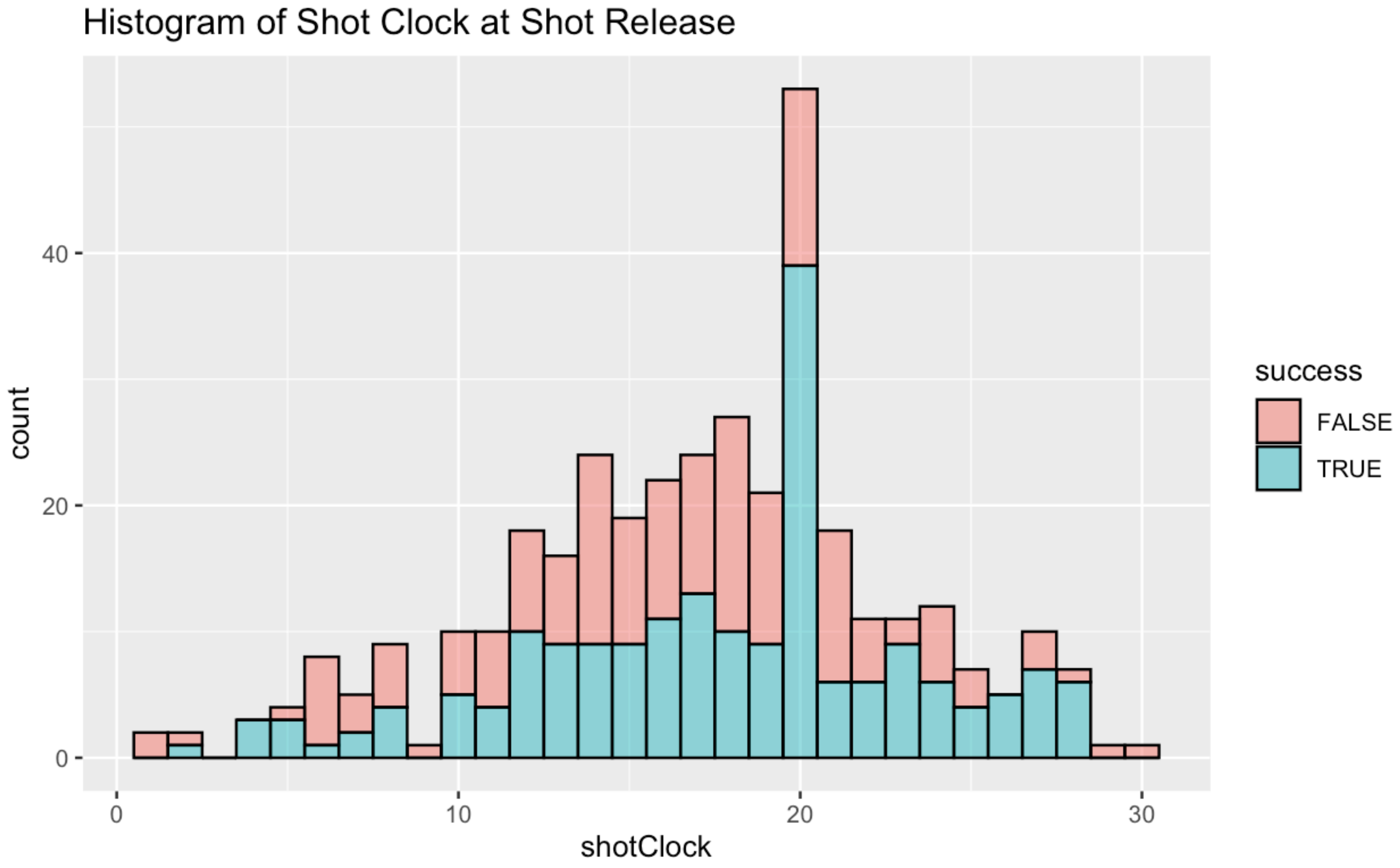
Hide

```
plot_court(use_short_three = T) + geom_hex(aes(x=x-25,y=-y+47.5),bins=15,data=py$shotsdf)
```



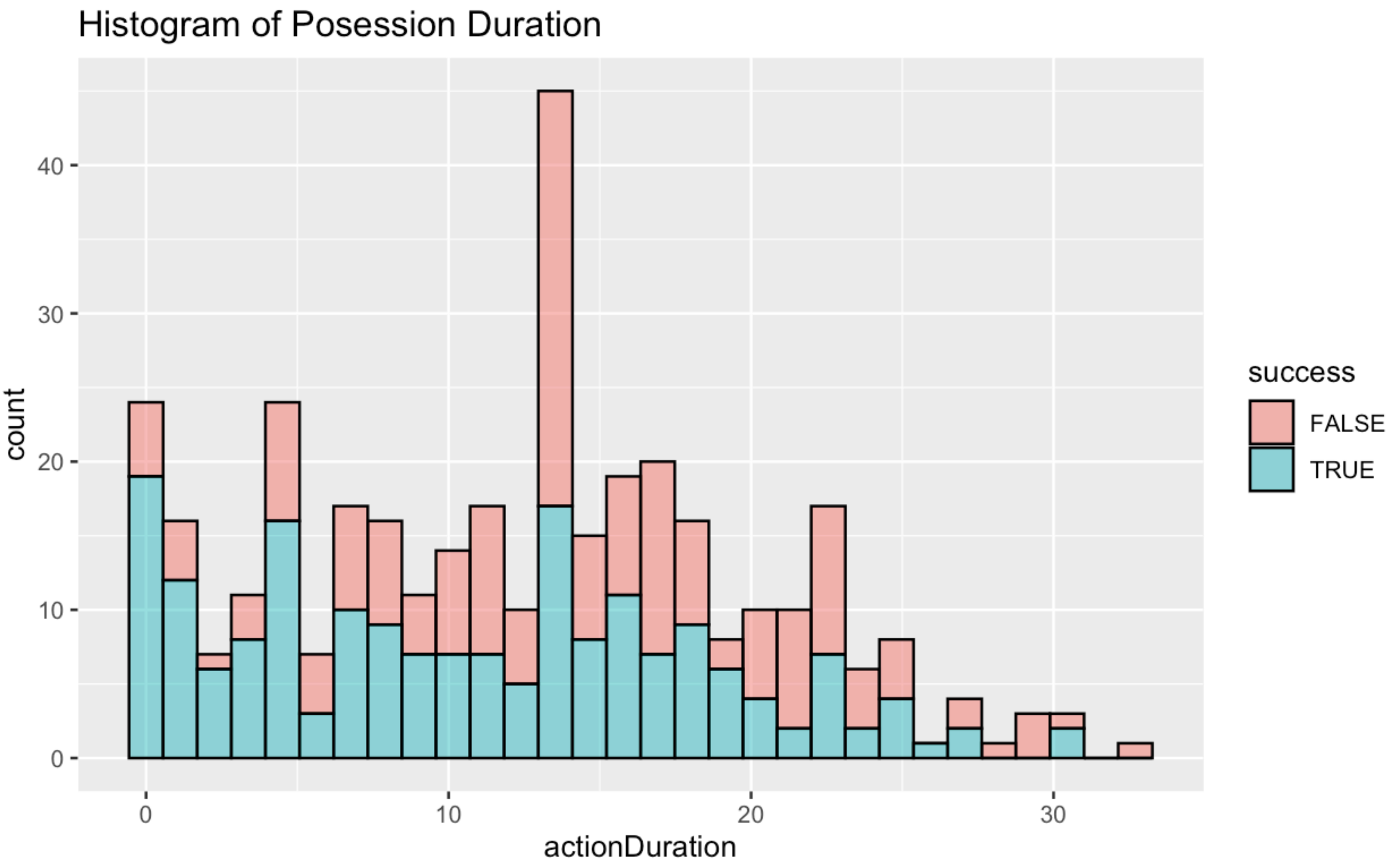
Hide

```
ggplot(py$shotsdf,aes(x=shotClock)) + geom_histogram(aes(fill = success),color='black',alpha=0.5)
+ ggtitle('Histogram of Shot Clock at Shot Release')
```



Hide

```
ggplot(py$shotsdf,aes(x=actionDuration)) + geom_histogram(aes(fill = success),color='black',alpha=
0.5) + ggtitle('Histogram of Posession Duration')
```



Hide

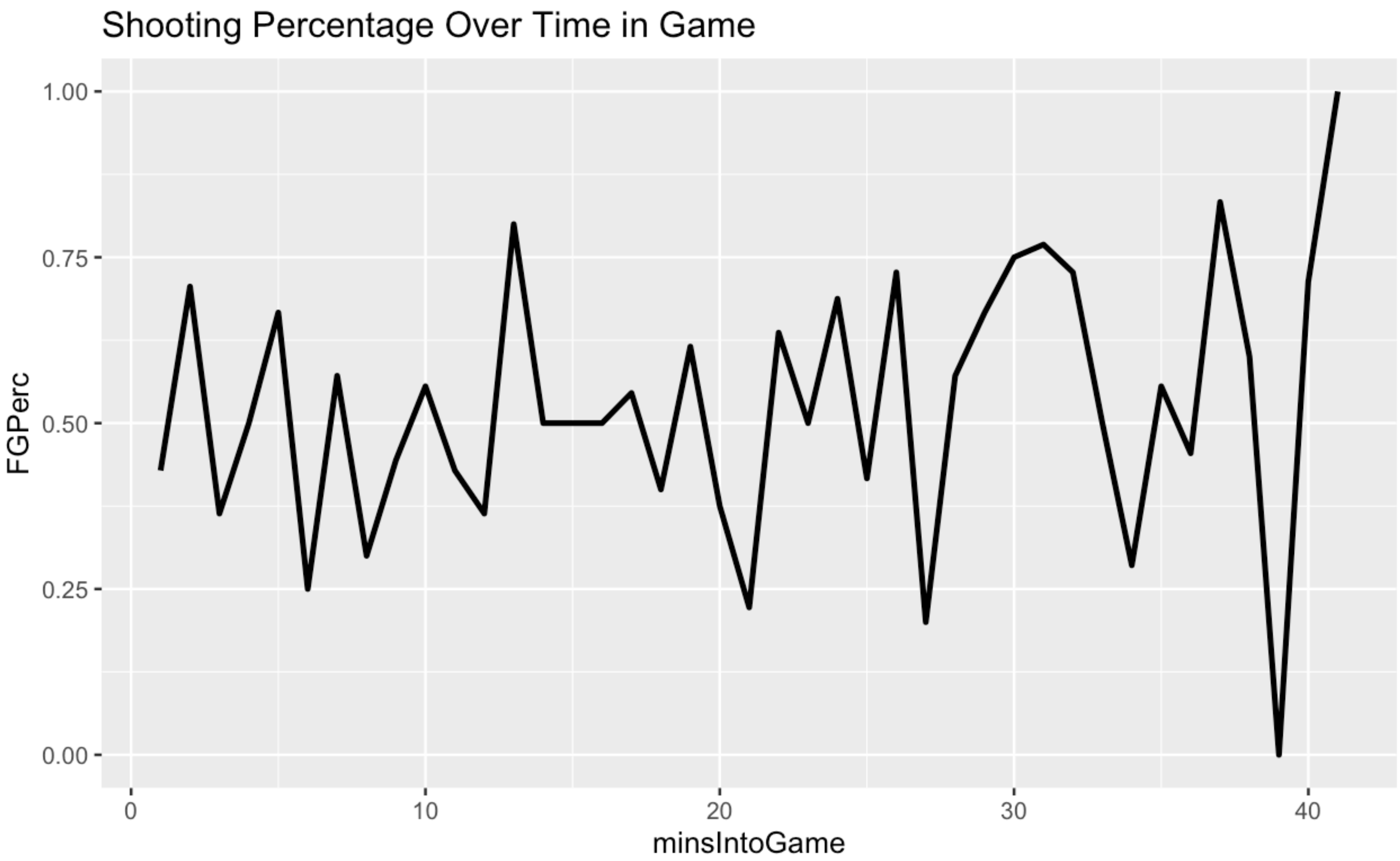
```
py$shotsdf %>% group_by(subType) %>% summarise(avgDist = mean(shotDist,na.rm=T),shots = n())
```

subType	avgDist	shots
<chr>	<dbl>	<int>
drivinglayup	3.3607436	39
dunk	2.5190244	41
floatingjumpshot	7.1104444	9
hookshot	8.2755333	15
jumpshot	15.8613556	90
layup	3.4159083	120
pullupjumpshot	14.6342667	15
stepbackjumpshot	15.3825000	8
tipindunk	1.7580000	1
tipinlayup	0.6885714	14

1-10 of 11 rows Previous **1** 2 Next

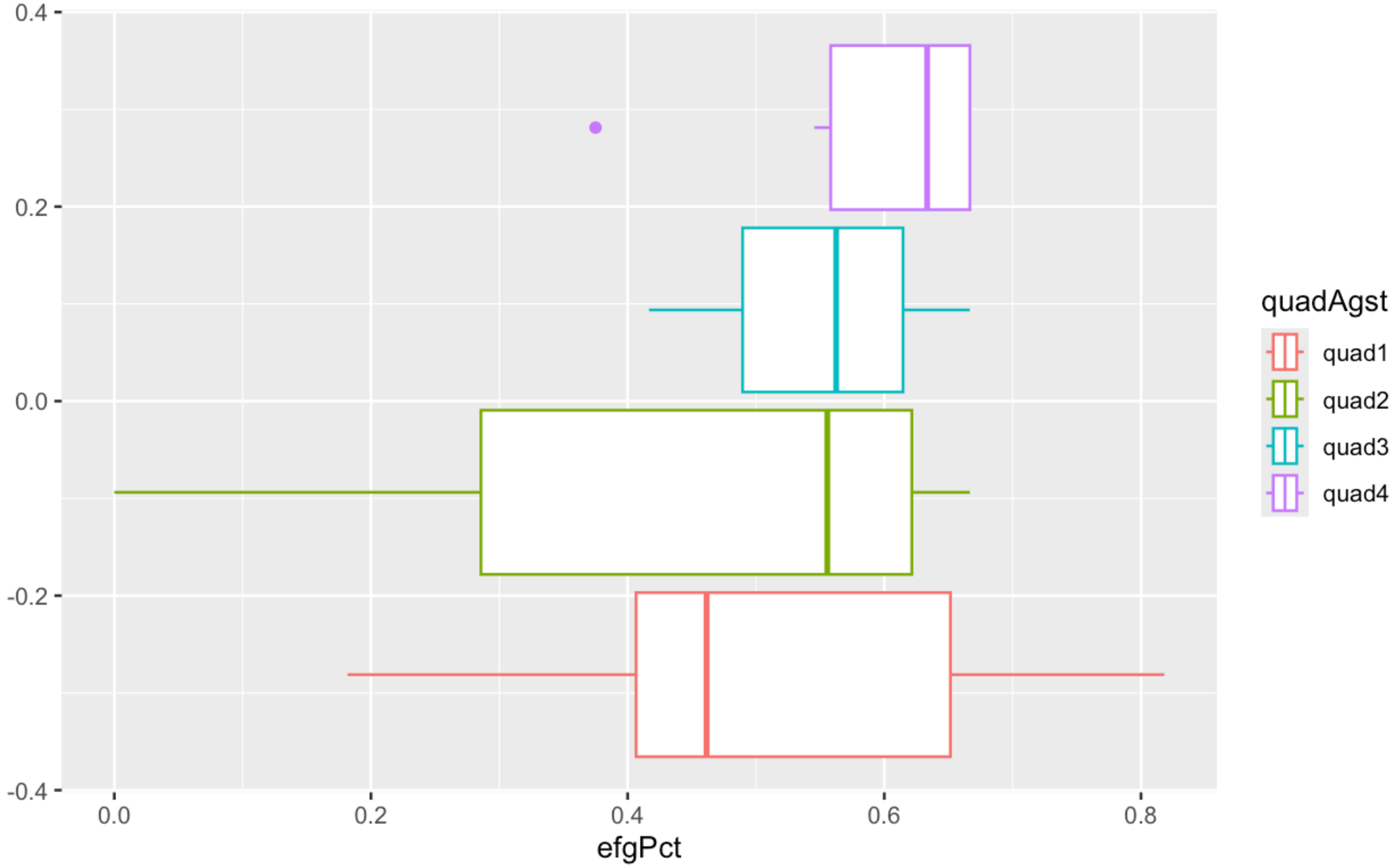
Hide

```
ggplot(py$shotsdf %>% group_by(minsIntoGame) %>% summarise(FGPerc = mean(success,na.rm=T)),aes(x=m
insIntoGame,y=FGPerc)) + geom_line(linewidth=1) + ggtitle('Shooting Percentage Over Time in Game')
```



Hide

```
ggplot(py$statsdf,aes(x=efgPct)) + geom_boxplot(aes(color=quadAgst))
```


0.4
0.2
0.0
-0.2
-0.4
0.0
0.2
0.4
0.6
0.8
efgPct
quadAgst
quad1
quad2
quad3
quad4