

# Basketball Analysis - NBA 2017-2018

Karl Mbouombouo

2023-09-18

## PACKAGES

```
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
```

```
library(ggplot2)  
library(gridExtra)
```

```
##  
## Attaching package: 'gridExtra'  
## The following object is masked from 'package:dplyr':  
##  
##   combine
```

```
library(vcd)
```

```
## Loading required package: grid
```

```
library(MASS)
```

```
##  
## Attaching package: 'MASS'  
## The following object is masked from 'package:dplyr':  
##  
##   select
```

```
library(lsr)  
library(tibble)  
library(base)  
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':  
##   method from  
##   +.gg    ggplot2
```

```
library(tidygraph)
```

```
##
## Attaching package: 'tidygraph'
## The following object is masked from 'package:MASS':
##
##     select
## The following object is masked from 'package:stats':
##
##     filter
```

```
library(igraph)
```

```
##
## Attaching package: 'igraph'
## The following object is masked from 'package:tidygraph':
##
##     groups
## The following object is masked from 'package:tibble':
##
##     as_data_frame
## The following objects are masked from 'package:dplyr':
##
##     as_data_frame, groups, union
## The following objects are masked from 'package:stats':
##
##     decompose, spectrum
## The following object is masked from 'package:base':
##
##     union
```

```
library(dendextend)
```

```
##
## -----
## Welcome to dendextend version 1.17.1
## Type citation('dendextend') for how to cite the package.
##
## Type browseVignettes(package = 'dendextend') for the package vignette.
## The github page is: https://github.com/talgalili/dendextend/
##
## Suggestions and bug-reports can be submitted at: https://github.com/talgalili/dendextend/issues
## You may ask questions at stackoverflow, use the r and dendextend tags:
##   https://stackoverflow.com/questions/tagged/dendextend
##
## To suppress this message use: suppressPackageStartupMessages(library(dendextend))
## -----
##
## Attaching package: 'dendextend'
## The following object is masked from 'package:stats':
##
```

```
##      cutree
RNGkind(sample.kind="Rounding")

## Warning in RNGkind(sample.kind = "Rounding"): non-uniform 'Rounding' sampler
## used

rm(list=ls())
# install.packages("devtools", repos="https://cran.stat.unipd.it/")
# devtools::install_github("sndmrc/BasketballAnalyzeR", force=TRUE)
library(BasketballAnalyzeR)

##
## If you want to reproduce the figures contained in the book of
## Zuccolotto and Manisera (2020) and
## if the version of your R machine is >= 3.6.0, you need to type
## RNGkind(sample.kind = "Rounding")
## at the beginning of your working session

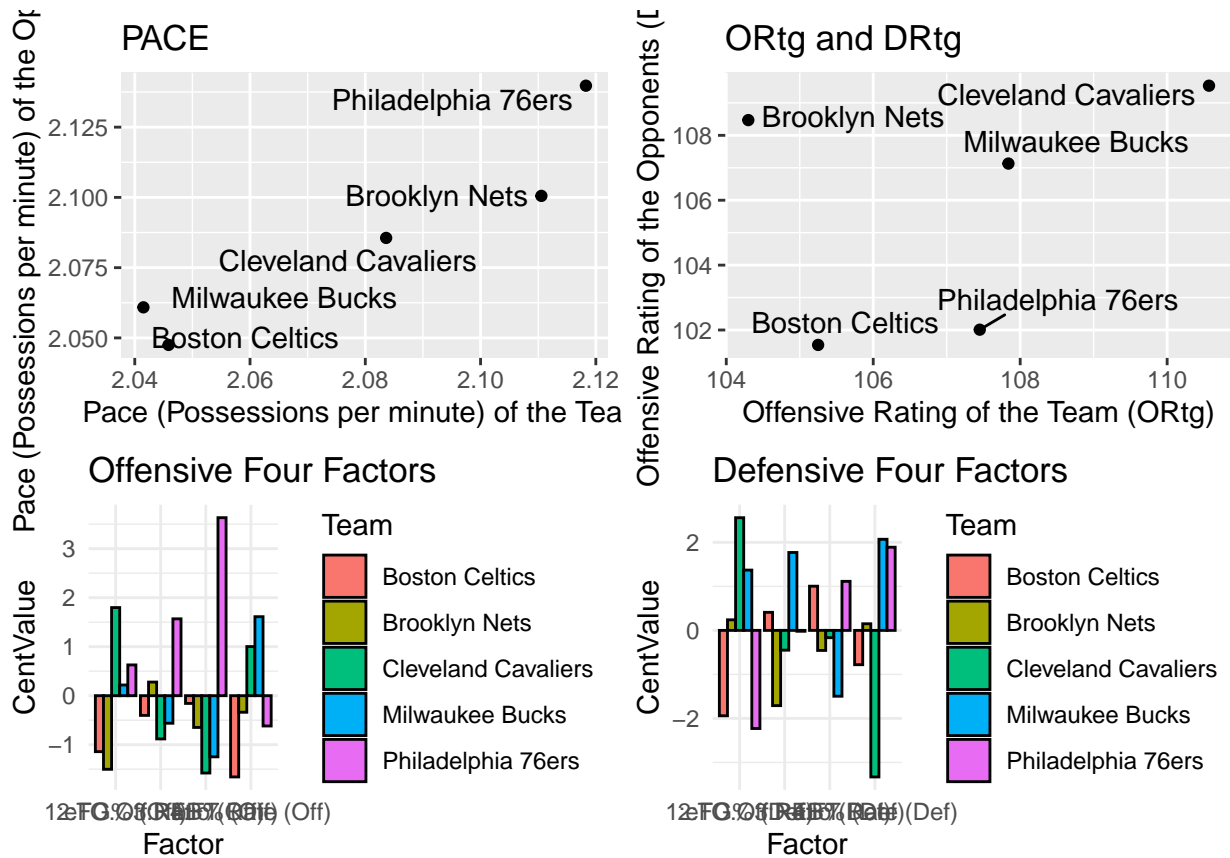
data(package="BasketballAnalyzeR")
nba_pbp <- PbPmanipulation(PbP.BDB)
```

## STATISTICAL ANALYSIS

```
# Analysis TOP 6 Eastern Conference
# four factor: Shooting actor, turnovers, rebounding, and free throws
rm(list=ls())

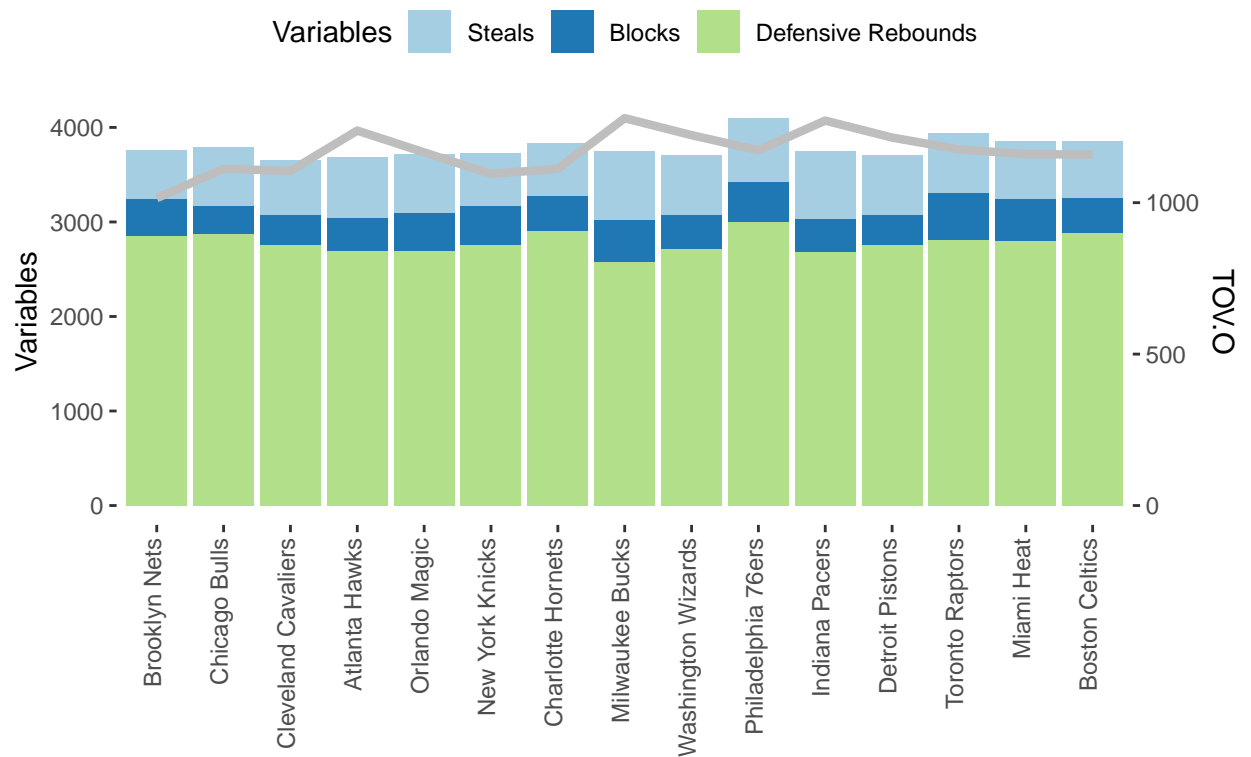
tm <- c("MIL", "BOS", "PHI", "CLE", "NY", "BKN")
selTeams <- which(Tadd$team %in% tm)
FF.sel <- fourfactors(Tbox[selTeams,], Obox[selTeams,])

plot(FF.sel)
```



## Bar Plots

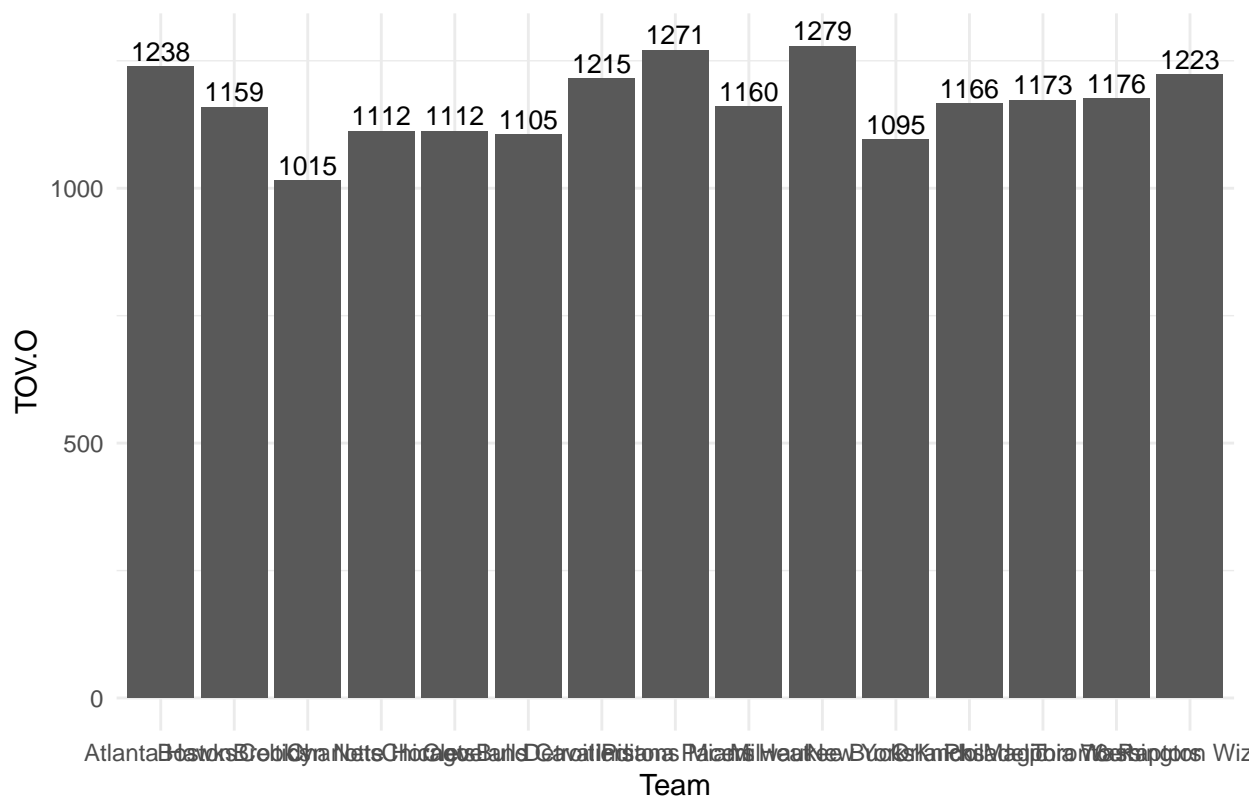
```
X <- data.frame(Tbox, PTS.0=0box$PTS, TOV.0=0box$TOV,
                CONF=Tadd$Conference)
XE <- subset(X, CONF=="E")
labs <- c("Steals", "Blocks", "Defensive Rebounds")
barline(data=XE, id="Team", bars=c("STL", "BLK", "DREB"),
        line="TOV.0", order.by="PTS.0", labels.bars=labs)
```



Bars ordered by PTS.O

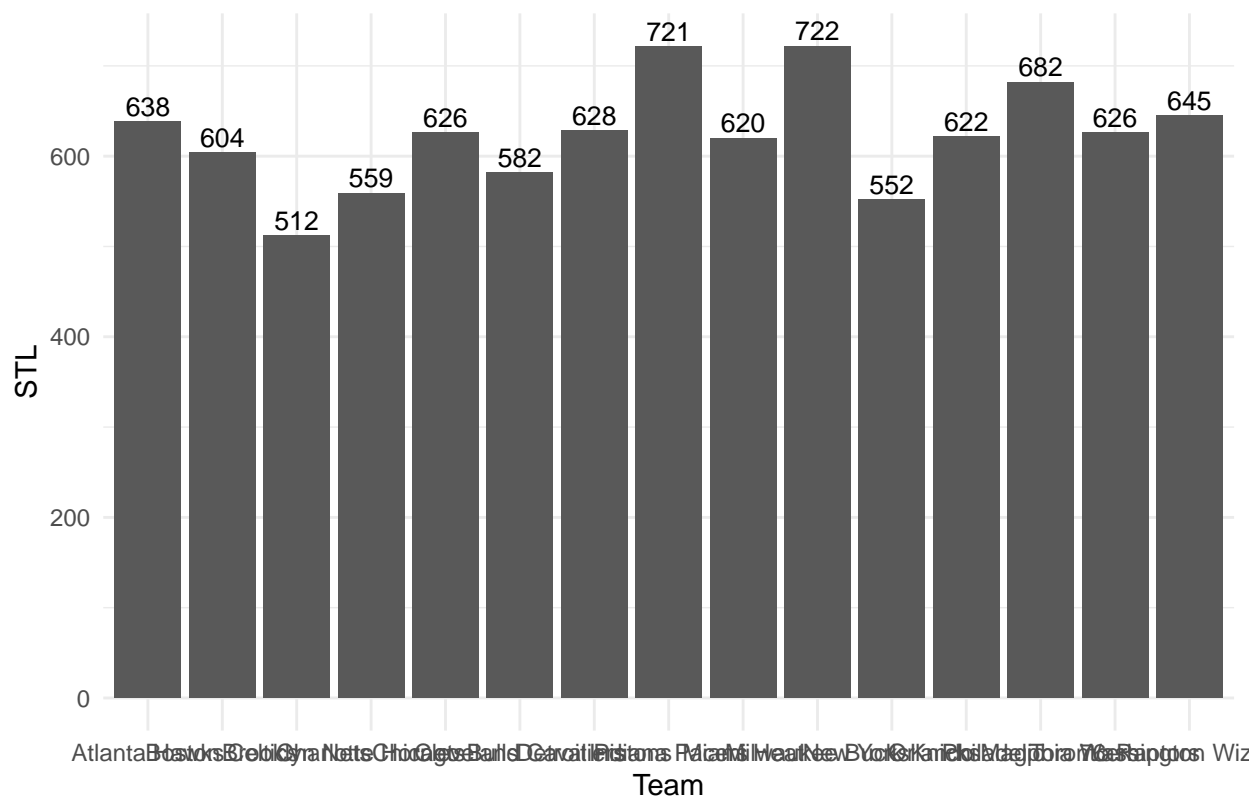
```
ggplot(XE, aes(x=Team, y=TOV.O)) +  
  geom_bar(stat="identity") +  
  geom_text(aes(label=TOV.O), vjust=-0.3, size=3.5) +  
  ggtitle("Team Eastern Conference, Total Turn Over") +  
  theme_minimal()
```

Team Eastern Conference, Total Turn Over

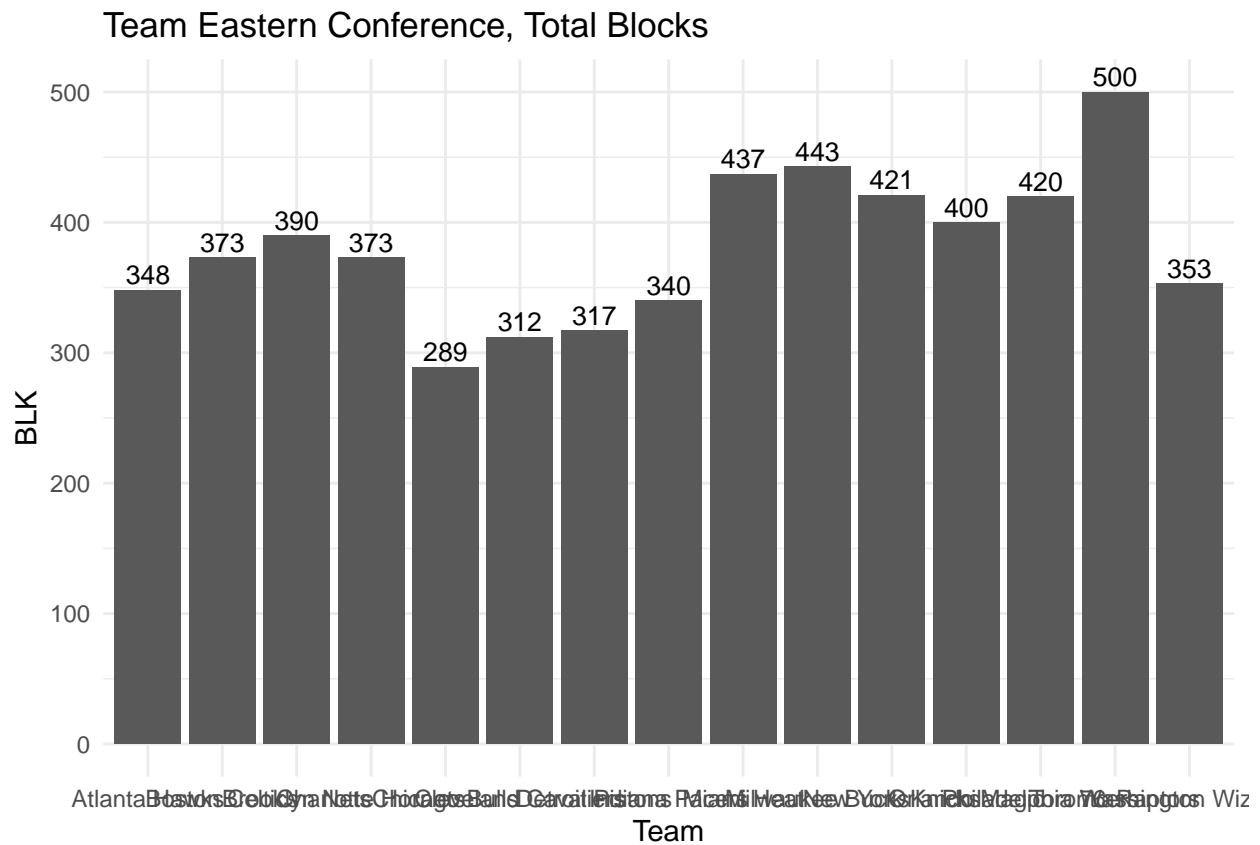


```
ggplot(XE, aes(x=Team, y=STL)) +
  geom_bar(stat="identity") +
  geom_text(aes(label=STL), vjust=-0.3, size=3.5) +
  ggtitle("Team Eastern Conference, Total Steals") +
  theme_minimal()
```

Team Eastern Conference, Total Steals



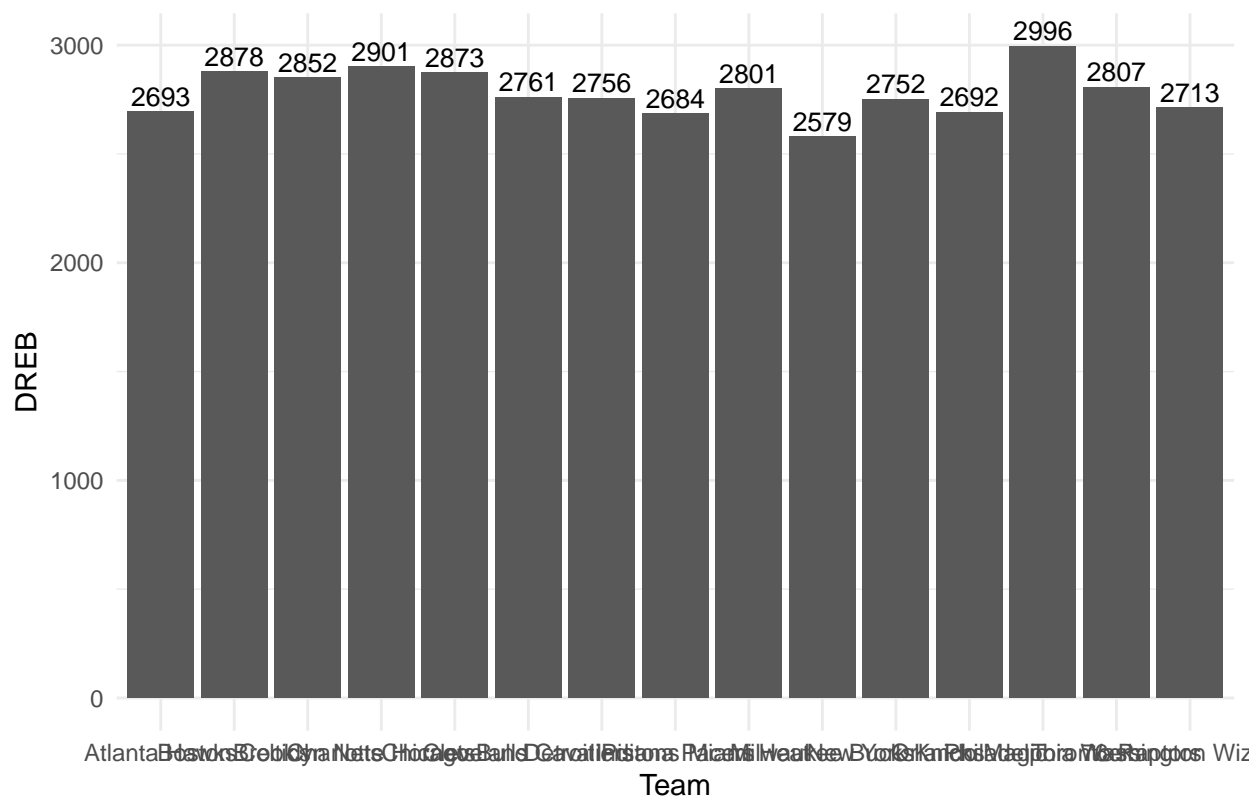
```
ggplot(XE, aes(x=Team, y=BLK)) +
  geom_bar(stat="identity") +
  geom_text(aes(label=BLK), vjust=-0.3, size=3.5) +
  ggtitle("Team Eastern Conference, Total Blocks") +
  theme_minimal()
```



```
ggplot(XE, aes(x=Team, y=DREB)) +
  geom_bar(stat="identity") +
  geom_text(aes(label=DREB), vjust=-0.3, size=3.5) +
  ggtitle("Team Eastern Conference, Total Defensive Rebound") +
  theme_minimal()
```



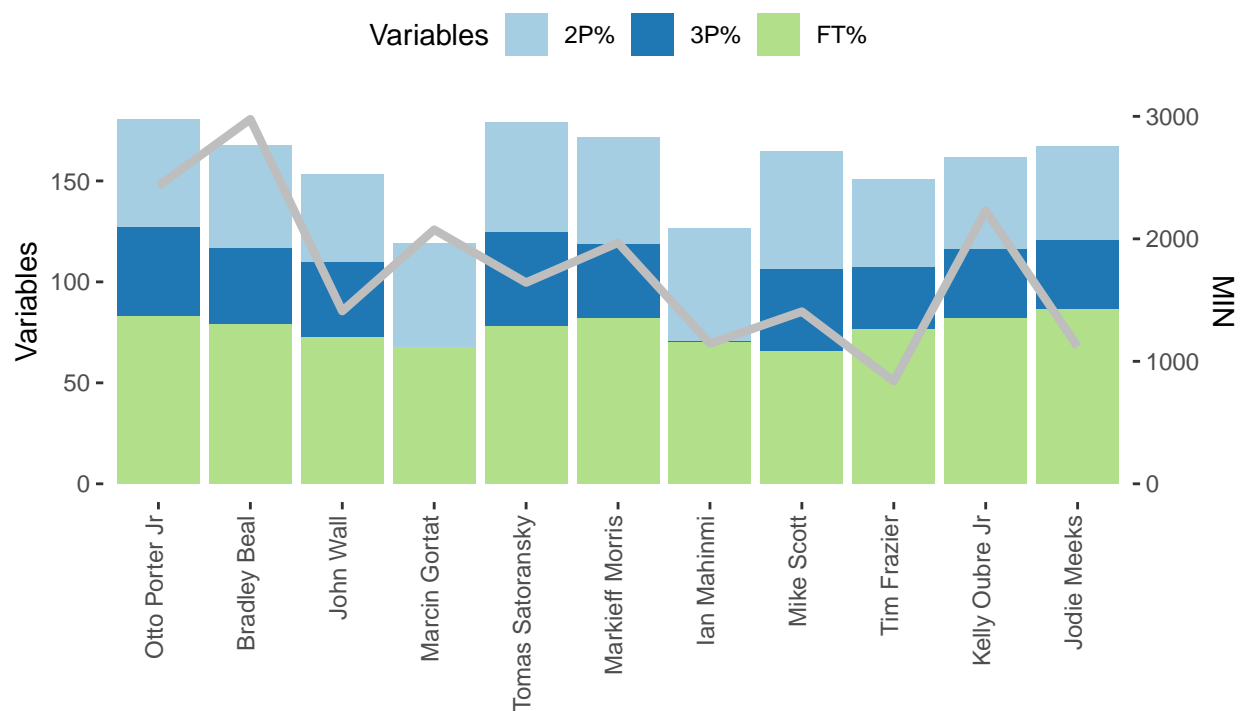
Team Eastern Conference, Total Defensive Rebound



```
Pbox.WW <- subset(Pbox, Team=="Washington Wizards" &
  MIN>=500)

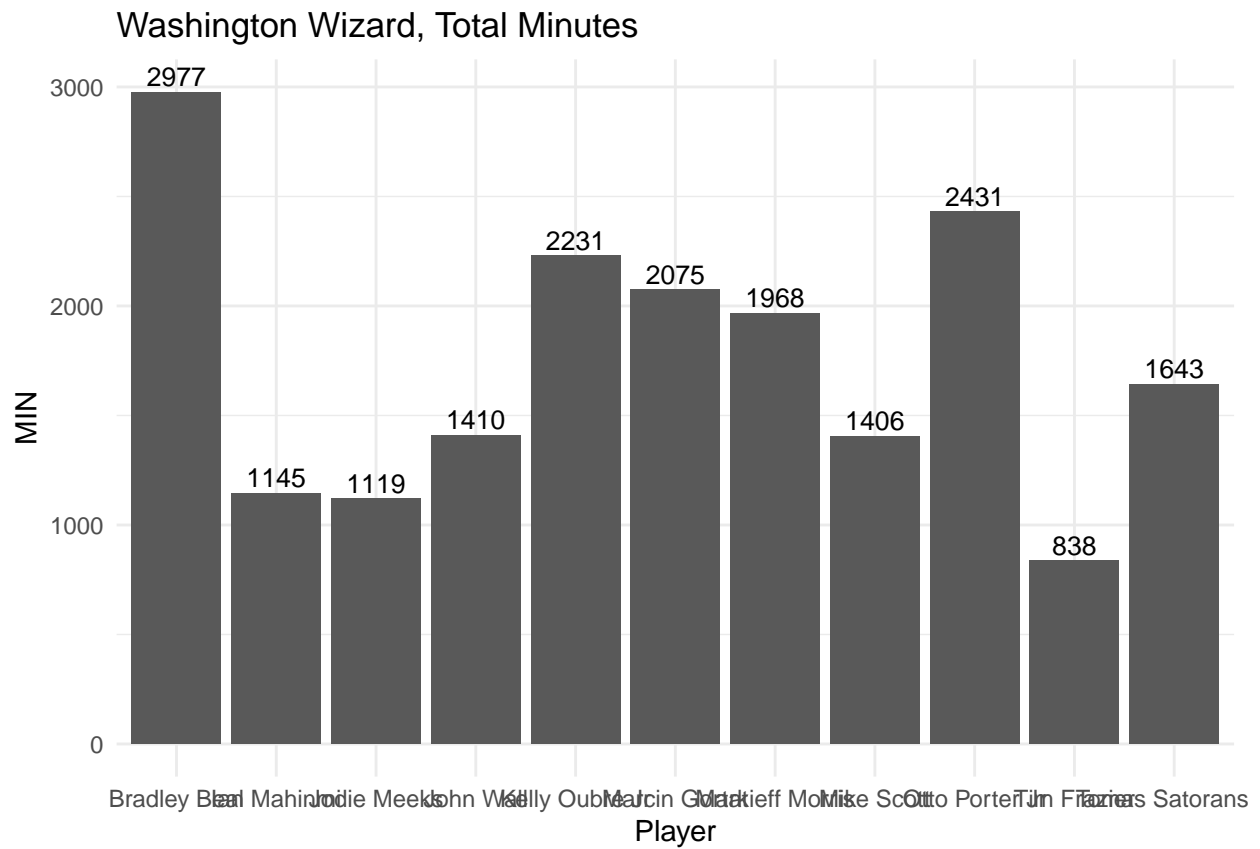
barline(data=Pbox.WW, id="Player",
  bars=c("P2p", "P3p", "FTp"), line="MIN",
  order.by="PM", labels.bars=c("2P%", "3P%", "FT%"),
  title="Washington Wizards")
```

## Washington Wizards



Bars ordered by PM

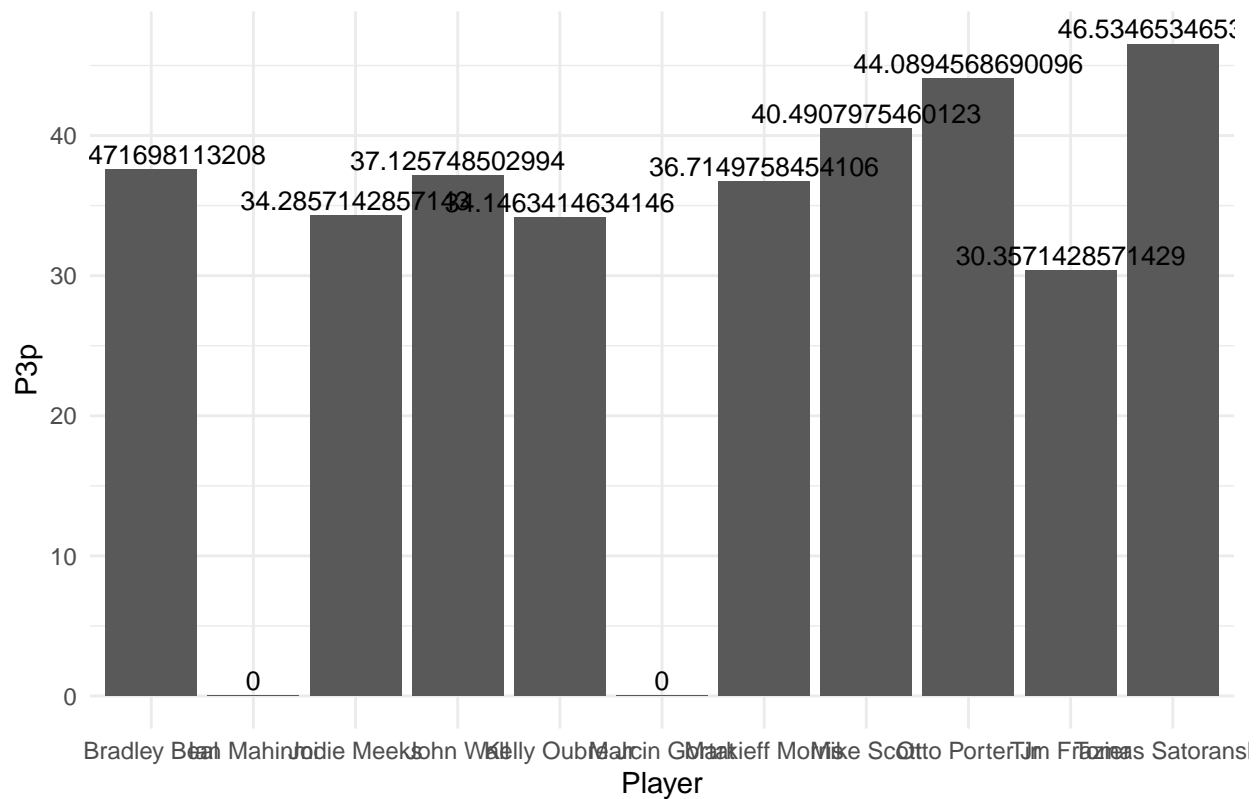
```
ggplot(Pbox.WW, aes(x=Player, y=MIN)) +
  geom_bar(stat="identity") +
  geom_text(aes(label=MIN), vjust=-0.3, size=3.5) +
  ggtitle("Washington Wizard, Total Minutes") +
  theme_minimal()
```



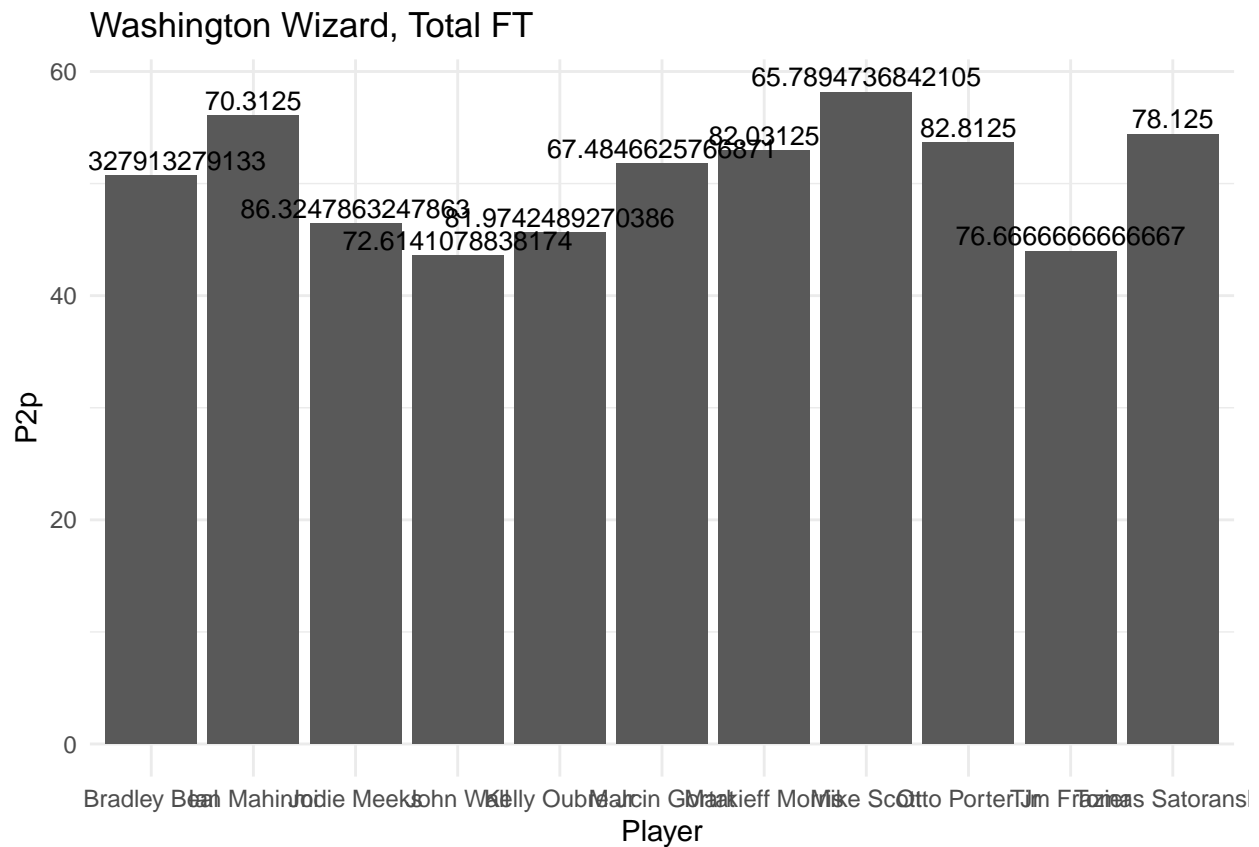
```
ggplot(Pbox.WW, aes(x=Player, y=P2p)) +
  geom_bar(stat="identity") +
  geom_text(aes(label=P2p), vjust=-0.3, size=3.5) +
  ggtitle("Washington Wizard, Total FG") +
  theme_minimal()
```



## Washington Wizard, Total 3FG



```
ggplot(Pbox.WW, aes(x=Player, y=P2p)) +
  geom_bar(stat="identity") +
  geom_text(aes(label=FTp), vjust=-0.3, size=3.5) +
  ggtitle("Washington Wizard, Total FT") +
  theme_minimal()
```



## Radial Plots

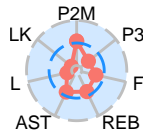
```
attach(Pbox.WW)

X2 <- data.frame(P2M, P3M, FTM, REB=OREB+DREB, AST,
                 STL, BLK)/MIN

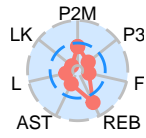
detach(Pbox.WW)

# Standardization OFF
radialprofile(X2, title=Pbox.WW$Player, std=FALSE)
```

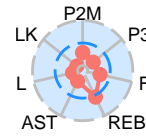
Bradley Beal



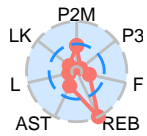
Otto Porter Jr



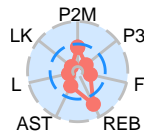
Kelly Oubre Jr



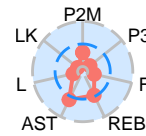
Marcin Gortat



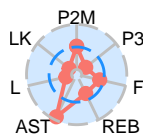
Markieff Morris



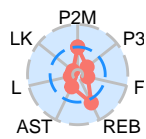
Tomas Satoransky



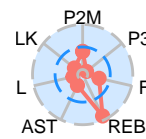
John Wall



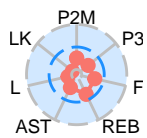
Mike Scott



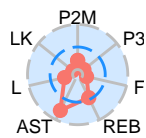
Ian Mahinmi



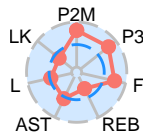
Jodie Meeks



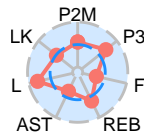
Tim Frazier



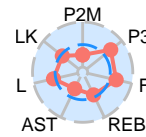
Bradley Beal



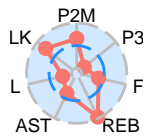
Otto Porter Jr



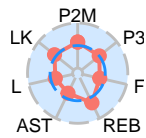
Kelly Oubre Jr



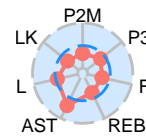
Marcin Gortat



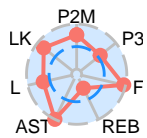
Markieff Morris



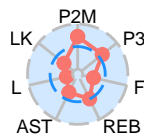
Tomas Satoransky



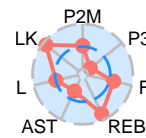
John Wall



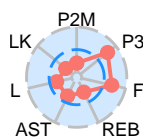
Mike Scott



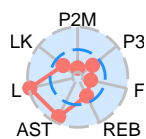
Ian Mahinmi



Jodie Meeks

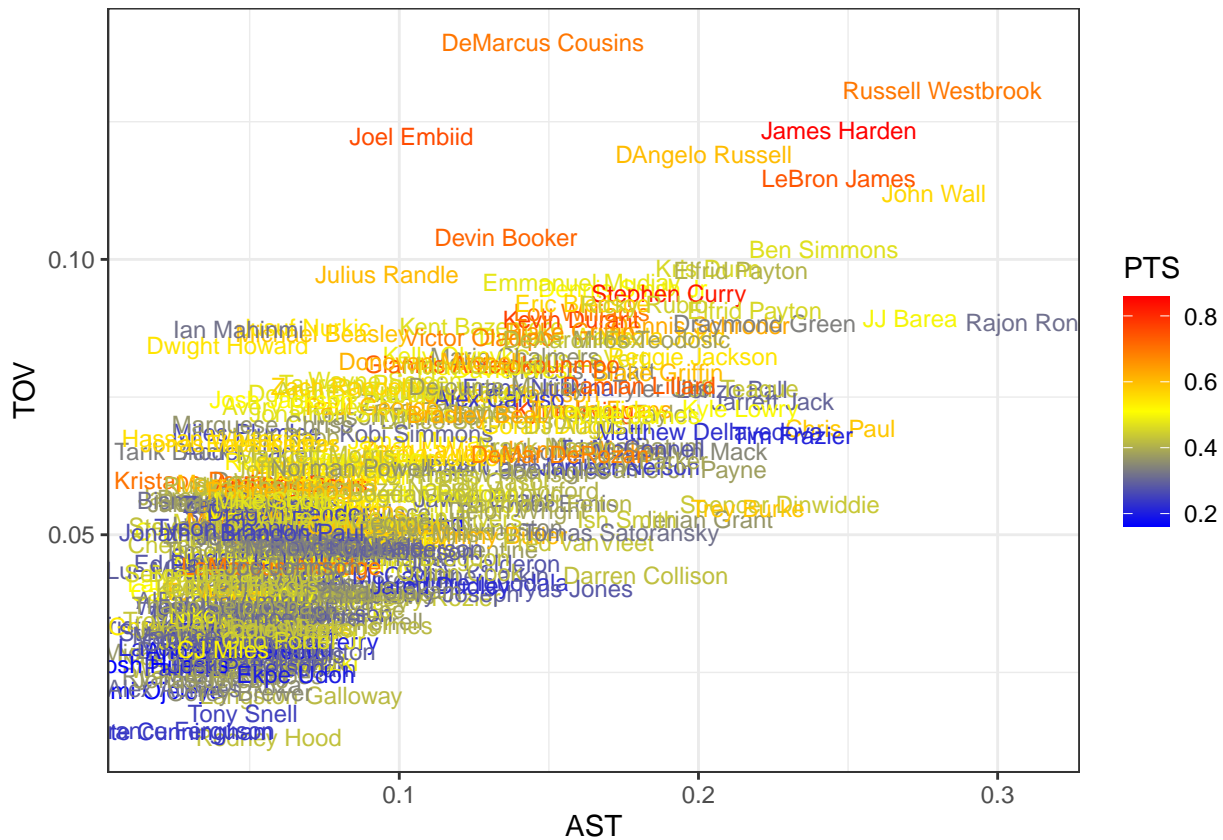


Tim Frazier



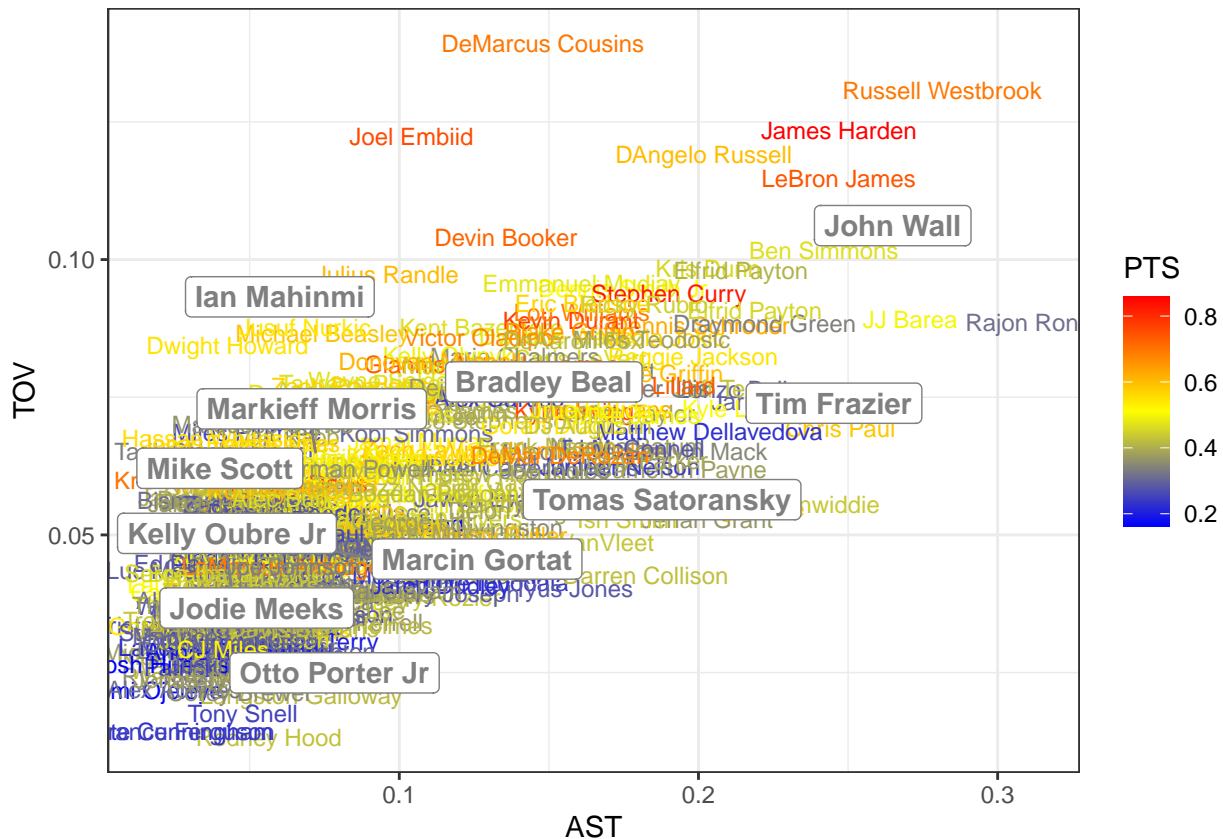
## Scatter Plot

```
Pbox.sel <- subset(Pbox, MIN>= 500)
attach(Pbox.sel)
X3 <- data.frame(AST, TOV, PTS)/MIN
detach(Pbox.sel)
mypal <- colorRampPalette(c("blue","yellow","red"))
scatterplot(X3, data.var=c("AST","TOV"), z.var="PTS",
            labels=Pbox.sel$Player, palette=mypal)
```



```
WAS <- which(Pbox.sel$Team == "Washington Wizards")
scatterplot(X3, data.var = c("AST", "TOV"), z.var="PTS",
            labels = Pbox.sel$Player, palette=mypal,
            subset = WAS)
```

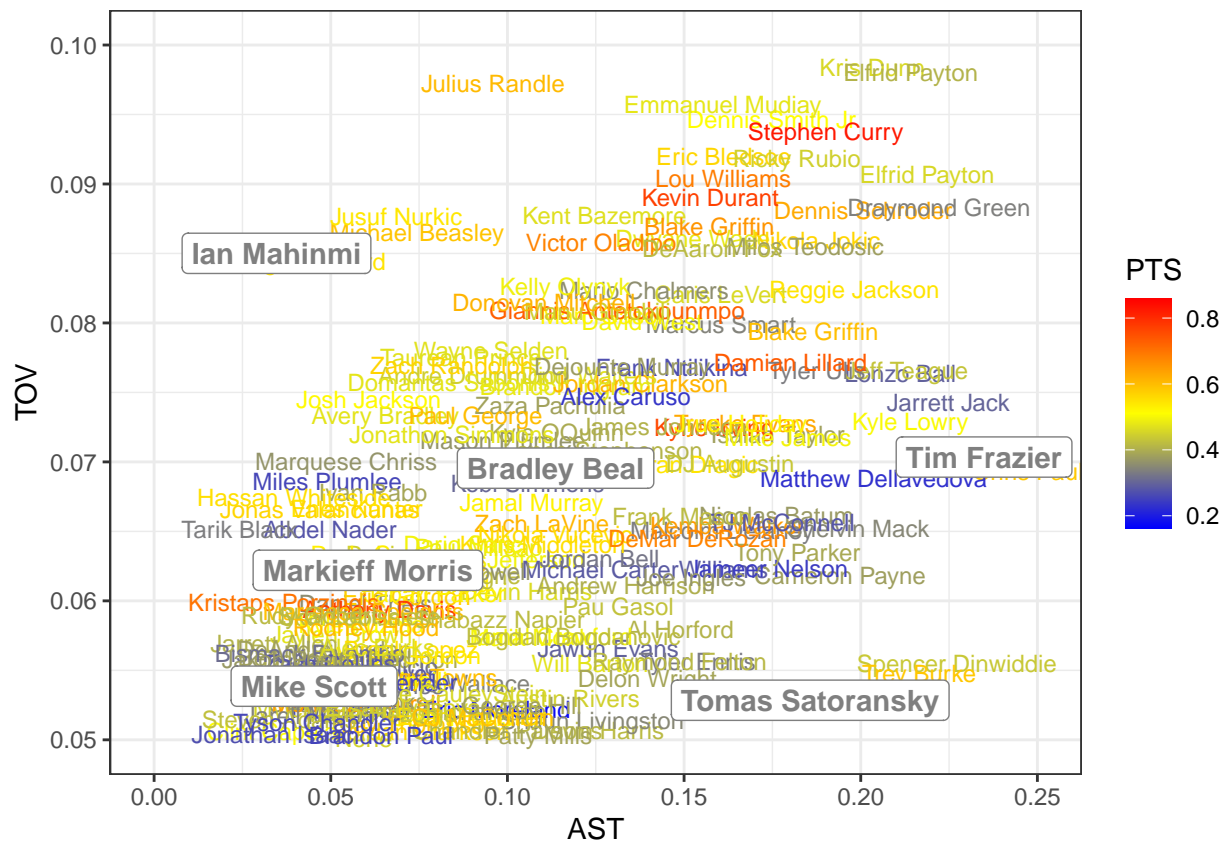




```
# Zoom the previous scatter plot
scatterplot(X3, data.var=c("AST","TOV"), z.var="PTS",
  labels=Pbox.sel$Player, palette=mypal,
  subset=WAS, zoom=c(0,0.25,0.05,0.10))
```

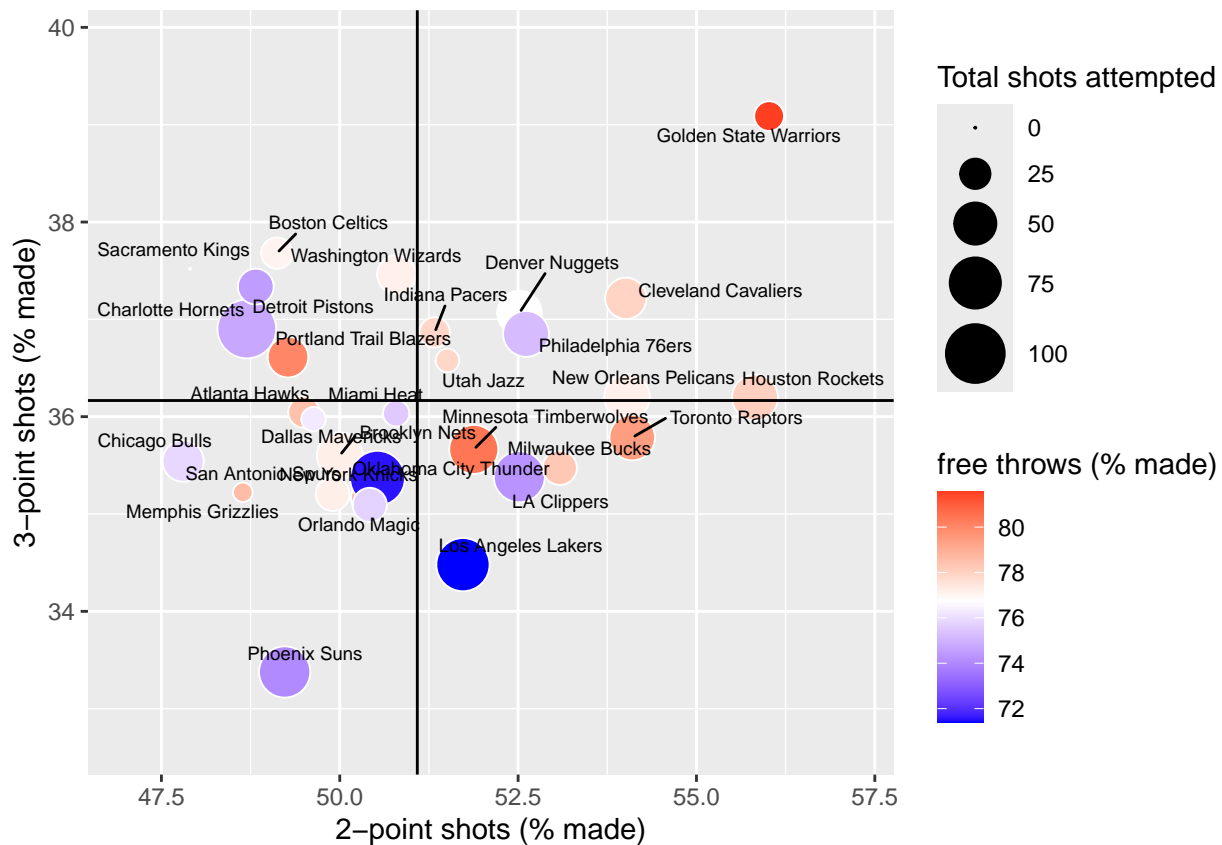
```
## Warning: Removed 169 rows containing missing values or values outside the scale range
## (`geom_text()`).
```

```
## Warning: Removed 5 rows containing missing values or values outside the scale range
## (`geom_label_repel()`).
```



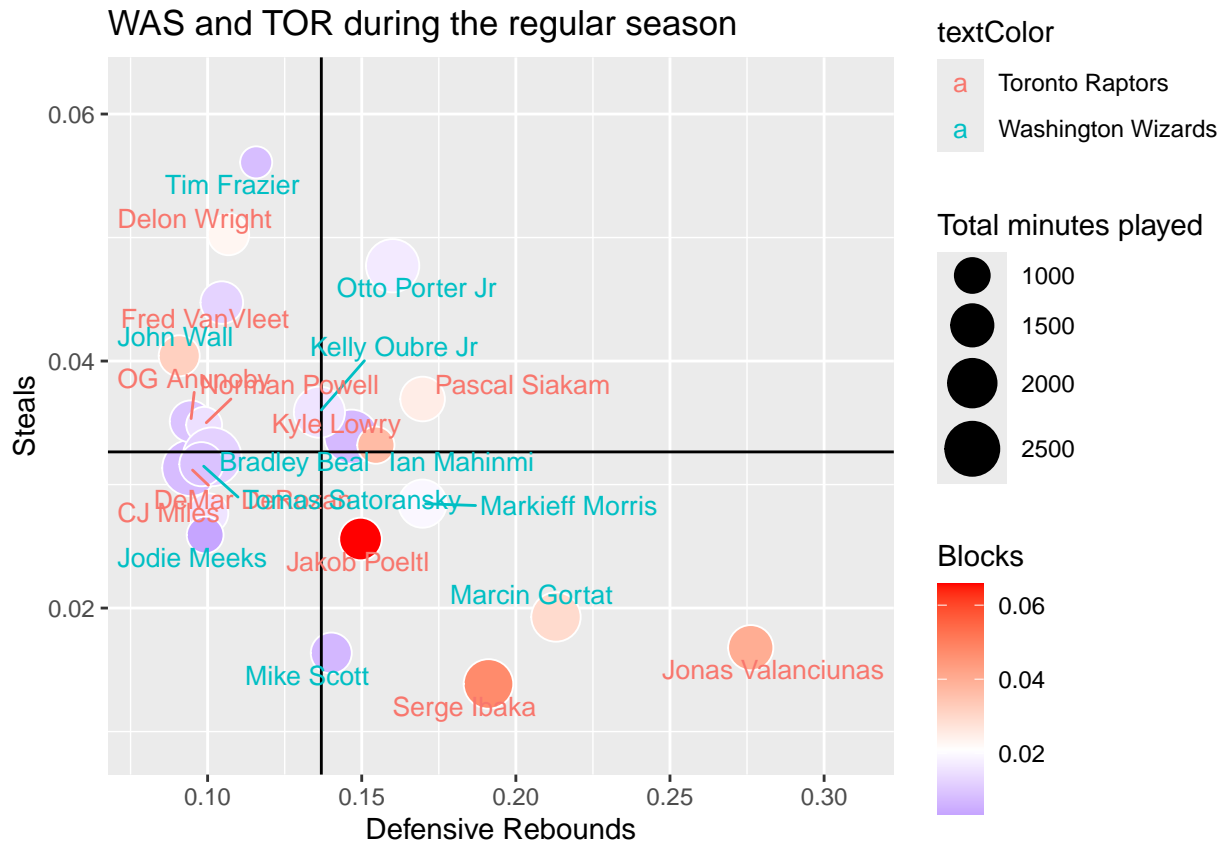
Bubble Plot

```
attach(Tbox)
X4 <- data.frame(T=Team, P2p, P3p, FTp, AS=P2A+P3A+FTA)
detach(Tbox)
labs <- c("2-point shots (% made)",
          "3-point shots (% made)",
          "free throws (% made)",
          "Total shots attempted")
bubbleplot(X4, id="T", x="P2p", y="P3p", col="FTp", size="AS", labels=labs)
```



```
# Defensive stats: WAS vs. TOR
Pbox.WAS.TOR <- subset(Pbox,
  (Team=="Washington Wizards" |
    Team=="Toronto Raptors") &
  MIN>=500)

attach(Pbox.WAS.TOR)
WAS.TOR <- data.frame(ID=Player, Team, V1=DREB/MIN, V2=STL/MIN,
  V3=BLK/MIN, V4=MIN)
detach(Pbox.WAS.TOR)
labs <- c("Defensive Rebounds", "Steals", "Blocks",
  "Total minutes played")
bubbleplot(WAS.TOR, id="ID", x="V1", y="V2", col="V3",
  size="V4", text.col="Team", labels=labs,
  title="WAS and TOR during the regular season",
  text.legend=TRUE, text.size=3.5, scale=FALSE)
```



## Variability analysis

```

vrb1 <- variability(data=Pbox.WW, data.var="P3p",
                    size.var="P3A")

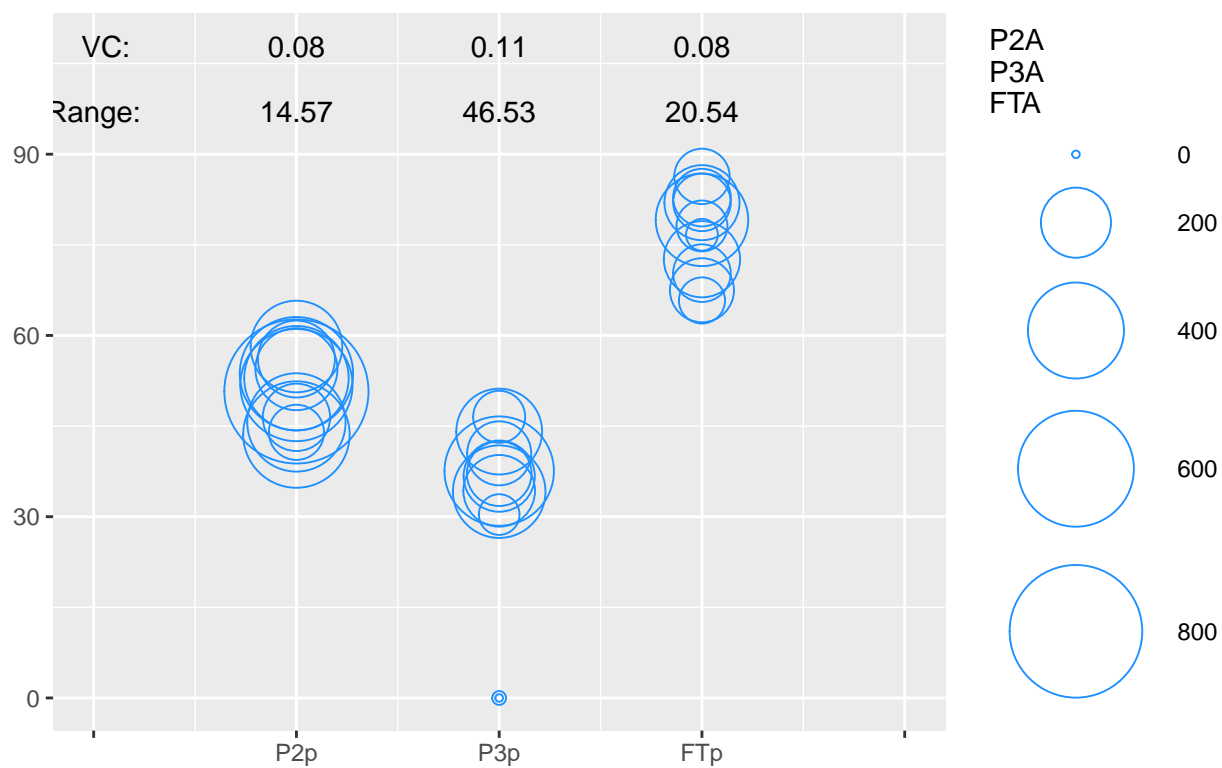
vrb1 <- variability(data=Pbox.WW, data.var="P3p",
                    size.var="P3A", weight=TRUE)

vrb2 <- variability(data=Pbox.WW,
                    data.var=c("P2p", "P3p", "FTp"),
                    size.var=c("P2A", "P3A", "FTA"),
                    weight=TRUE)

plot(vrb2, title="Variability diagram - WAS")

```

## Variability diagram – WAS



## Inequality analysis

```
ineqWW <- inequality(Pbox.WW$PTS, nplayers=8)

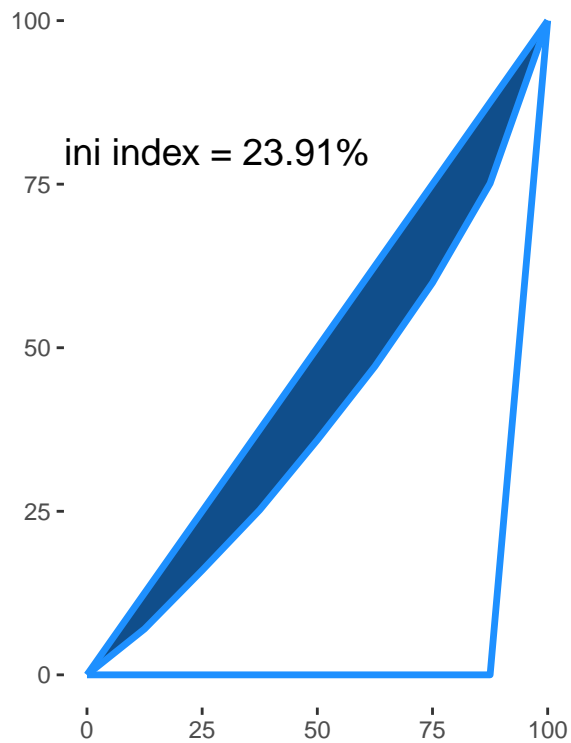
Pbox.BC <- subset(Pbox, Team == "Boston Celtics")
ineqBC <- inequality(Pbox.BC$PTS, nplayers=8)

p1 <- plot(ineqWW, title = "Washington Wizards")

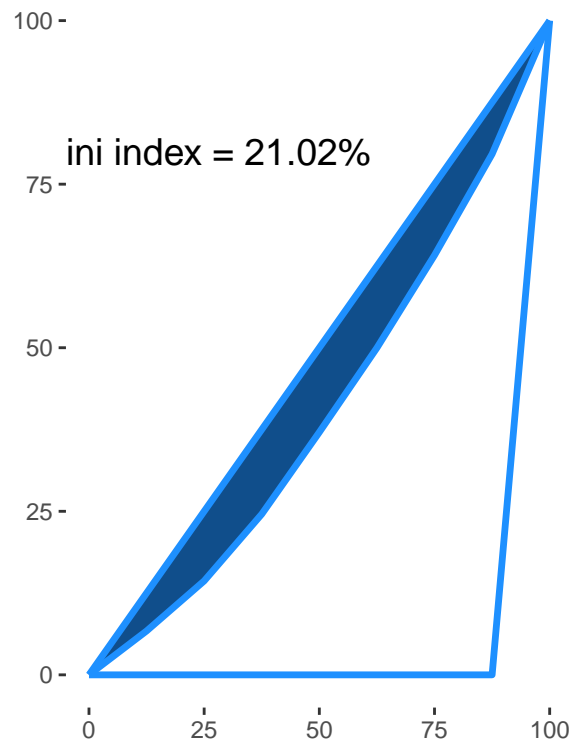
p2 <- plot(ineqBC, title = "Boston Celtics")

grid.arrange(p1, p2, nrow = 1)
```

Washington Wizards

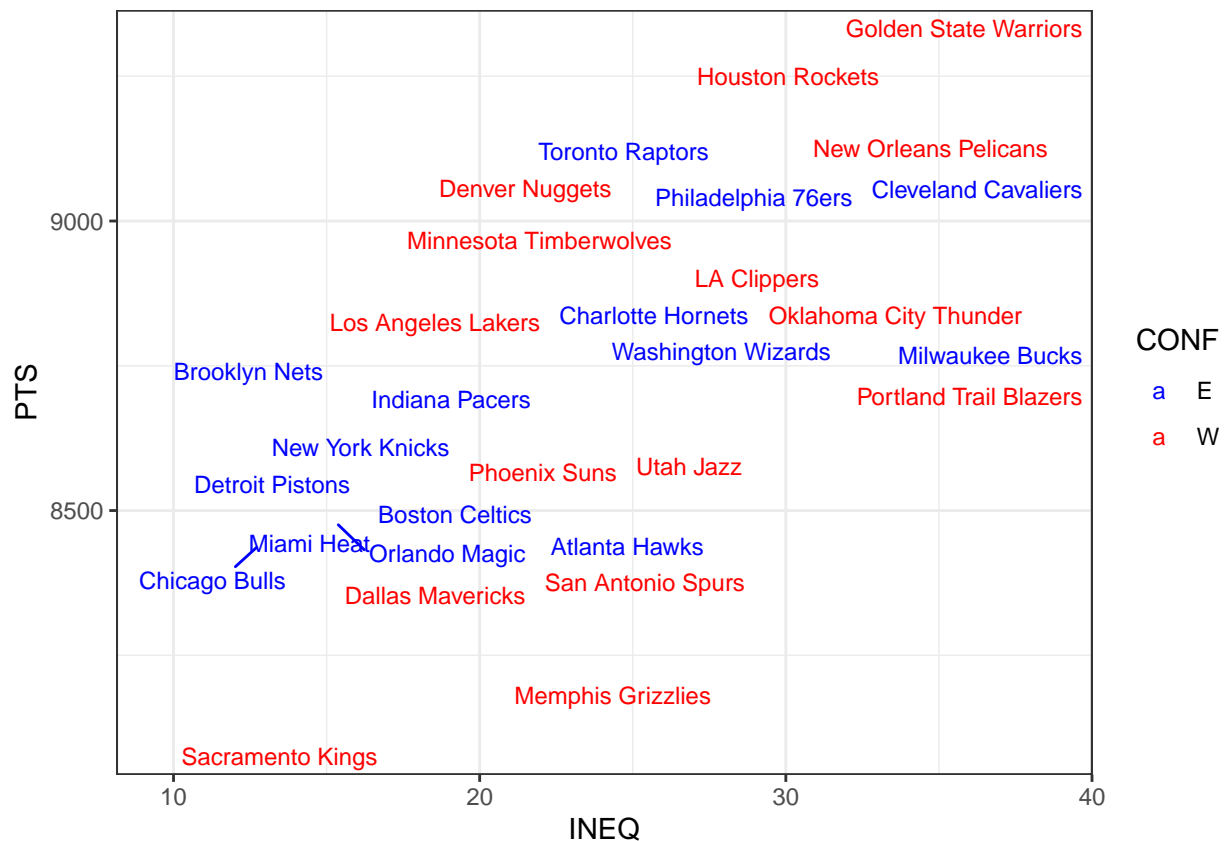


Boston Celtics



```
no.teams <- nrow(Tbox)
INEQ <- array(0, no.teams)
for (k in 1:no.teams) {
  Teamk <- Tbox$Team[k]
  Pbox.sel <- subset(Pbox, Team==Teamk)
  index <- inequality(Pbox.sel$PTS, npl=8)
  INEQ[k] <- index$Gini
}

dts <- data.frame(INEQ, PTS=Tbox$PTS,
                  CONF=Tadd$Conference)
mypal2 <- colorRampPalette(c("blue", "red"))
scatterplot(dts, data.var=c("INEQ", "PTS"), z.var="CONF",
            labels=Tbox$Team, palette=mypal,
            repel_labels=TRUE)
```



```
PbP <- PbPmanipulation(PbP.BDB)

PbP.WAS <- subset(PbP, team == "WAS")

lineup <- c("Bradley Beal", "John Wall", "Otto Porter Jr.", "Kelly Oubre Jr.",
            "Marcin Gortat")

filt5 <- apply(PbP.WAS[, 4:13], 1,
               function(x) {
                 x <- as.character(x)
                 sum(x %in% lineup)==5
               })

subPbP.WAS <- PbP.WAS[filt5, ]

PTS5 <- sapply(lineup,
               function(x) {
                 filt <- subPbP.WAS$player==x
                 sum(subPbP.WAS$points[filt], na.rm=T)
               })

inequality(PTS5,nplayer=5)

## $Gini
## [1] 34.72
##
## $Lorenz
```

```
##           F           Q
##       0.0 0.00000000
## John Wall    0.2 0.08333333
## Bradley Beal 0.4 0.19444444
## Kelly Oubre Jr. 0.6 0.36111111
## Otto Porter Jr. 0.8 0.66666667
## Marcin Gortat 1.0 1.00000000
##
## attr("class")
## [1] "inequality" "list"

PbP.WAS.TOR <- subset(PbP, team=="WAS" & oppTeam=="TOR")

filt5.2 <- apply(PbP.WAS.TOR[, 4:13], 1,
  function(x) {
    x <- as.character(x)
    sum(x %in% lineup)==5
  })

subPbP.WAS.TOR <- PbP.WAS.TOR[filt5.2, ]

PTS5 <- sapply(lineup,
  function(x) {
    filt <- subPbP.WAS.TOR$player==x
    sum(subPbP.WAS.TOR$points[filt], na.rm=T)
  })

inequality(PTS5,nplayer=5)

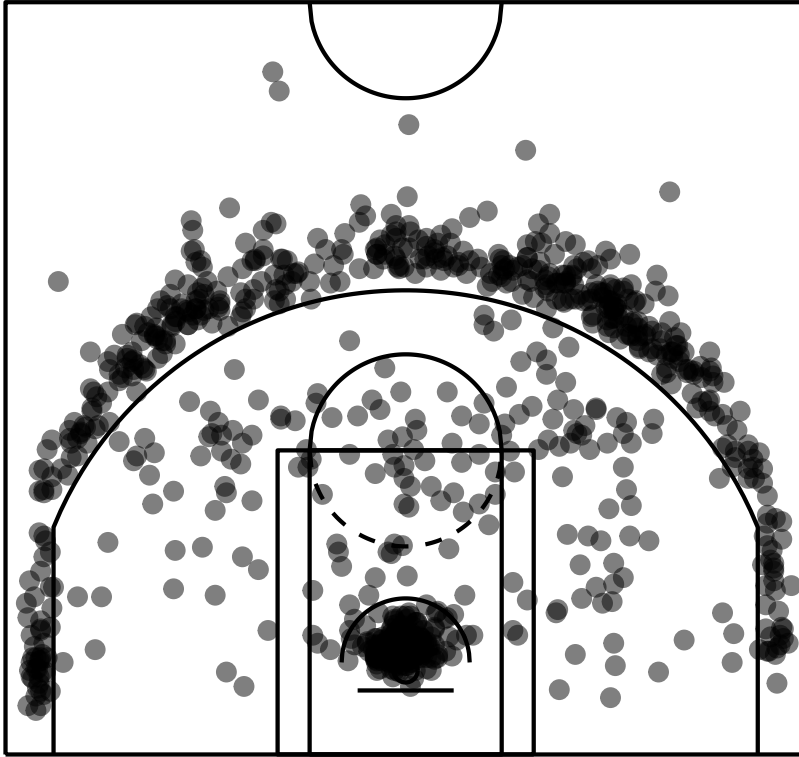
## $Gini
## [1] NaN
##
## $Lorenz
##           F           Q
##       0.0           0
## Marcin Gortat 0.2 NaN
## Kelly Oubre Jr. 0.4 NaN
## Otto Porter Jr. 0.6 NaN
## John Wall      0.8 NaN
## Bradley Beal   1.0 NaN
##
## attr("class")
## [1] "inequality" "list"
```

## Shot charts

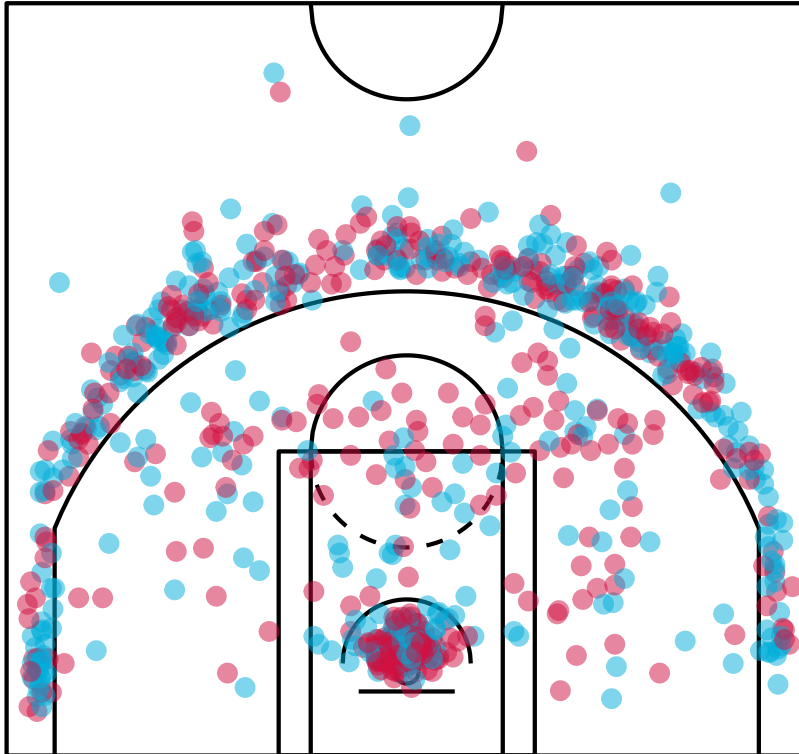
```
#unique(PbP$player)
subdata <- subset(PbP, player=="Stephen Curry")
subdata$xx <- subdata$original_x/10
subdata$yy <- subdata$original_y/10-41.75

shotchart(data=subdata, x="xx", y="yy", type=NULL,
  scatter=TRUE)
```





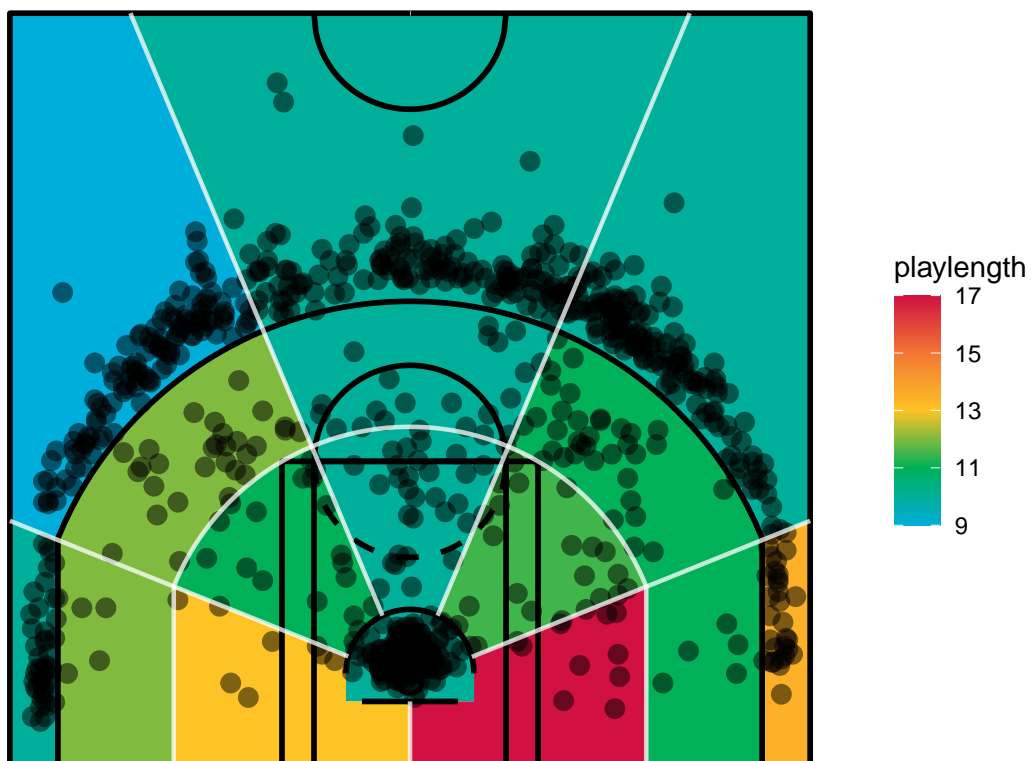
```
shotchart(data=subdata, x="xx", y="yy", z="result", type=NULL,
          scatter=TRUE)
```



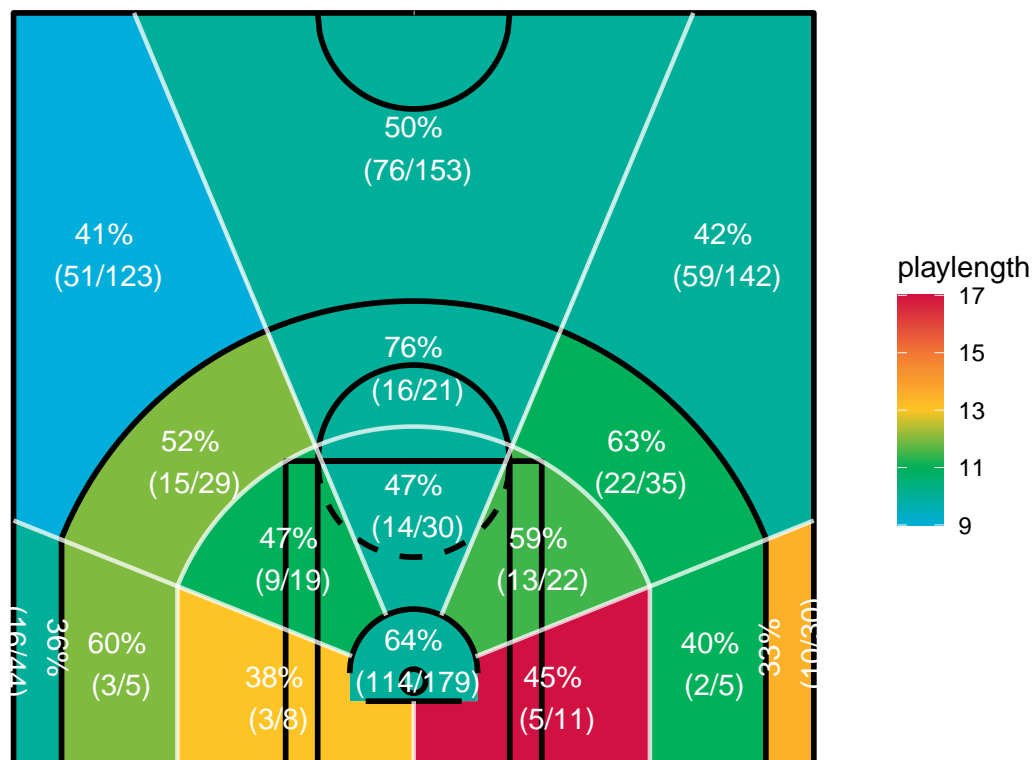
result

- made
- missed

```
shotchart(data=subdata, x="xx", y="yy", z="playlength",
          num.sect=5, type="sectors", scatter = TRUE)
```



```
shotchart(data=subdata, x="xx", y="yy", z="playlength",
          num.sect=5, type="sectors", scatter=FALSE, result="result")
```



# PATTERN IN DATA

## Statistical dependence

```
PbP <- PbPmanipulation(PbP.BDB)

PbP.GSW <- subset(PbP, team=="GSW")

ev <- c("ejection","end of period","jump ball",
       "start of period","unknown","violation",
       "timeout","sub","foul","turnover")

event.unsel <- which(PbP.GSW$event_type %in% ev)

PbP.GSW.ev <- PbP.GSW[-event.unsel,]

attach(PbP.GSW.ev)
T <- table(oppTeam, event_type, exclude=ev)
detach(PbP.GSW.ev)

assocstats(T)
```

```
##                X^2 df P(> X^2)
## Likelihood Ratio 115.26 84 0.013396
## Pearson          116.25 84 0.011421
##
## Phi-Coefficient   : NA
## Contingency Coeff.: 0.097
## Cramer's V        : 0.056
```

## Mean Dependence

```
FF <- fourfactors(Tbox, Obox)
attach(Tbox)
attach(FF)

## The following object is masked from Tbox:
##
##      Team

df_stats <- data.frame(PTS, P2M, P3M, FTM, REB=OREB+DREB, AST, STL, BLK, ORtg, DRtg)
detach(Tbox)
detach(FF)

Playoff <- Tadd$Playoff

eta <- sapply(df_stats, function(Y){
  cm <- round(tapply(Y, Playoff, mean), 1)
  eta2 <- etaSquared(aov(Y~Playoff))[1]*100
  c(cm, round(eta2, 2))
}) %>%
  t() %>%
  as.data.frame() %>%
  rename(No=N, Yes=Y, eta2=V3) %>%
  rownames_to_column('rownm') %>%
```

```
arrange(-eta2) %>%
column_to_rownames('rownm')
```

```
eta
```

```
##           No      Yes eta2
## DRtg  107.9  104.6 42.53
## ORtg  104.0  108.1 40.25
## STL   601.9  659.6 28.77
## PTS  8576.0 8844.8 19.28
## BLK   365.6  420.4 18.12
## FTM  1328.0 1394.4  5.58
## P2M  2353.7 2417.2  3.28
## AST  1875.5 1931.6  3.17
## P3M   846.9  871.9  1.07
## REB  3558.1 3577.5  0.49
```

## Correlation

```
df_ast_tov <- subset(Pbox, MIN>=500)
attach(df_ast_tov)
ast_tov <- data.frame(AST, TOV)/MIN
detach(df_ast_tov)

cor(ast_tov$AST, ast_tov$TOV)
```

```
## [1] 0.6873883
```

```
^
```

```
cor(rank(ast_tov$AST), rank(ast_tov$TOV))
```

```
## [1] 0.6679628
```

```
cor(ast_tov$AST, ast_tov$TOV, method="spearman")
```

```
## [1] 0.6679628
```

```
cor(ast_tov)
```

```
##           AST      TOV
## AST 1.0000000 0.6873883
## TOV 0.6873883 1.0000000
```

## Linear correlation among variables

```
data <- merge(Pbox, Tadd, by="Team")
data <- subset(data, MIN >= 500)

attach(data)
```

```
## The following object is masked _by_ .GlobalEnv:
```

```
##
```

```
##      Playoff
```

```
df_off <- data.frame(
  Player = data$Player,
```

```

PTS = data$PTS / data$MIN,
P3M = data$P3M / data$MIN,
P2M = data$P2M / data$MIN,
REB = (data$OREB + data$DREB) / data$MIN,
AST = data$AST / data$MIN,
TOV = data$TOV / data$MIN,
STL = data$STL / data$MIN,
BLK = data$BLK / data$MIN,
Playoff = data$Playoff
)
detach(data)

str(df_off)

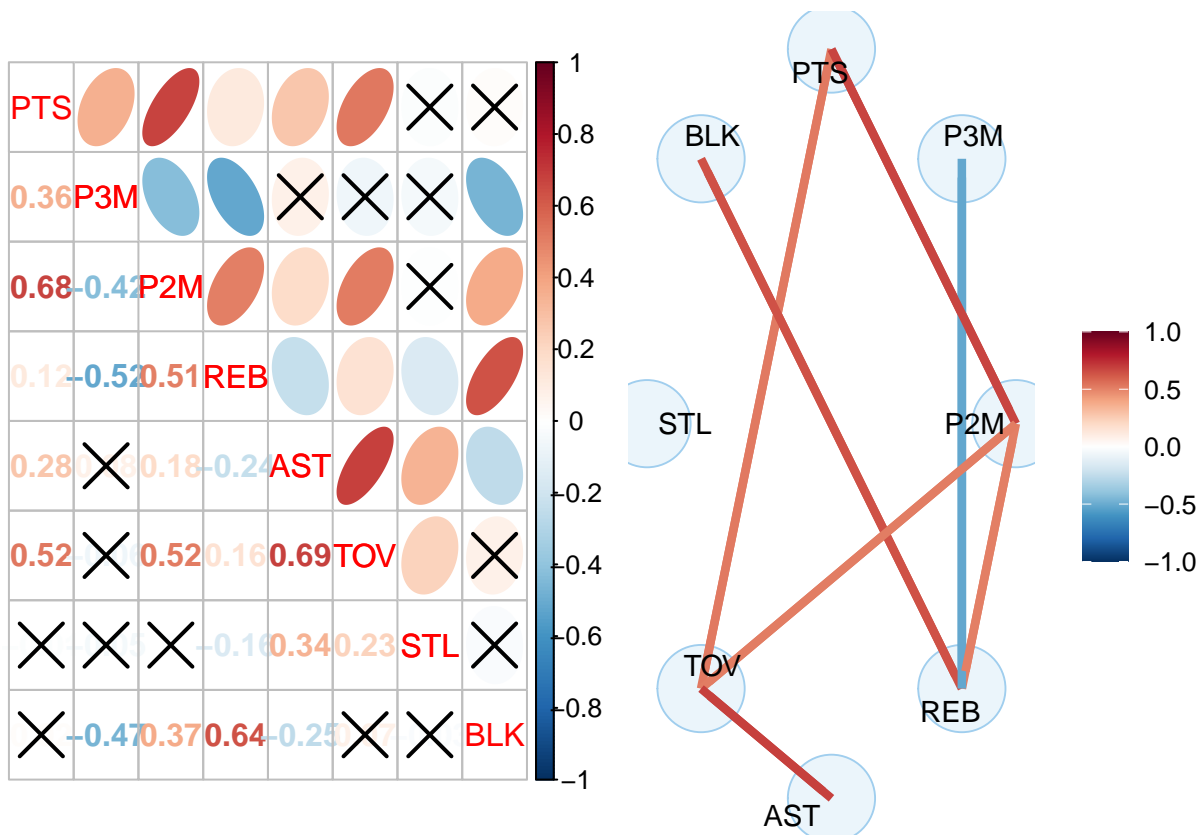
## 'data.frame': 361 obs. of 10 variables:
## $ Player : chr "Taurean Prince" "Dennis Schroder" "Kent Bazemore" "John Collins" ...
## $ PTS : num 0.47 0.626 0.467 0.435 0.4 ...
## $ P3M : num 0.07143 0.03657 0.06037 0.00896 0.03243 ...
## $ P2M : num 0.097 0.204 0.0984 0.1669 0.1297 ...
## $ REB : num 0.158 0.1 0.139 0.303 0.317 ...
## $ AST : num 0.0869 0.2007 0.1274 0.0549 0.0584 ...
## $ TOV : num 0.0775 0.0881 0.0878 0.0588 0.0558 ...
## $ STL : num 0.0345 0.0346 0.0559 0.0263 0.0259 ...
## $ BLK : num 0.01705 0.00289 0.02515 0.04482 0.03307 ...
## $ Playoff: Factor w/ 2 levels "N","Y": 1 1 1 1 1 1 1 1 1 1 ...

numeric_columns <- sapply(df_off, is.numeric)

corrmatrix <- corranalysis(df_off[,2:9], threshold=0.5)
plot(corrmatrix)

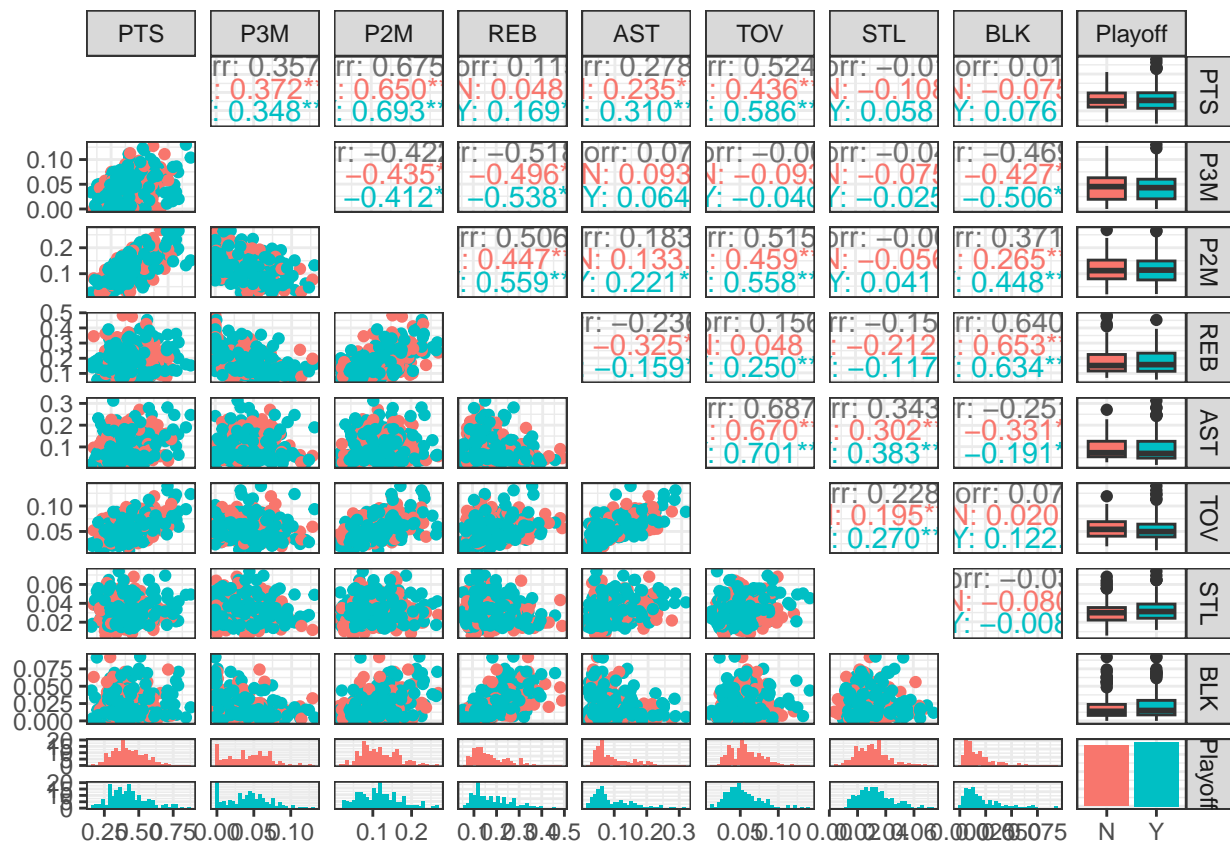
## Warning: The `guide` argument in `scale_*()` cannot be `FALSE`. This was deprecated in
## ggplot2 3.3.4.
## i Please use "none" instead.
## i The deprecated feature was likely used in the BasketballAnalyzeR package.
## Please report the issue to the authors.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

```

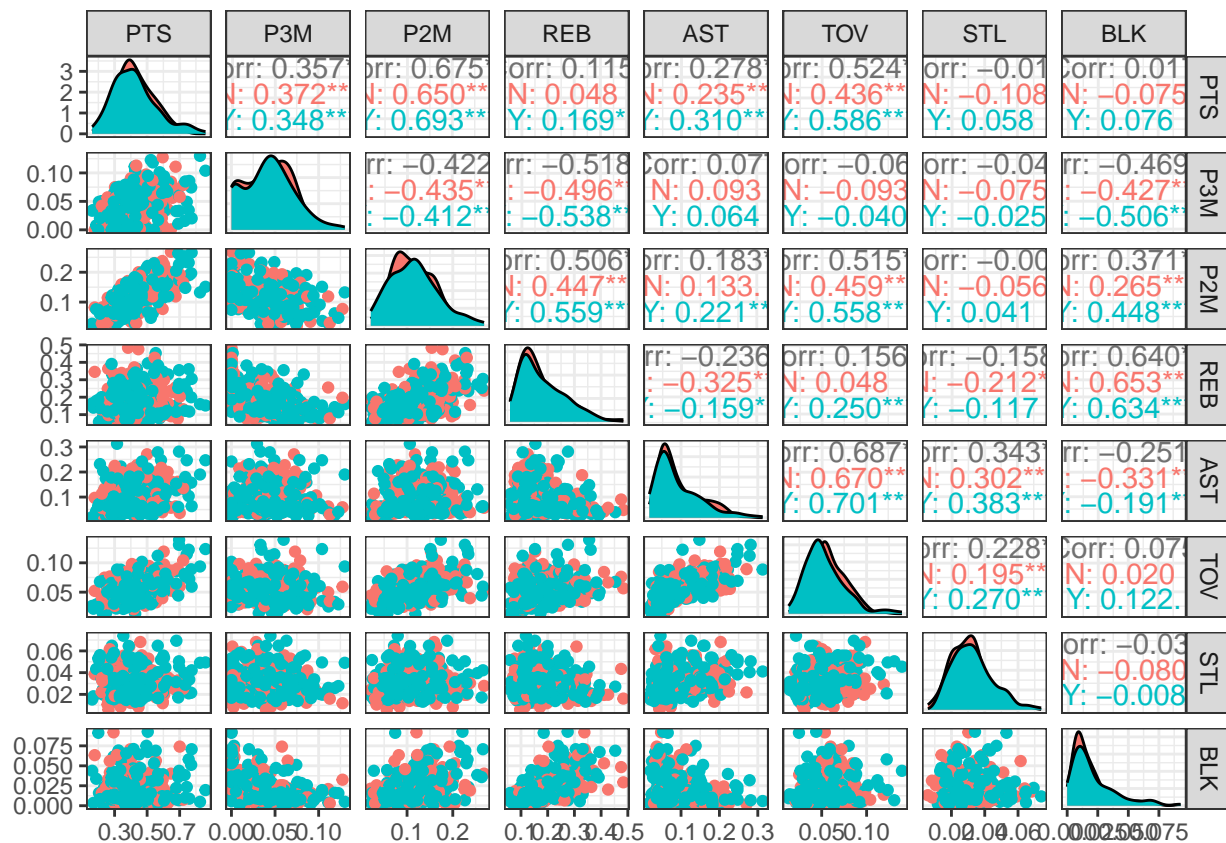


```
scatterplot(df_off, data.var=2:9, z.var="Playoff",
            diag=list(continuous="blankDiag"))
```

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



```
ggpairs(df_off, columns = 2:9, mapping = aes(color = Playoff)) +
  theme_bw()
```



## Individual cases according to their similarity

```
attach(Pbox)
data2 <- data.frame(PTS, P3M, P2M, REB=OREB+DREB,
                    AST, TOV, STL, BLK)
detach(Pbox)
data2 <- subset(data2, Pbox$MIN>=1500)
id <- Pbox$Player[Pbox$MIN>=1500]
```

```
# Multidimensional scaling
mds <- MDSmap(data2)
```

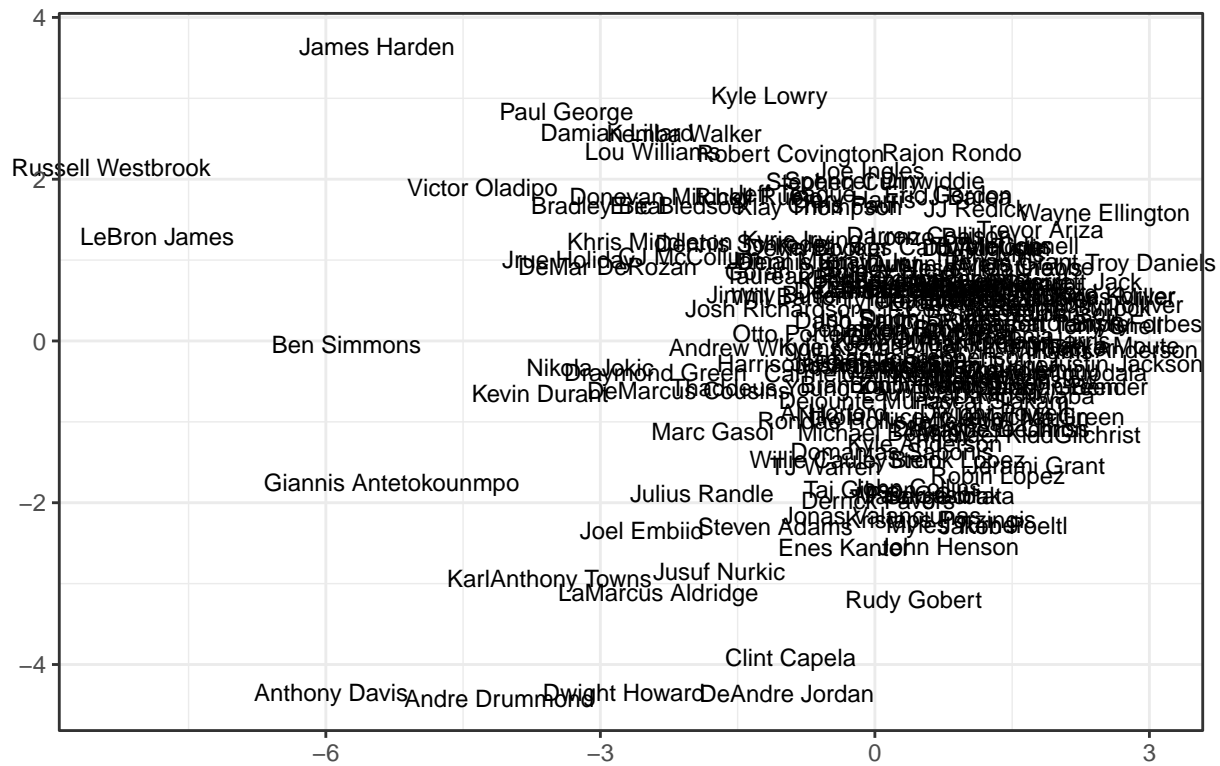
```
## initial value 15.910514
## iter 5 value 13.124944
## final value 12.967089
## converged
```

```
plot(mds, labels=id)
```



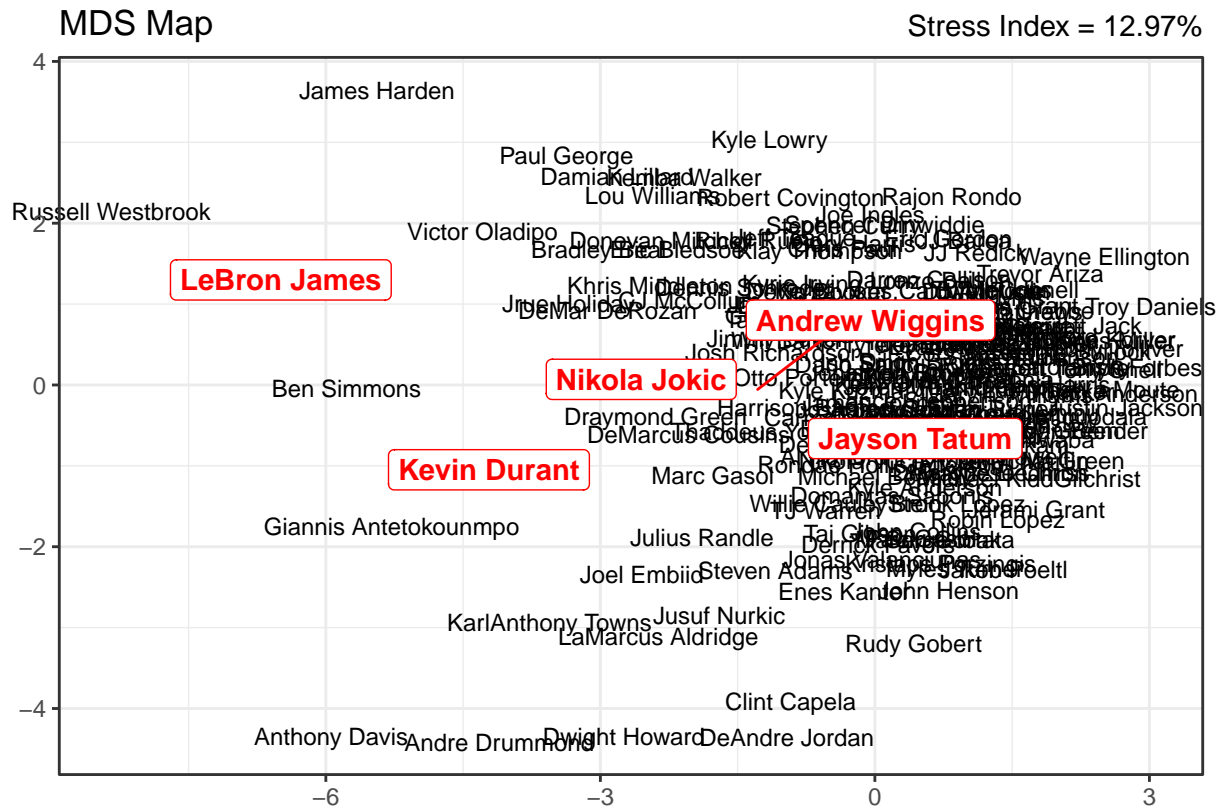
MDS Map

Stress Index = 12.97%



```
selp <- which(id=="LeBron James" | id=="Kevin Durant" |
              id=="Jayson Tatum" | id=="Nikola Jokic" |
              id=="Andrew Wiggins")

plot(mds, labels=id, subset=selp, col.subset="red")
```



## Network Relationship

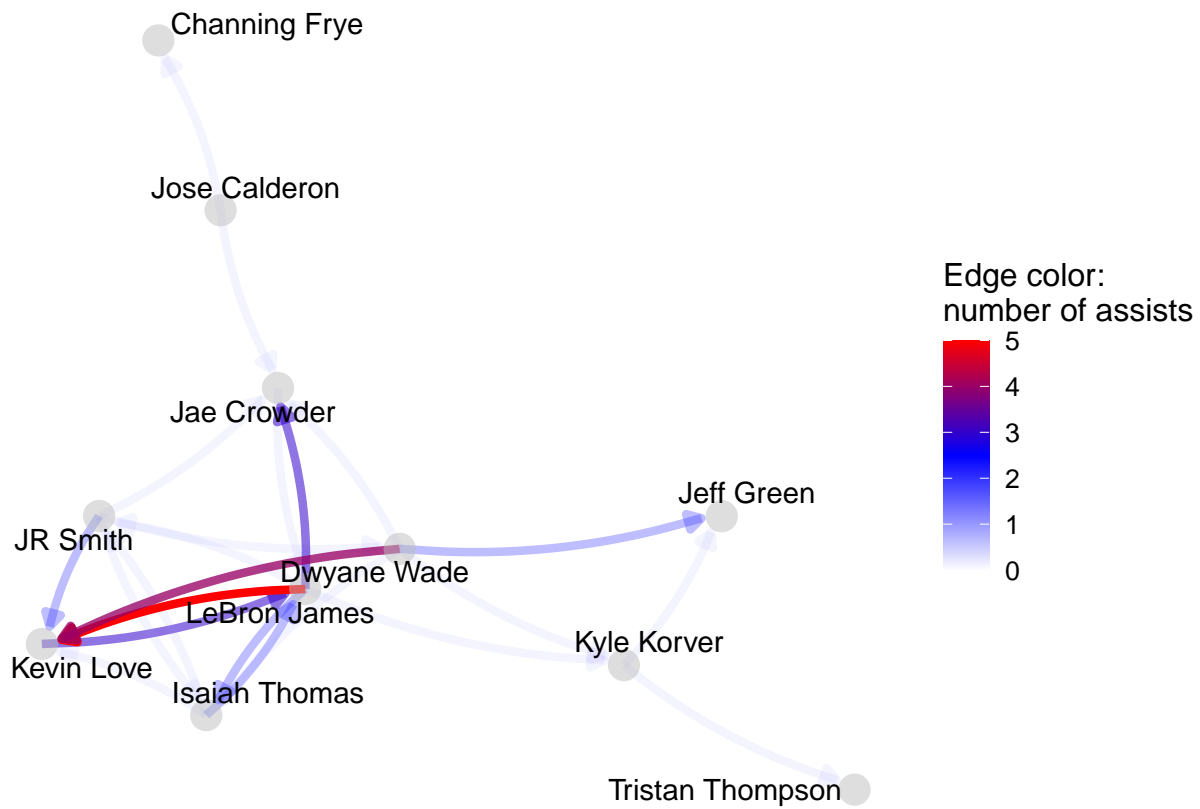
### Cleveland

```
PbP <- PbPmanipulation(PbP.BDB)

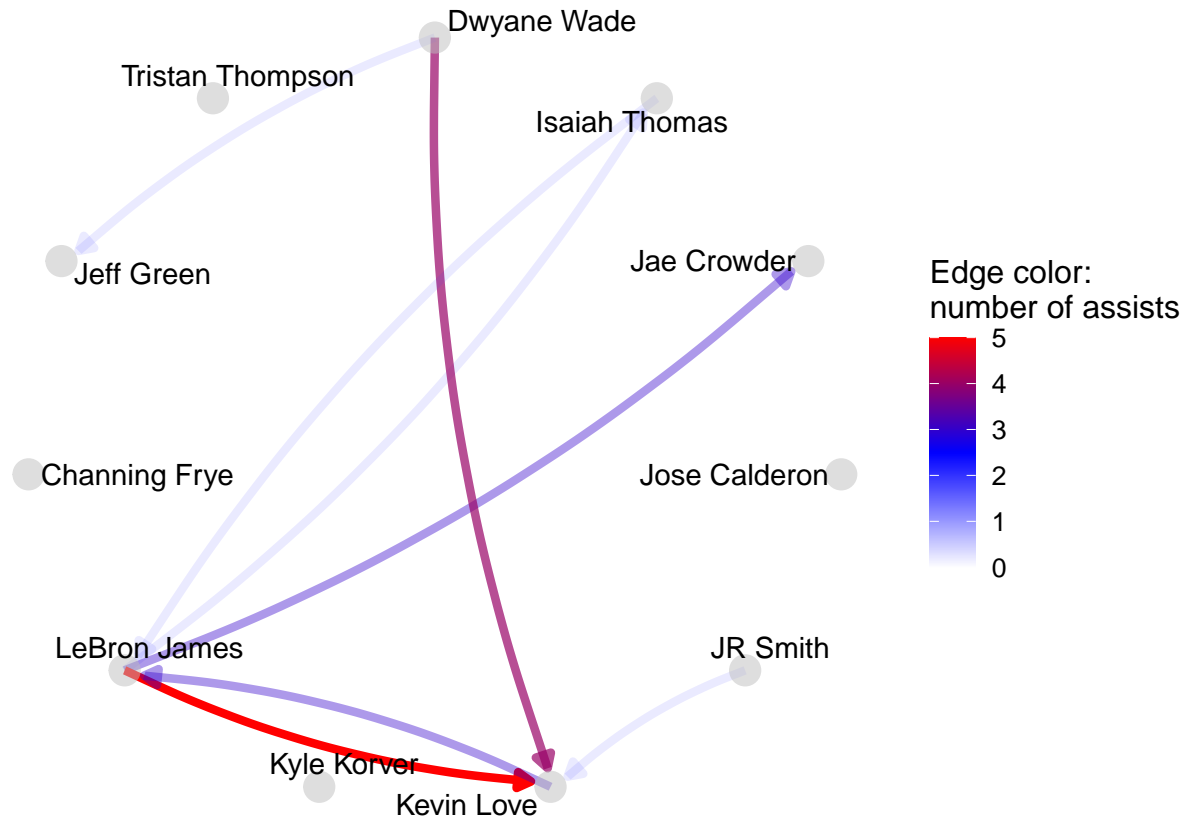
PbP.CLE <- subset(PbP, team=="CLE")

netdata <- assistnet(PbP.CLE)

set.seed(7)
plot(netdata)
```



```
plot(netdata, layout="circle", edge.thr=1)
```



```

#Golden State Warriors
PbP.GSW <- subset(PbP, team=="GSW")

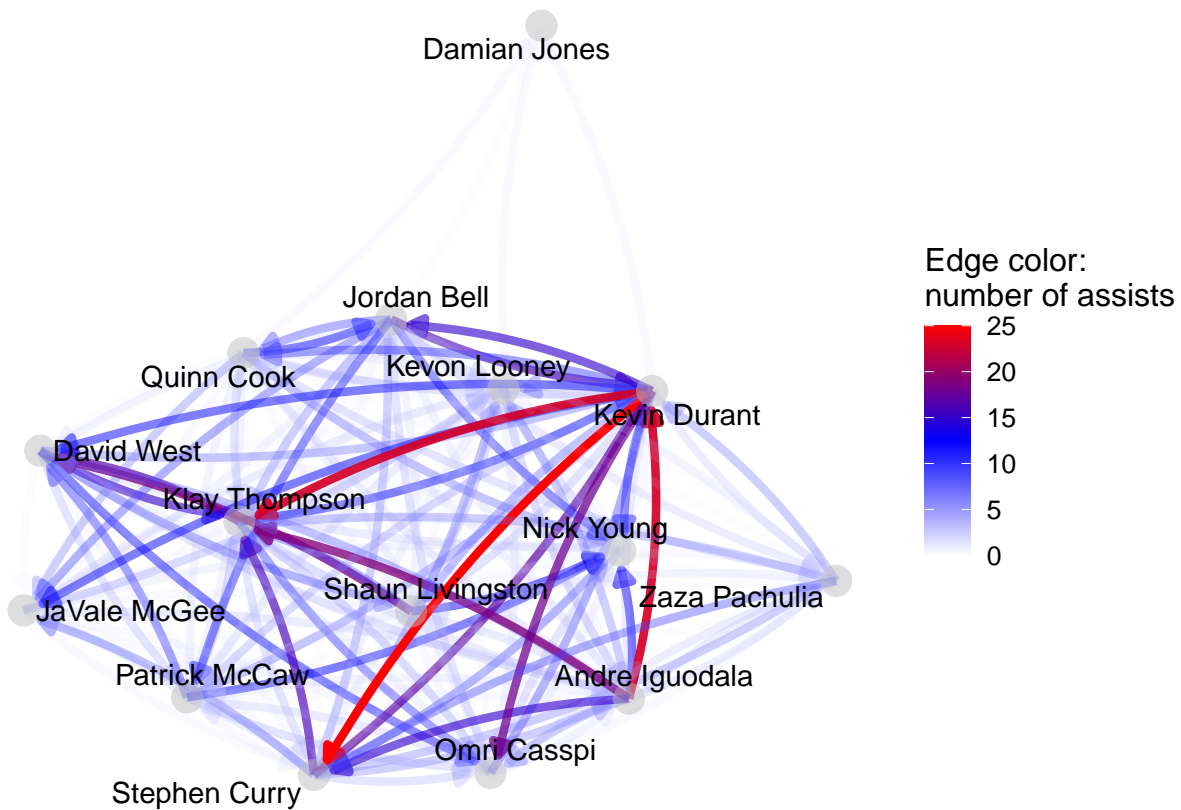
netdataGSW <- assistnet(PbP.GSW)

cols <- paste0(c("a","h"), rep(1:5,each=2))

PbP.GSW.DG0 <- PbP.GSW[!apply(PbP.GSW[,cols], 1, "%in%",
                             x="Draymond Green"),]
netdata.DG0 <- assistnet(PbP.GSW.DG0)

set.seed(1)
plot(netdata.DG0)

```



```

PbP.GSW.DG0 <- subset(PbP.GSW.DG0,
                     ShotType=="2P" | ShotType=="3P")

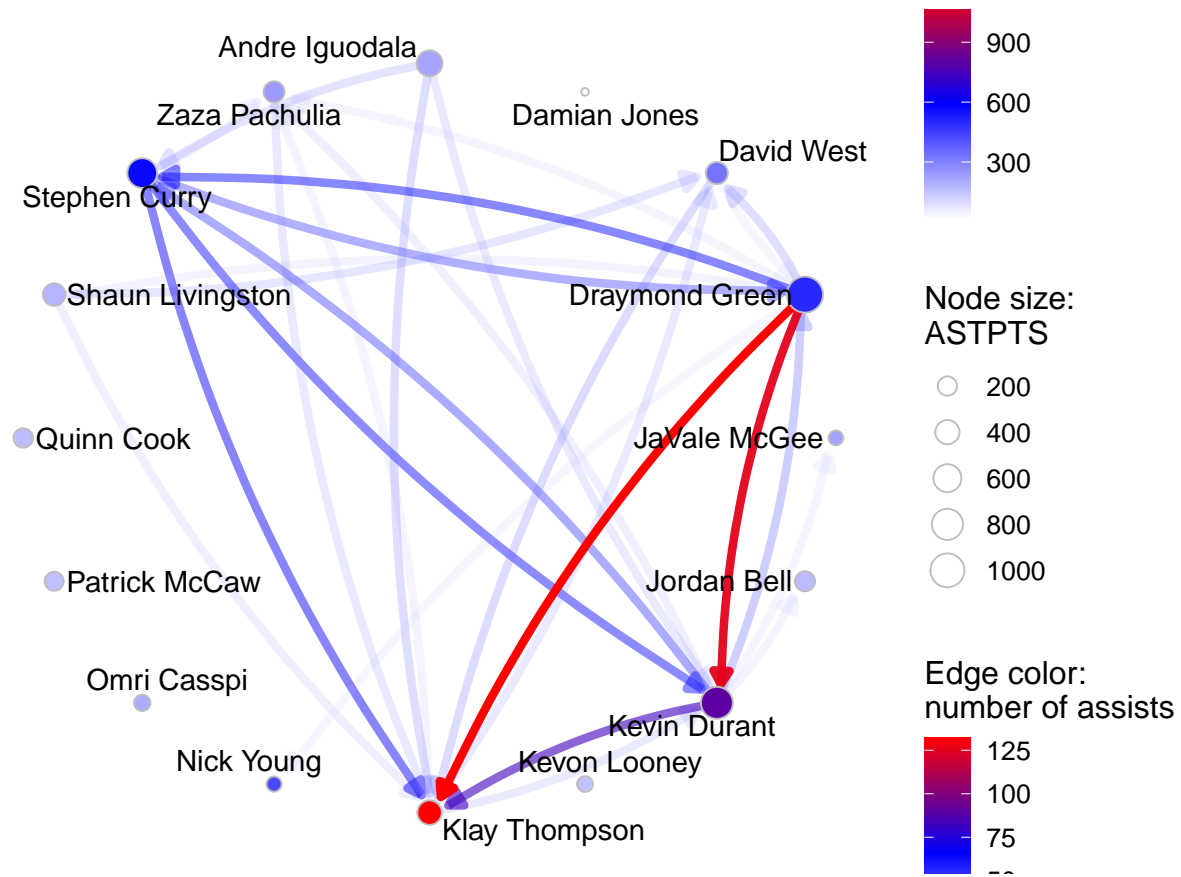
p0 <- mean(PbP.GSW.DG0$points)
pl0 <- mean(PbP.GSW.DG0$playlength)

PbP.GSW.DG1 <- PbP.GSW[apply(PbP.GSW[,cols], 1, "%in%",
                             x="Draymond Green"),]
PbP.GSW.DG1 <- subset(PbP.GSW.DG1,
                     ShotType=="2P" | ShotType=="3P")

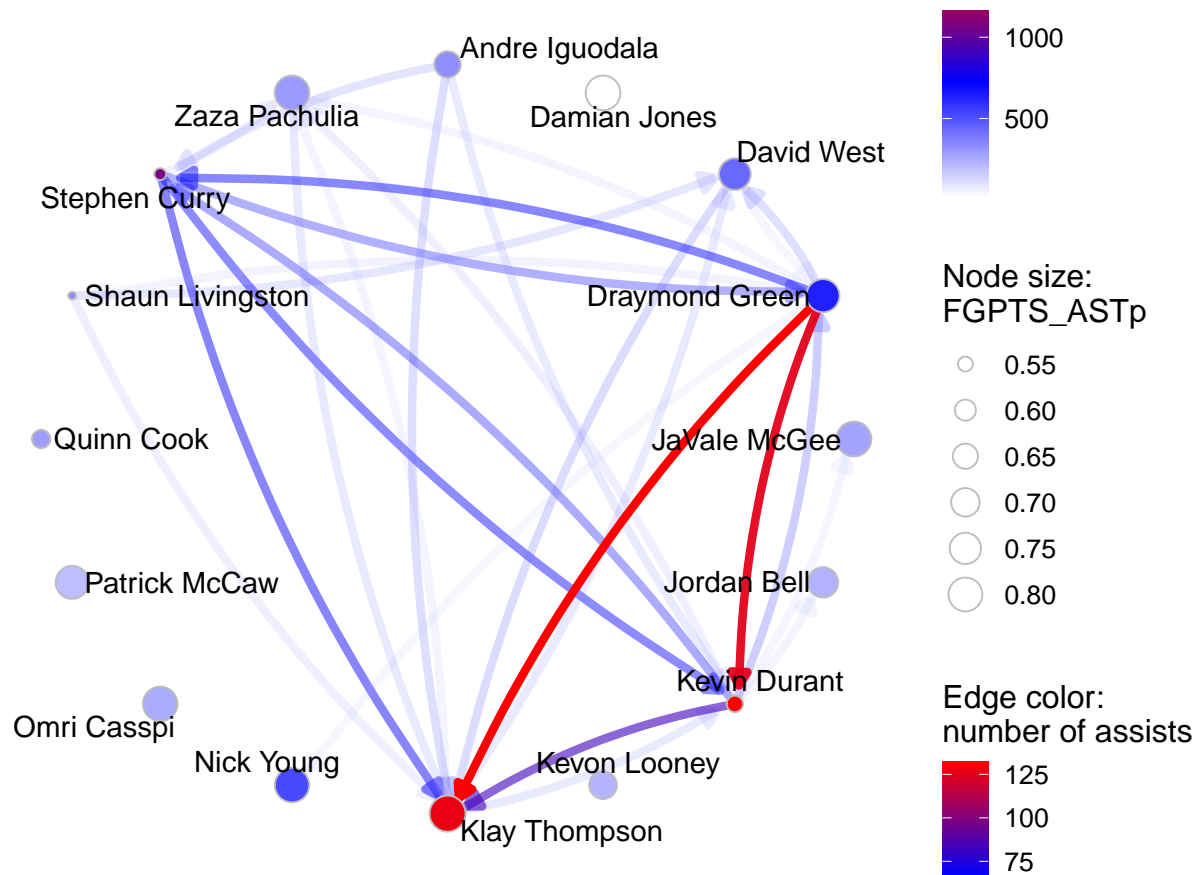
p1 <- mean(PbP.GSW.DG1$points)
pl1 <- mean(PbP.GSW.DG1$playlength)

```

```
plot(netdataGSW, layout="circle", edge.thr=20,
     node.col="FGPTS_AST", node.size="ASTPTS")
```



```
plot(netdataGSW, layout="circle", edge.thr=20,
     node.col="FGPTS", node.size="FGPTS_ASTp")
```



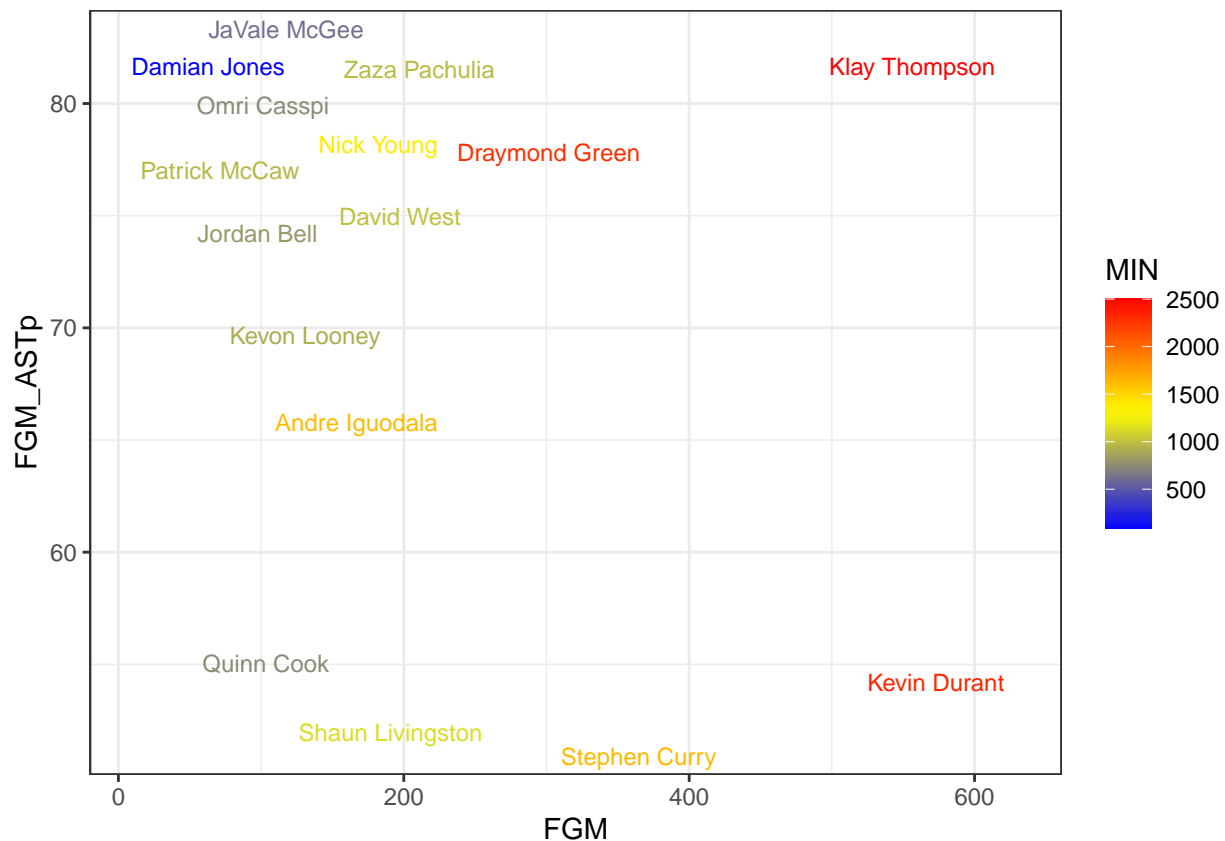
```
TAB <- netdataGSW$assistTable
N <- netdataGSW$nodeStats

names(N)[1] <- "Player"

data3 <- merge(N, Pbox, by="Player")

mypal <- colorRampPalette(c("blue", "yellow", "red"))

scatterplot(data3, data.var=c("FGM", "FGM_ASTp"),
            z.var="MIN", labels=data3$Player,
            palette=mypal, repel_labels=TRUE)
```



```

sel <- which(data3$MIN > 984)
print(sel)

## [1] 1 3 4 7 9 10 14 15

dim(TAB)

## [1] 16 16

tab <- TAB[sel, sel]

no.pl <- nrow(tab)

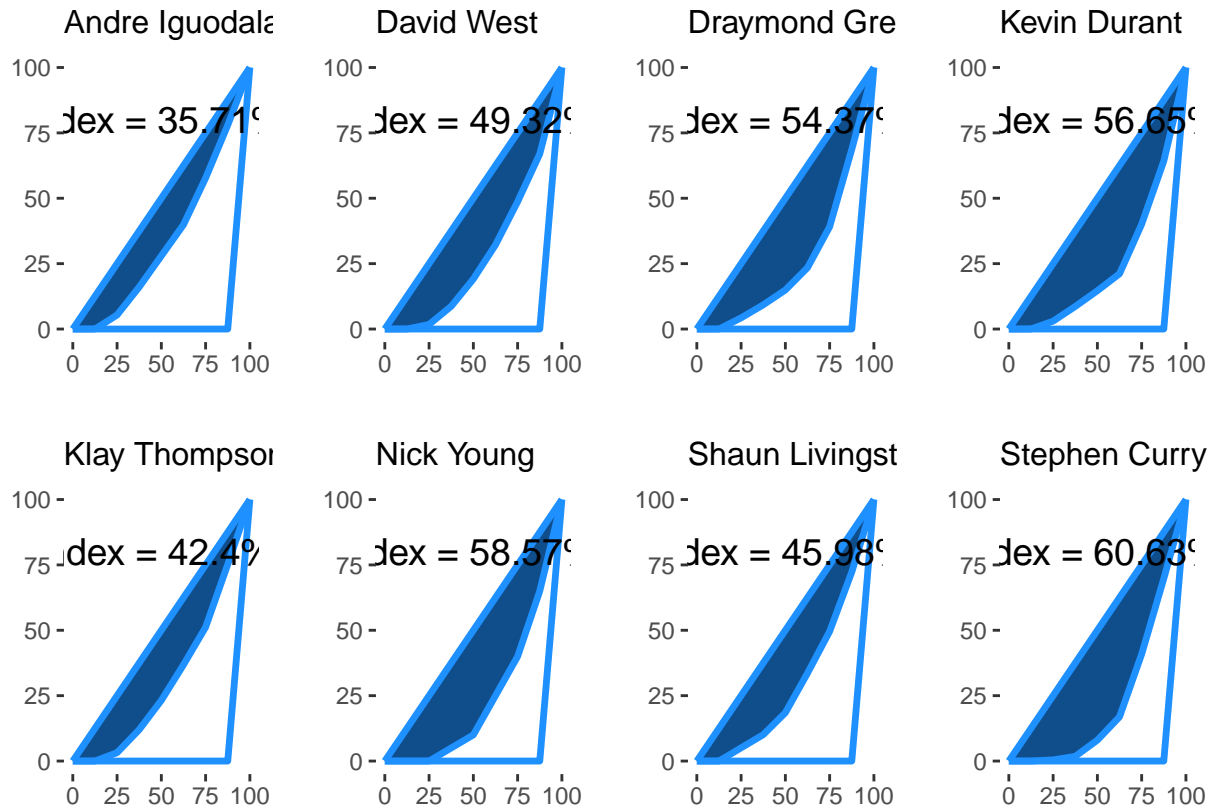
pR <- pM <- vector(no.pl, mode="list")

GiniM <- array(NA, no.pl)
GiniR <- array(NA, no.pl)

for (pl in 1:no.pl) {
  ineqplM <- inequality(tab[pl,], npl=no.pl)
  GiniM[pl] <- ineqplM$Gini
  ineqplR <- inequality(tab[,pl], npl=no.pl)
  GiniR[pl] <- ineqplR$Gini
  title <- rownames(tab)[pl]
  pM[[pl]] <- plot(ineqplM, title=title)
  pR[[pl]] <- plot(ineqplR, title=title)
}

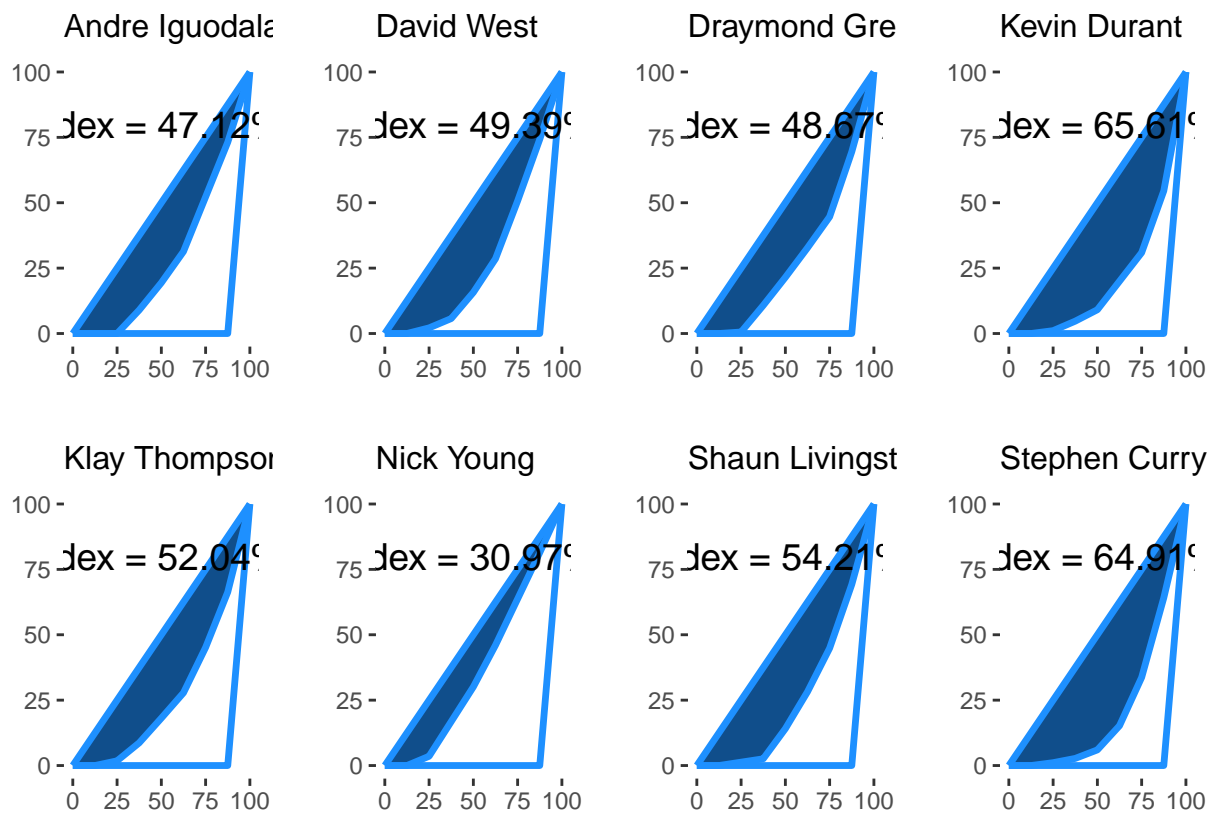
```

```
grid.arrange(grobs=pM, nrow=2)
```



```
grid.arrange(grobs=pR, nrow=2)
```





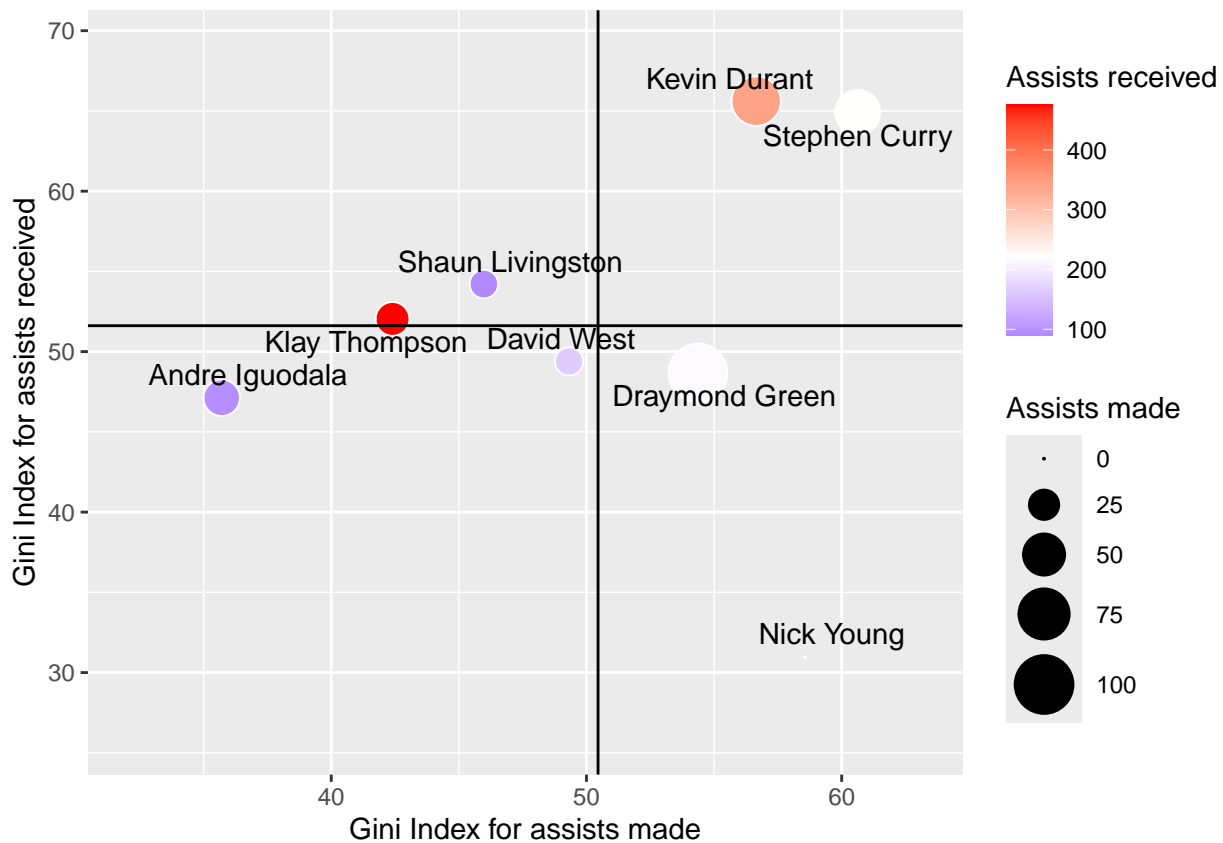
```
assocstats(tab)
```

```
##                X^2 df P(> X^2)
## Likelihood Ratio 670.48 49      0
## Pearson          507.67 49      0
##
## Phi-Coefficient   : NA
## Contingency Coeff.: 0.512
## Cramer's V        : 0.226
```

```
XX <- data.frame(N[sel,], GiniM, GiniR)
```

```
labs <- c("Gini Index for assists made",
          "Gini Index for assists received",
          "Assists received", "Assists made")
```

```
bubbleplot(XX, id="Player", x="GiniM", y="GiniR",
           col="FGM_AST", size="AST",
           labels=labs, text.size=4)
```



```
net1 <- as_tbl_graph(netdata$assistNet)
class(net1) <- "igraph"
centr_degree(net1)
```

```
## $res
## [1] 6 6 5 2 6 5 4 8 1 2 1
##
## $centralization
## [1] 0.21
##
## $theoretical_max
## [1] 200
```

```
alpha_centrality(net1)
```

```
##      Dwyane Wade    Isaiah Thomas    Jae Crowder    Jose Calderon
##      0.15384615     -0.76923077      0.23076923      1.00000000
##      JR Smith       Kevin Love       Kyle Korver      LeBron James
##      -0.84615385     -1.53846154      0.07692308      -1.07692308
##      Channing Frye   Jeff Green   Tristan Thompson
##      2.00000000      1.23076923      1.07692308
```

```
closeness(net1, mode="all")
```

```
##      Dwyane Wade    Isaiah Thomas    Jae Crowder    Jose Calderon
##      0.06666667      0.05000000      0.05882353      0.04166667
##      JR Smith       Kevin Love       Kyle Korver      LeBron James
##      0.05882353      0.05000000      0.05263158      0.06250000
##      Channing Frye   Jeff Green   Tristan Thompson
```

```
##      0.03030303      0.04761905      0.03571429
```

```
betweenness(net1)
```

```
##      Dwyane Wade      Isaiah Thomas      Jae Crowder      Jose Calderon
##      3.5000000      1.6666667      8.6666667      0.0000000
##      JR Smith      Kevin Love      Kyle Korver      LeBron James
##      6.0000000      0.6666667      11.5000000      24.0000000
##      Channing Frye      Jeff Green      Tristan Thompson
##      0.0000000      0.0000000      0.0000000
```

## Estimate density events

Density with respect to a concurrent variable

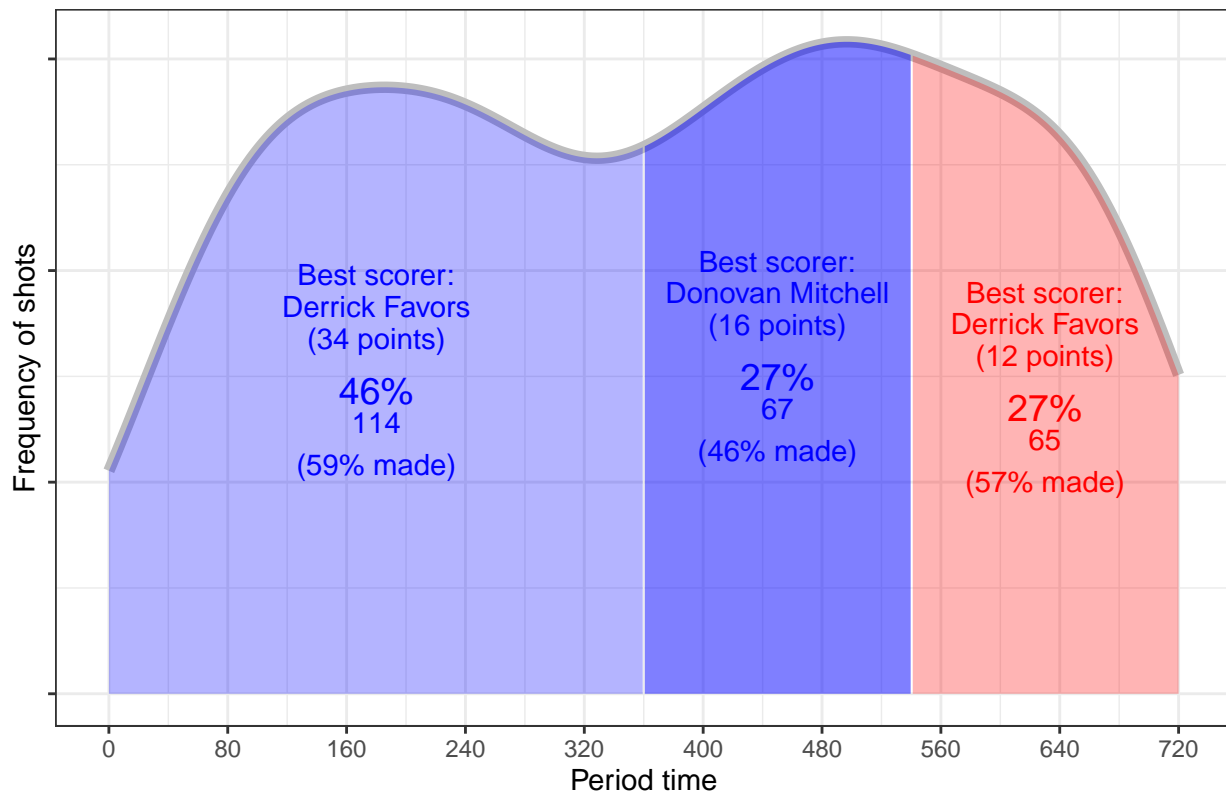
```
PbP <- PbPmanipulation(PbP.BDB)
unique(PbP$team)
```

```
## [1] HOU GSW NOP MEM DAL TOR WAS DET LAC SAS DEN MIA MIN PHI ORL BOS BKN OKC
## [20] CHI SAC LAL CHA POR CLE UTA MIL NYK PHX ATL IND
## 31 Levels: ATL BKN BOS CHA CHI CLE DAL DEN DET GSW HOU IND LAC LAL MEM ... WAS
```

```
data.team.UTA <- subset(PbP, team=="UTA" & result!="")
data.opp <- subset(PbP, team!="UTA" & result!="")
```

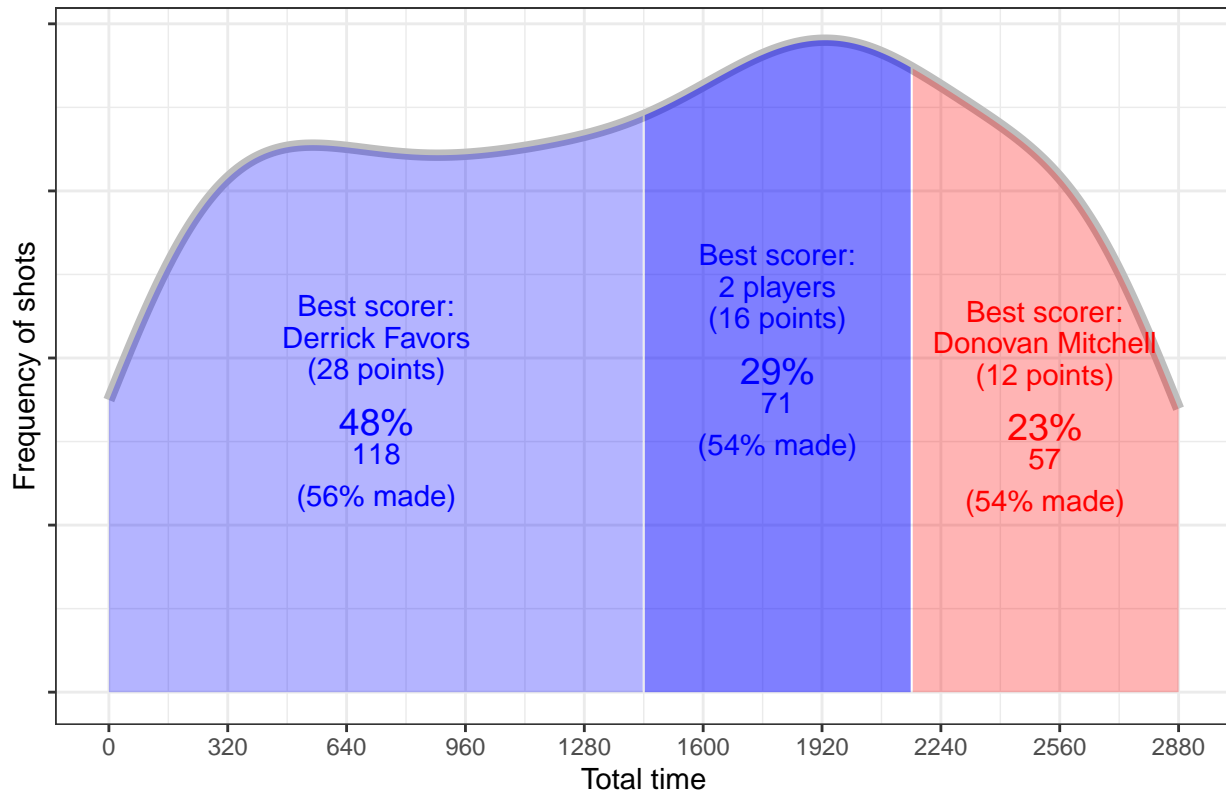
```
densityplot(data=data.team.UTA, shot.type="2P",
             var="periodTime", best.scorer=TRUE)
```

2P

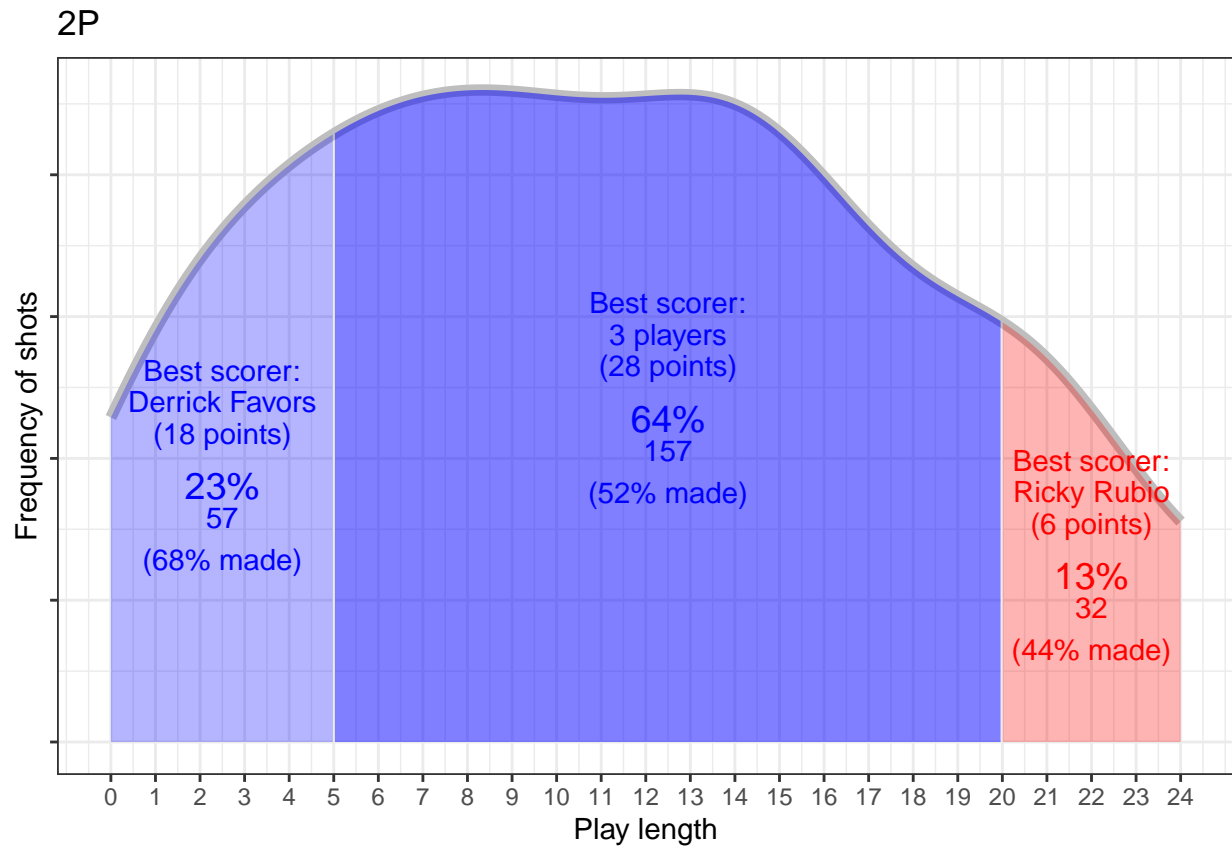


```
densityplot(data=data.team.UTA, shot.type="2P",
            var="totalTime", best.scorer=TRUE)
```

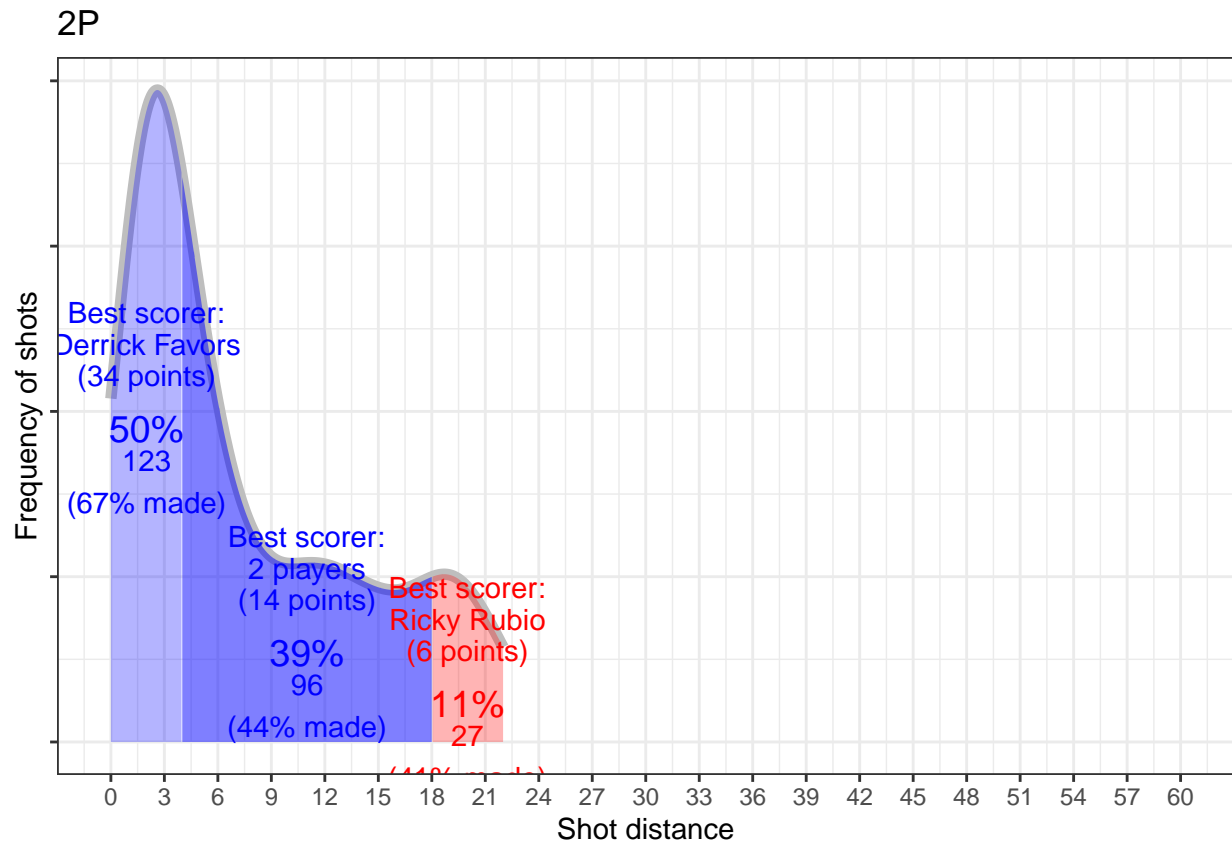
2P



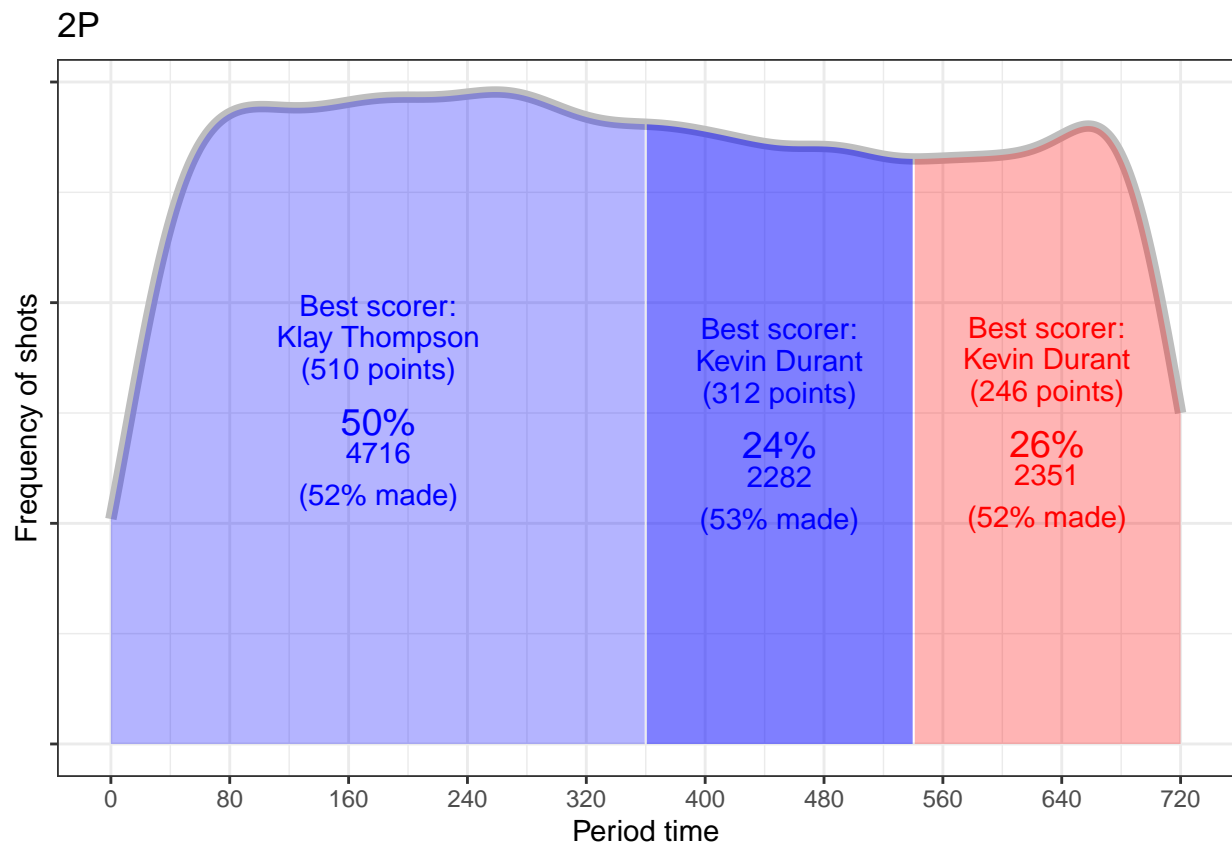
```
densityplot(data=data.team.UTA, shot.type="2P",
            var="playlength", best.scorer=TRUE)
```



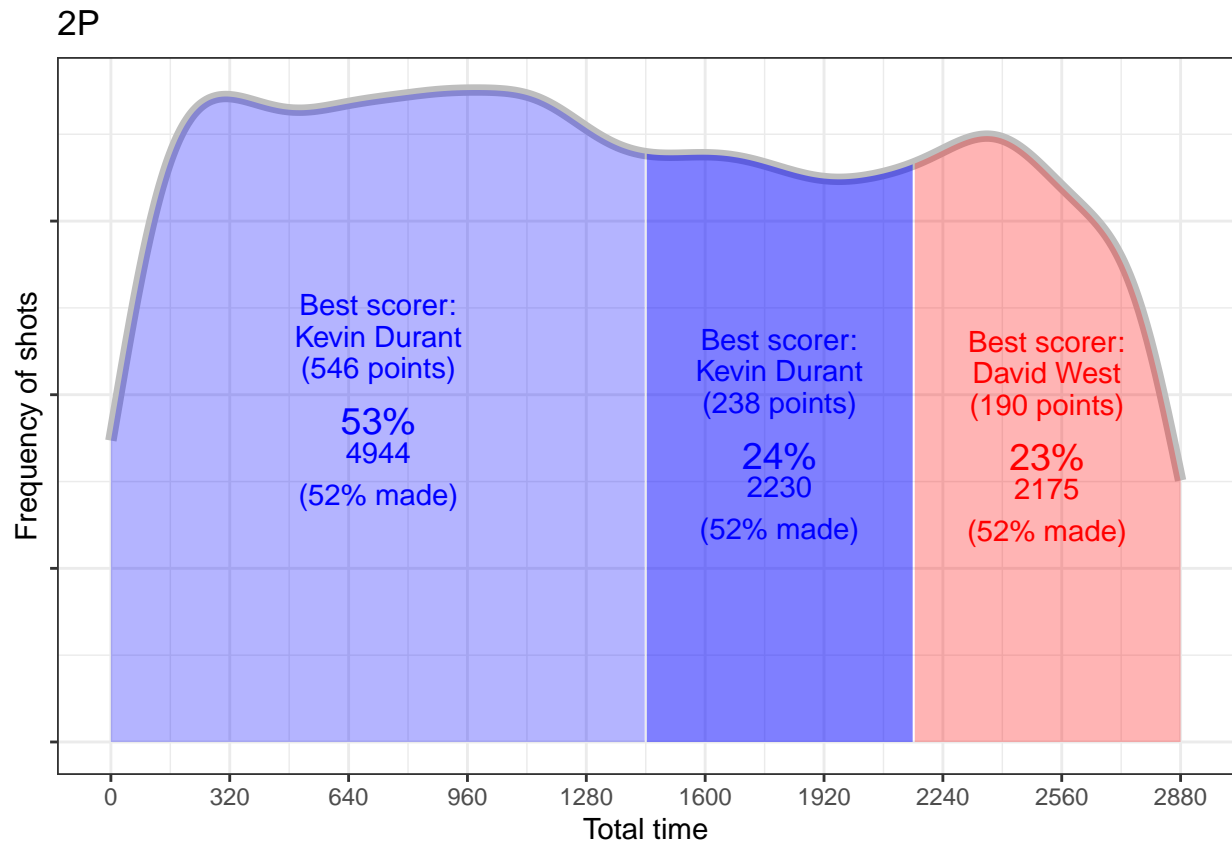
```
densityplot(data=data.team.UTA, shot.type="2P",  
            var="shot_distance", best.scorer=TRUE)
```



```
densityplot(data=data.opp, shot.type="2P",
            var="periodTime", best.scorer=TRUE)
```



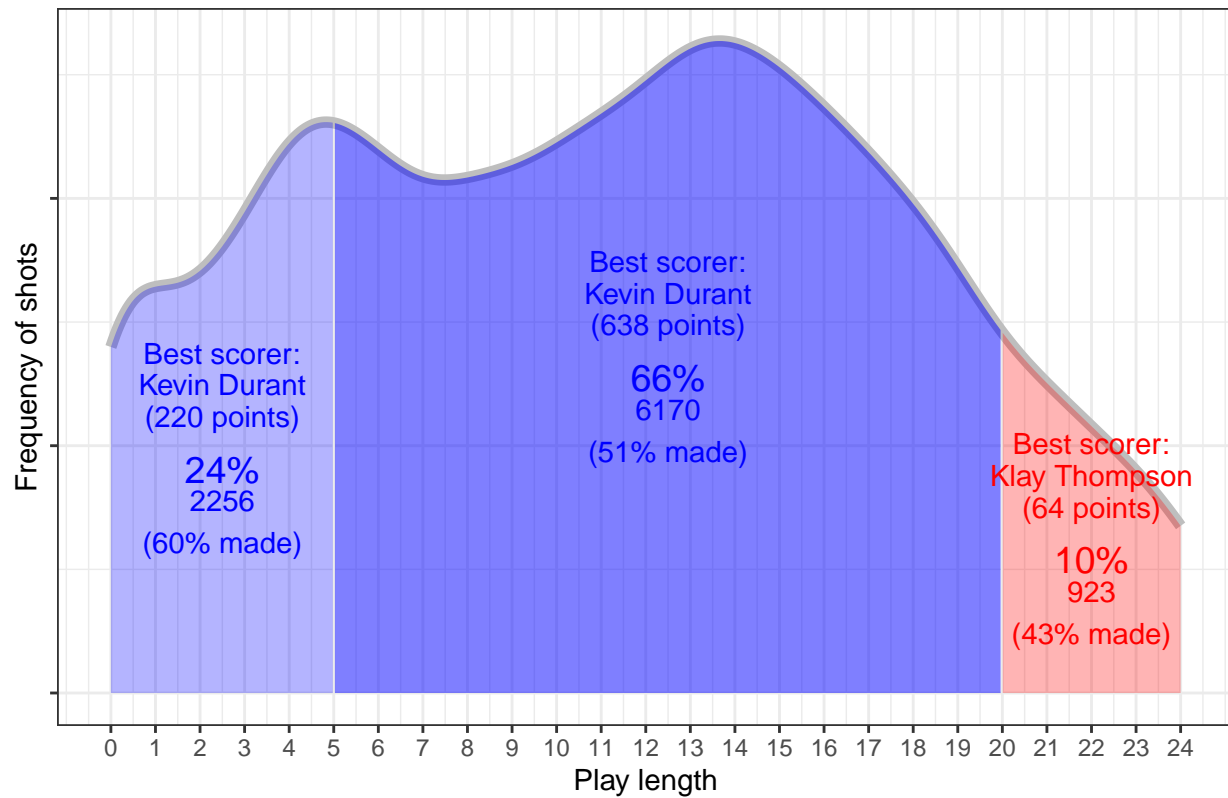
```
densityplot(data=data.opp, shot.type="2P",  
            var="totalTime", best.scorer=TRUE)
```



```
densityplot(data=data.opp, shot.type="2P",  
            var="playlength", best.scorer=TRUE)
```

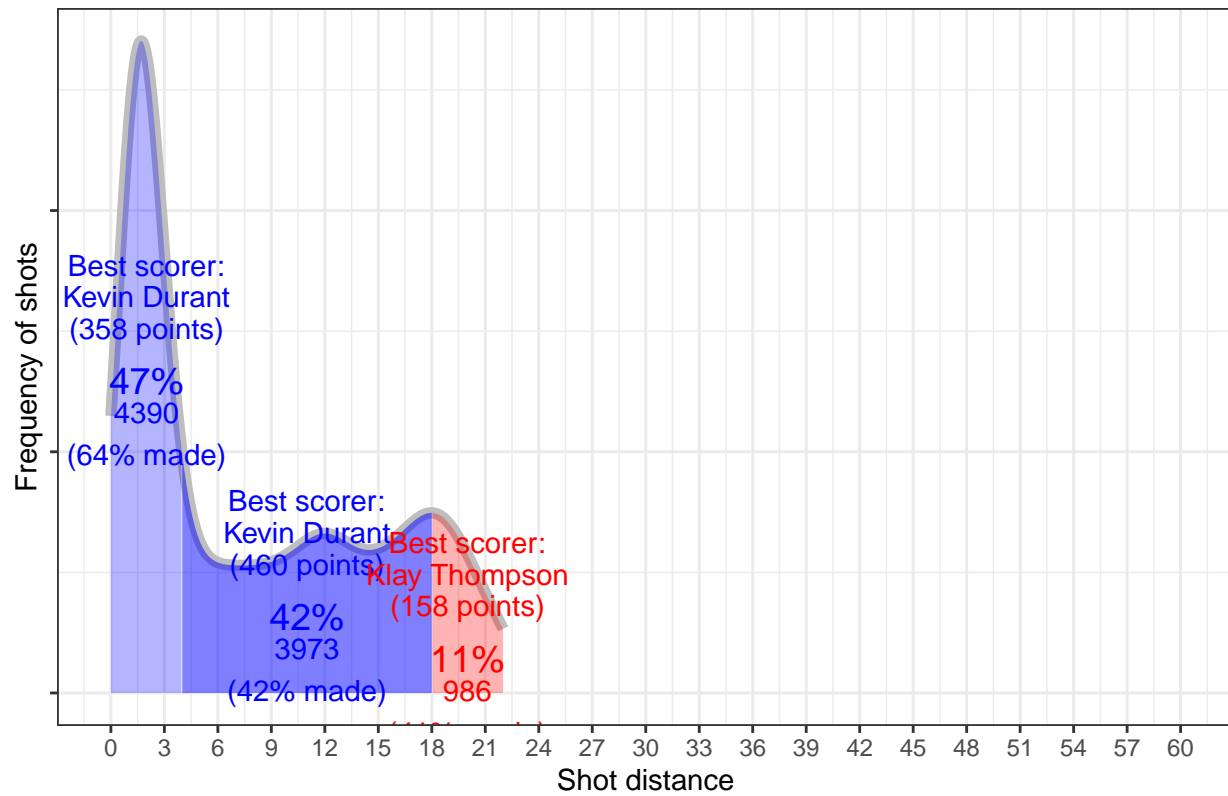


2P



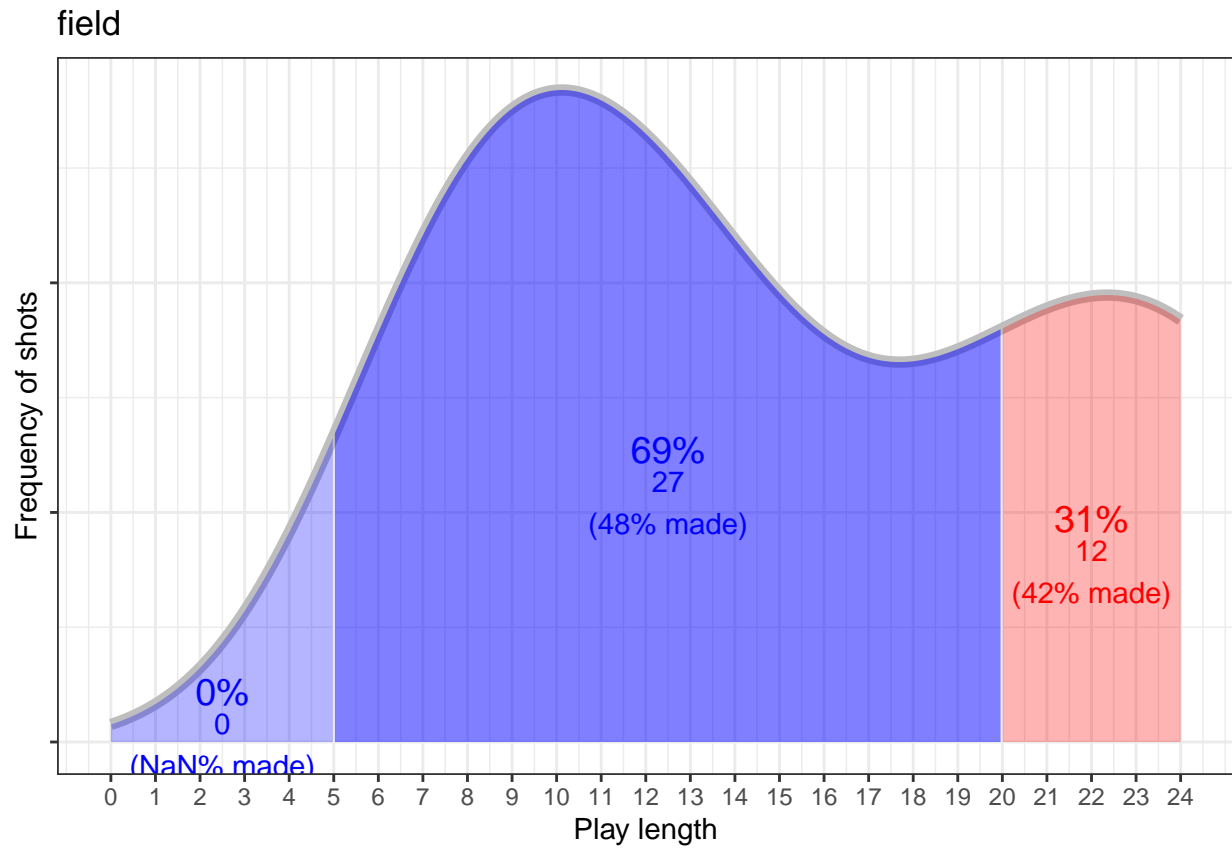
```
densityplot(data=data.opp, shot.type="2P",  
            var="shot_distance", best.scorer=TRUE)
```

2P

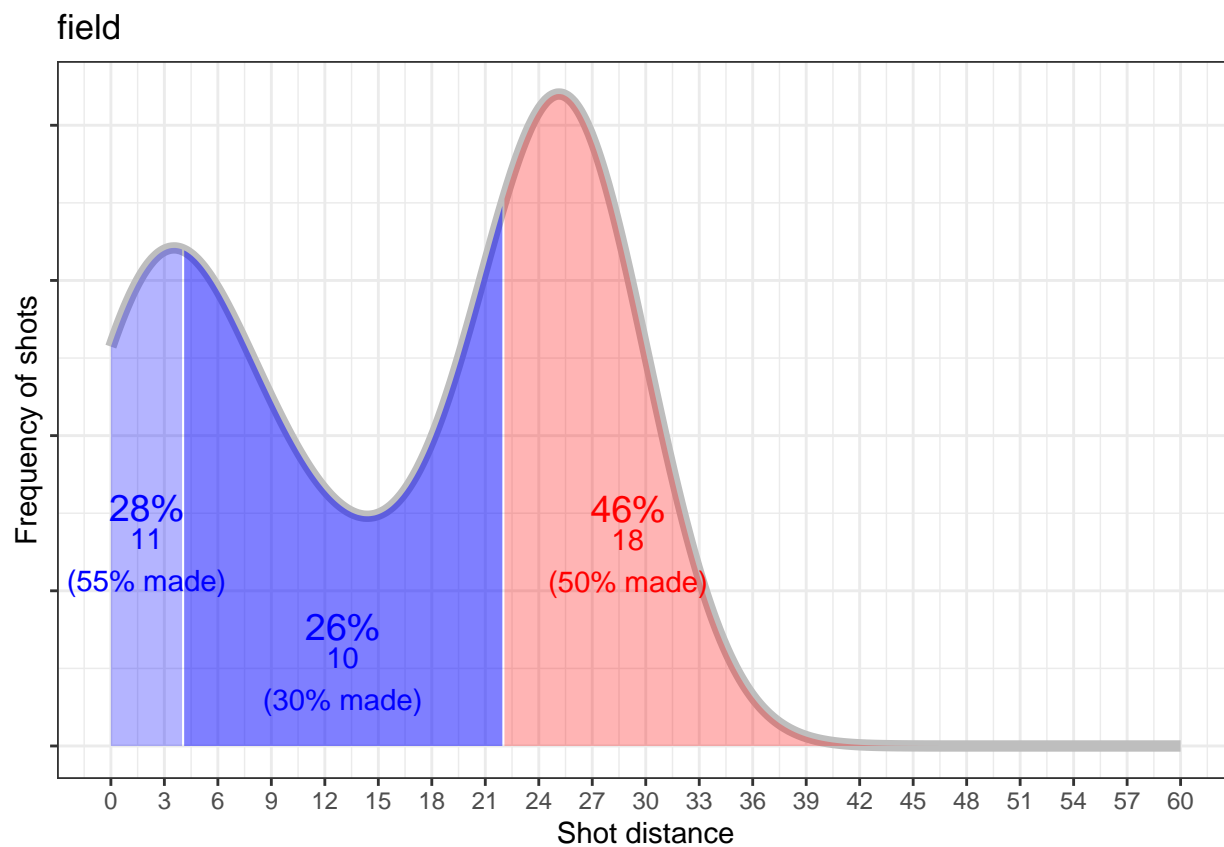


```
LJ <- subset(PbP, player=="LeBron James" & result!="")
AD <- subset(PbP, player=="Anthony Davis" & result!="")
JH <- subset(PbP, player=="James Harden" & result!="")
```

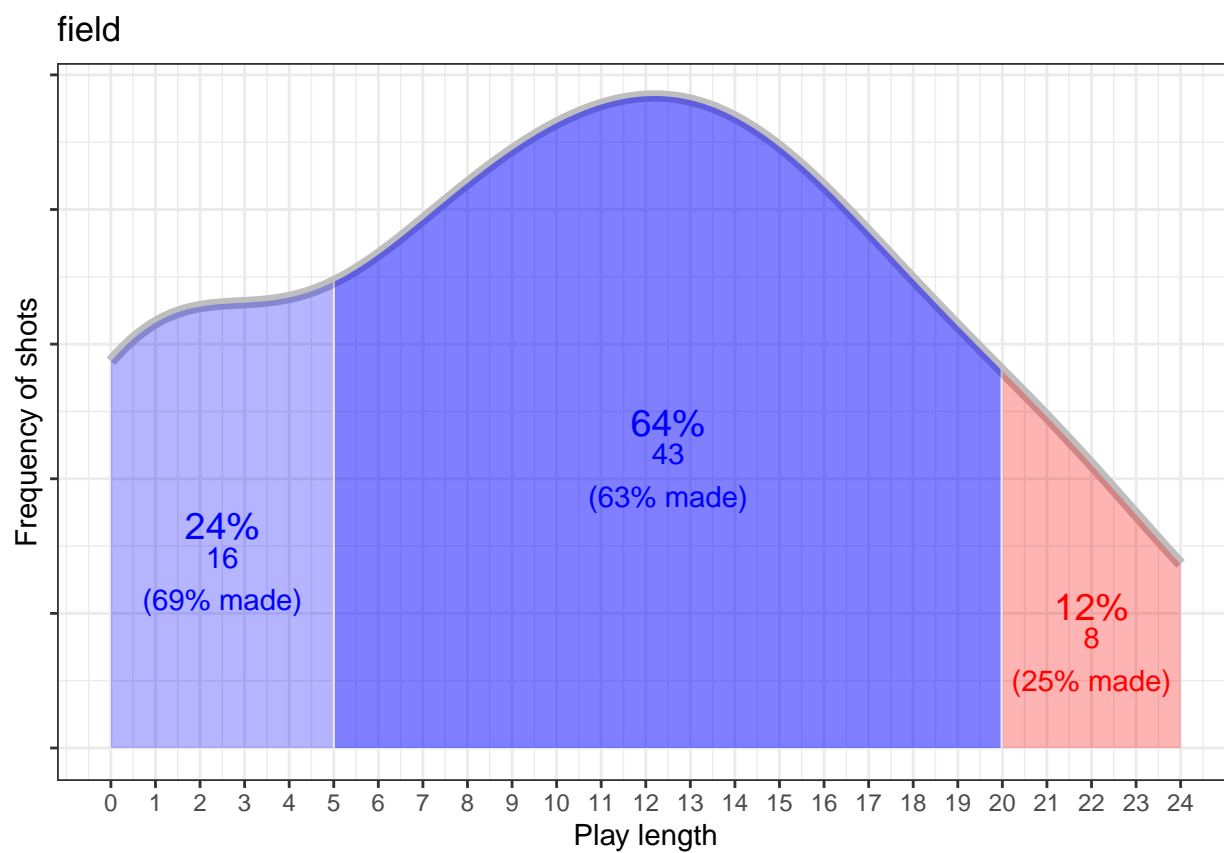
```
# James Harden
densityplot(data=JH, shot.type="field",
             var="playlength")
```



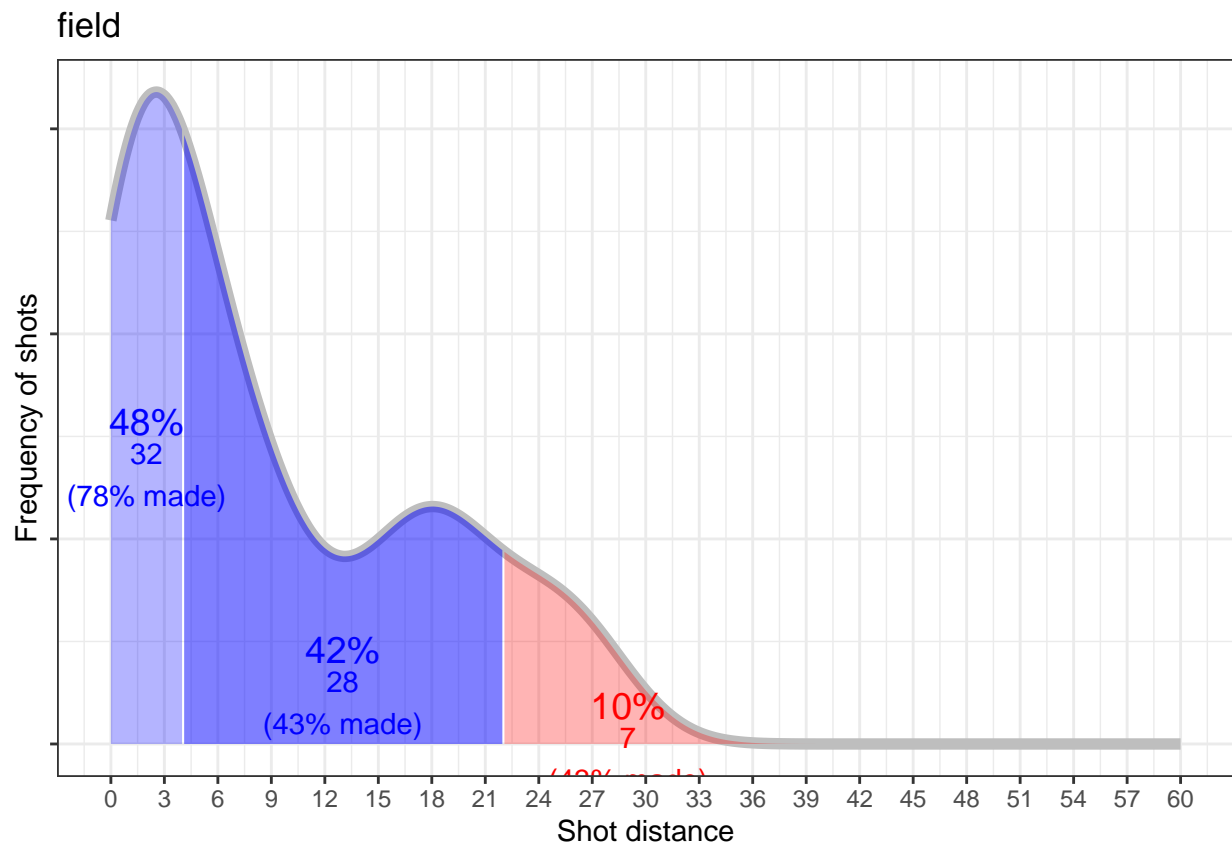
```
densityplot(data=JH, shot.type="field",  
            var="shot_distance")
```



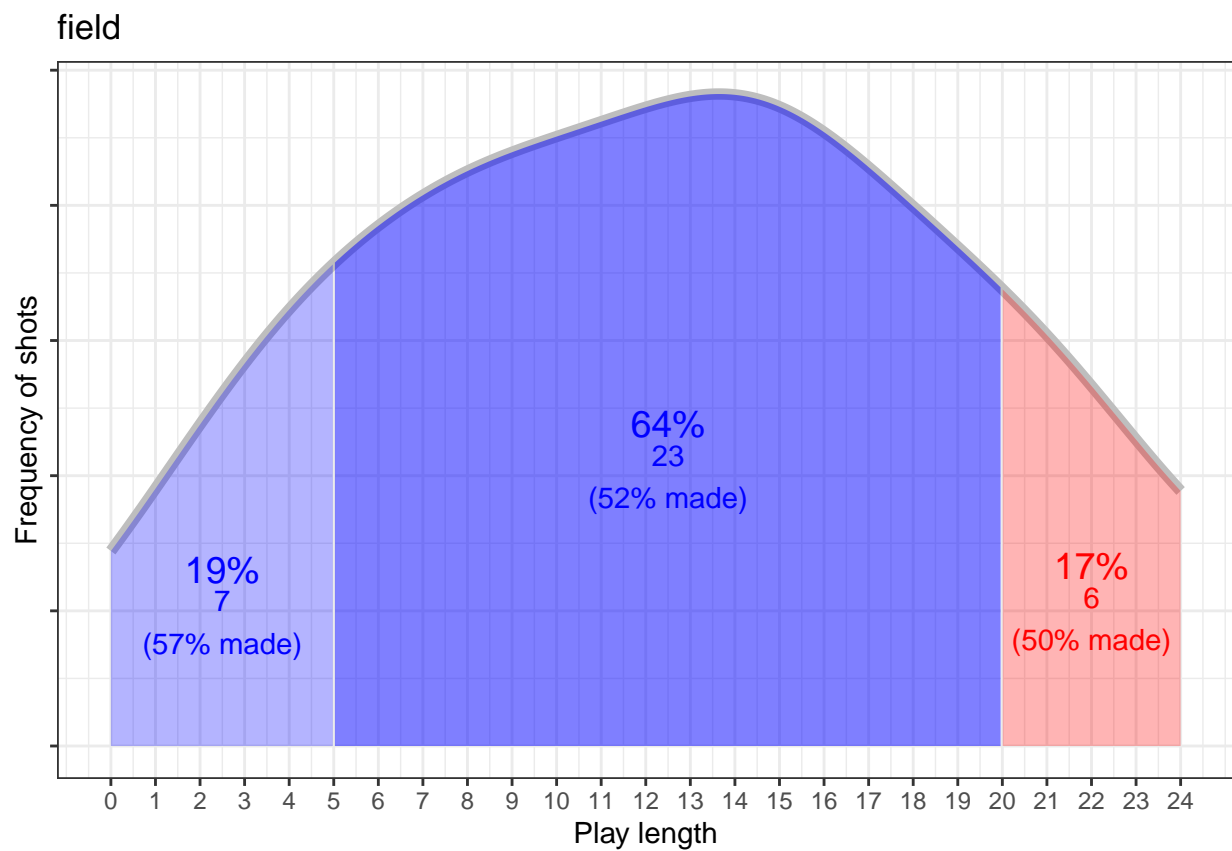
```
# Anthony Davis  
densityplot(data=AD, shot.type="field",  
            var="playlength")
```



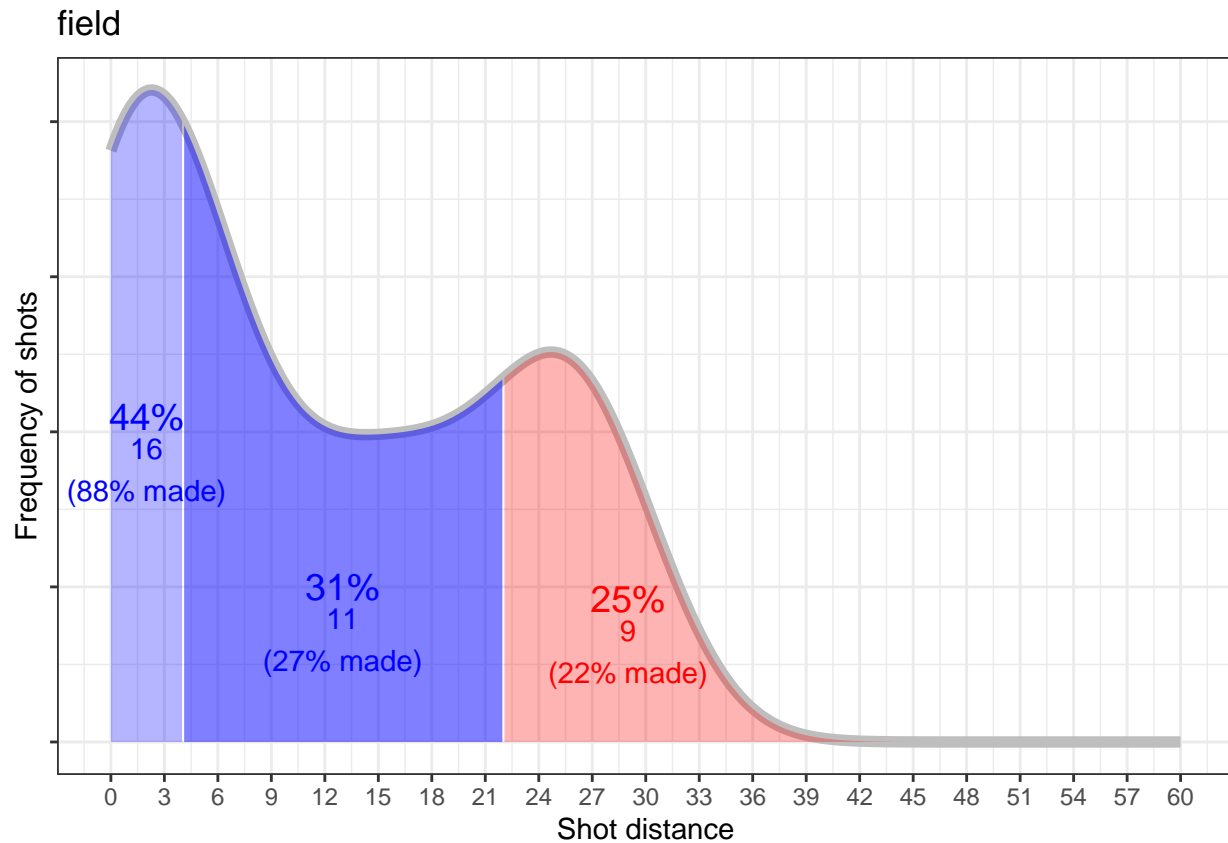
```
densityplot(data=AD, shot.type="field",  
            var="shot_distance")
```



```
# LeBron James
densityplot(data=LJ, shot.type="field",
            var="playlength")
```



```
densityplot(data=LJ, shot.type="field",  
            var="shot_distance")
```



## Density in Space

```
PbP <- PbPmanipulation(PbP.BDB)
```

```
PbP$xx <- PbP$original_x/10
```

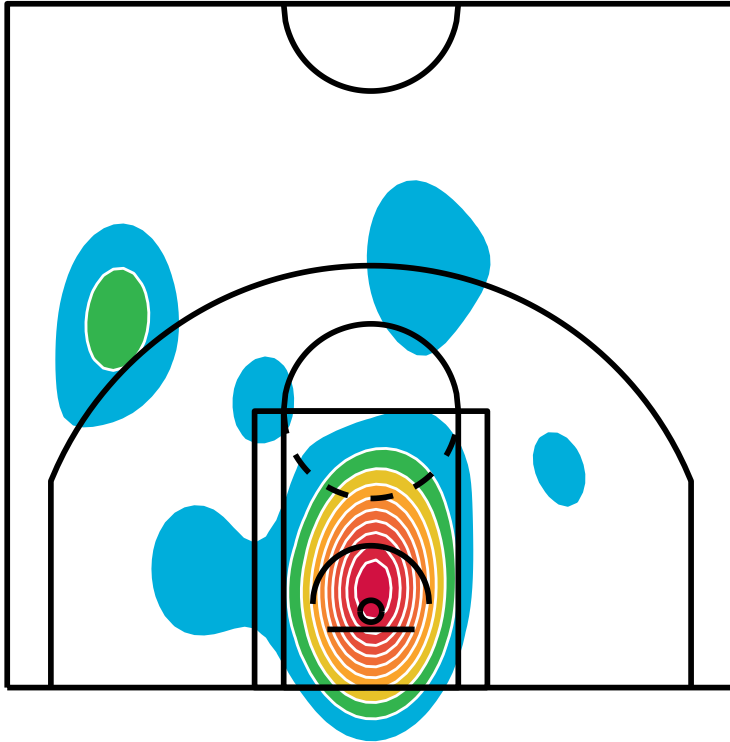
```
PbP$yy <- PbP$original_y/10 - 41.75
```

```
LJ <- subset(PbP, player=="LeBron James")
```

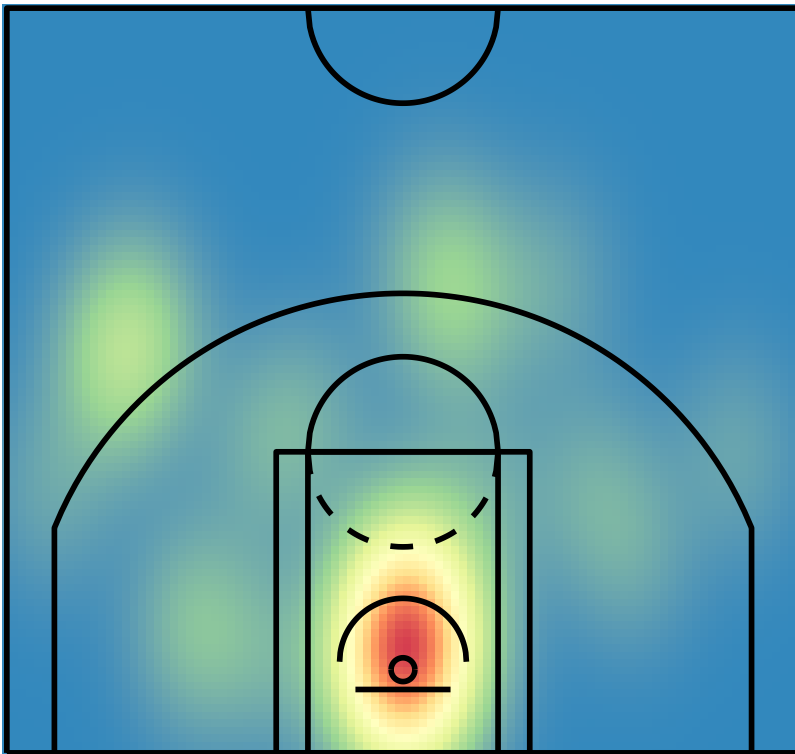
```
shotchart(data=LJ, x="xx", y="yy",
           type="density-polygons")
```

```
## Warning: The dot-dot notation (`..level..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(level)` instead.
## i The deprecated feature was likely used in the BasketballAnalyzeR package.
## Please report the issue to the authors.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

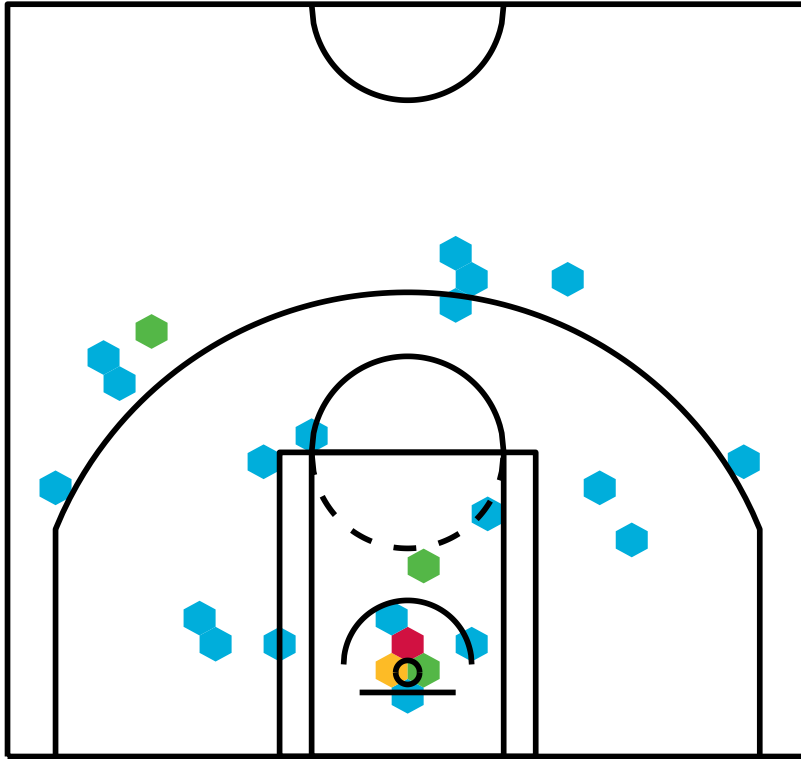




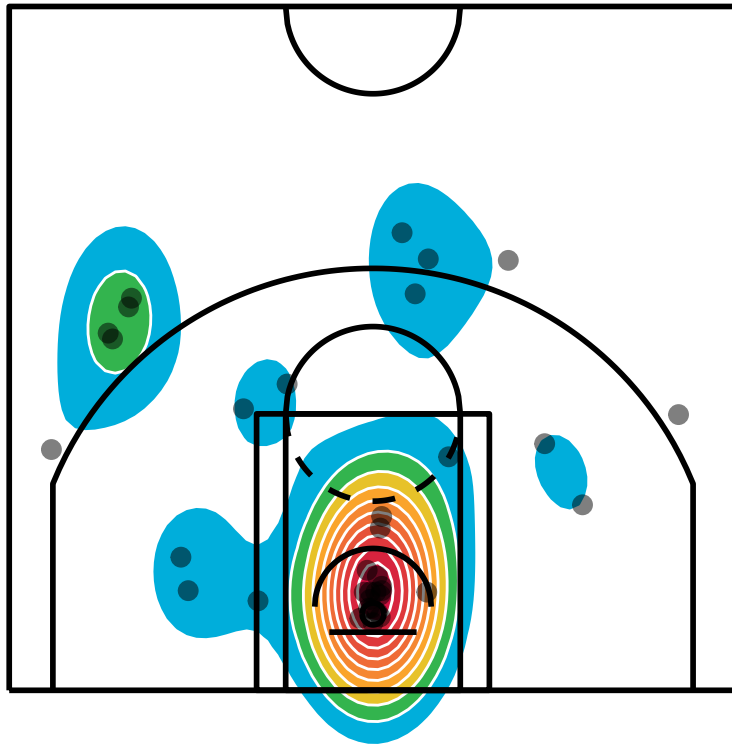
```
shotchart(data=LJ, x="xx", y="yy", type="density-raster")
```



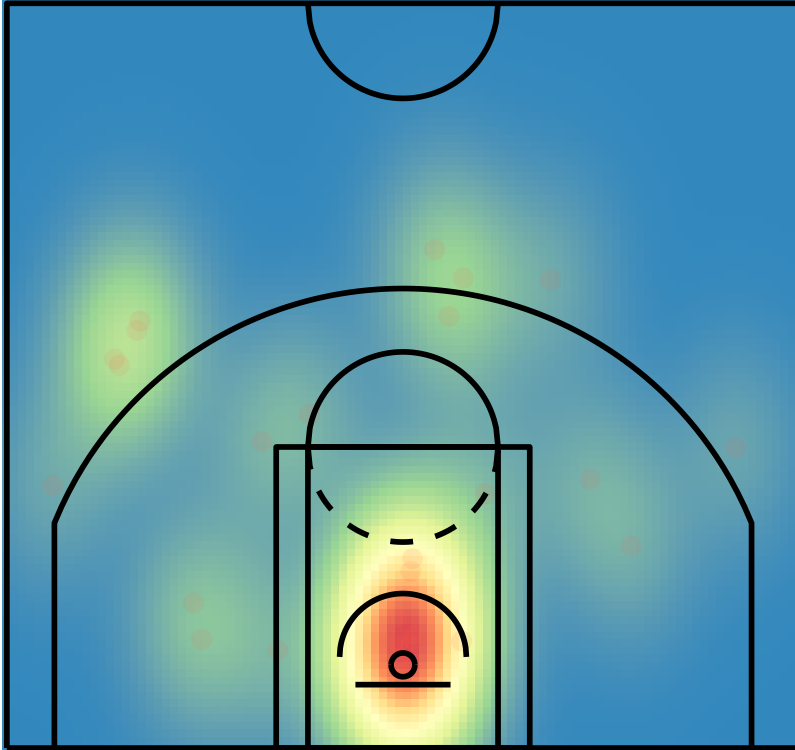
```
shotchart(data=LJ, x="xx", y="yy", type="density-hexbin")
```



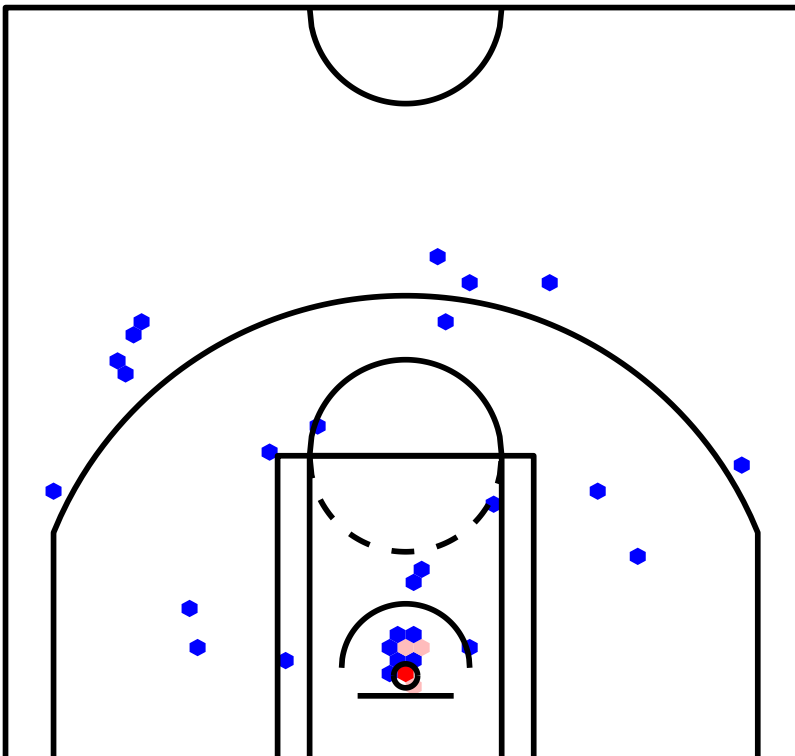
```
shotchart(data=LJ, x="xx", y="yy",
          type="density-polygons", scatter=TRUE)
```



```
shotchart(data=LJ, x="xx", y="yy", type="density-raster",
          scatter=TRUE, pt.col="tomato", pt.alpha=0.1)
```



```
shotchart(data=LJ, x="xx", y="yy", type="density-hexbin",
          nbins=50, palette="bwr")
```

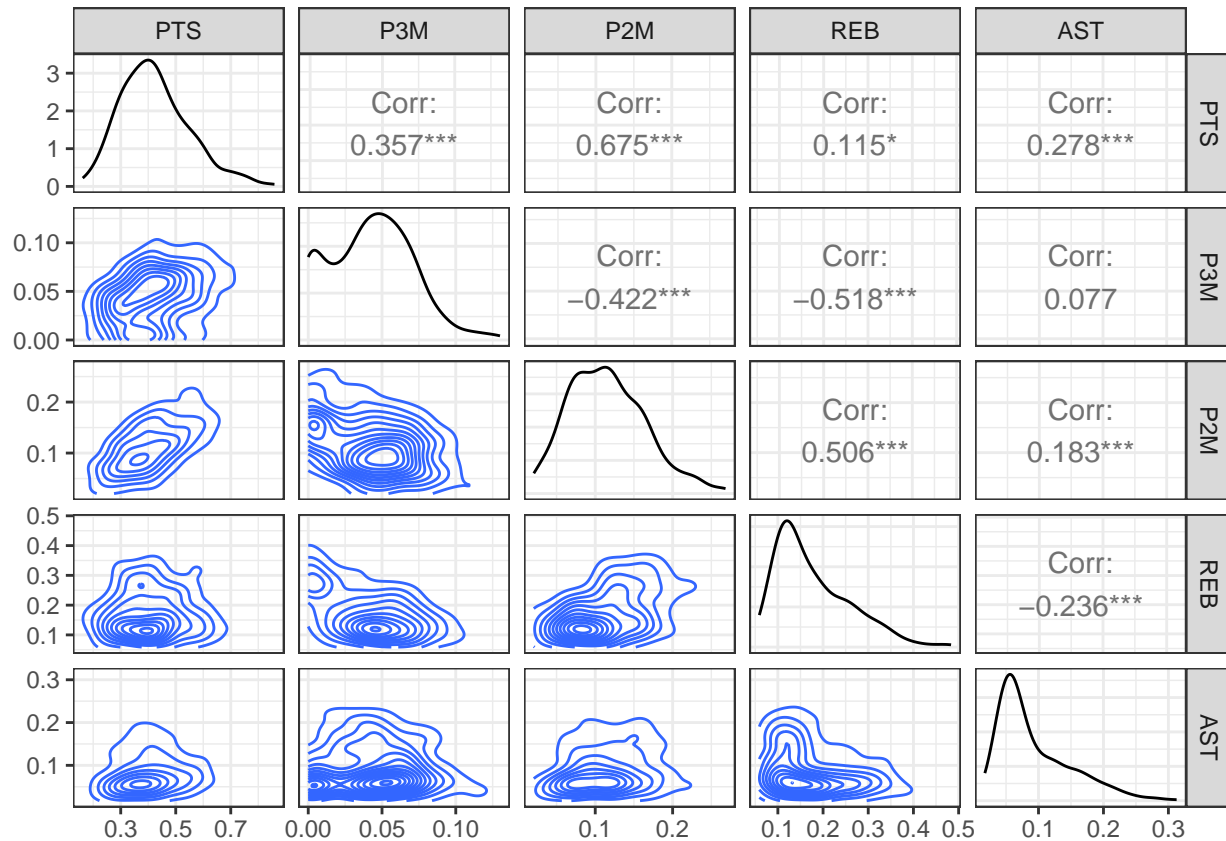


## Joint density of two variables

```
data.density <- subset(Pbox, MIN>=500)

attach(data.density)
X.density <- data.frame(PTS, P3M, P2M, REB=OREB+DREB, AST)/MIN
detach(data.density)

scatterplot(X.density, data.var=1:5,
            lower=list(continuous="density"),
            diag=list(continuous="densityDiag"))
```



## FINDING GROUP IN DATA

### Cluster Analysis

```
FF <- fourfactors(Tbox, Obox)

OD.Rtg <- FF$ORtg/FF$DRtg

F1.r <- FF$F1.Off/FF$F1.Def

F2.r <- FF$F2.Def/FF$F2.Off

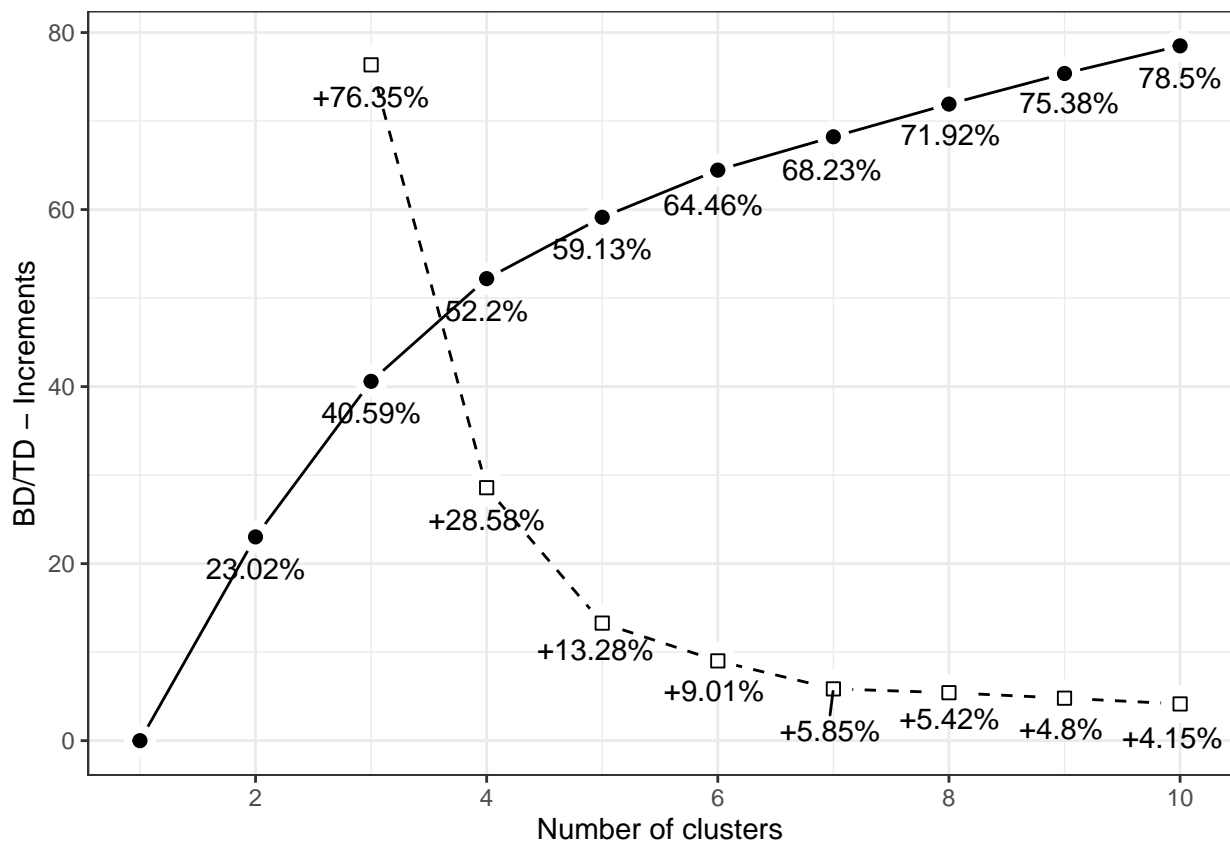
F3.Off <- FF$F3.Off
```

```
F3.Def <- FF$F3.Def
P3M <- Tbox$P3M
STL.r <- Tbox$STL/Obox$STL
data <- data.frame(OD.Rtg, F1.r, F2.r, F3.Off, F3.Def,
  P3M, STL.r)
```

```
set.seed(123)
```

```
kclu1 <- kclustering(data)
```

```
plot(kclu1)
```

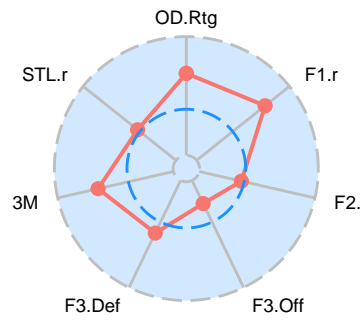


```
set.seed(123)
```

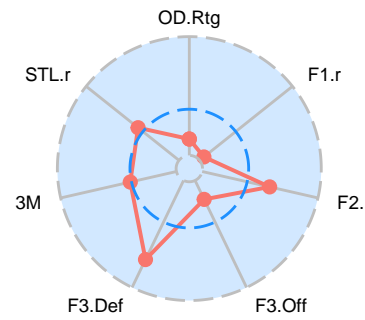
```
kclu2 <- kclustering(data, labels=Tbox$Team, k=5)
```

```
plot(kclu2)
```

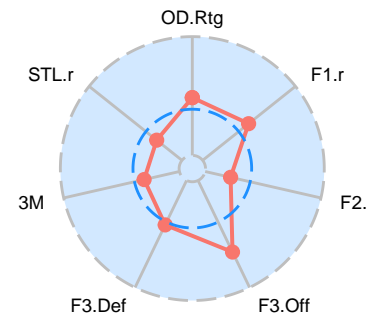
Cluster 1 – CHI = 0.48



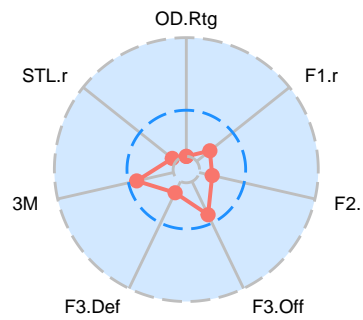
Cluster 2 – CHI = 0.41



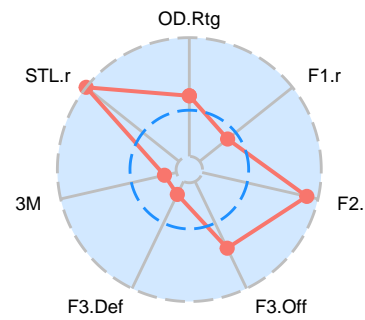
Cluster 3 – CHI = 0.31



Cluster 4 – CHI = 0.37



Cluster 5 – CHI = 0.39

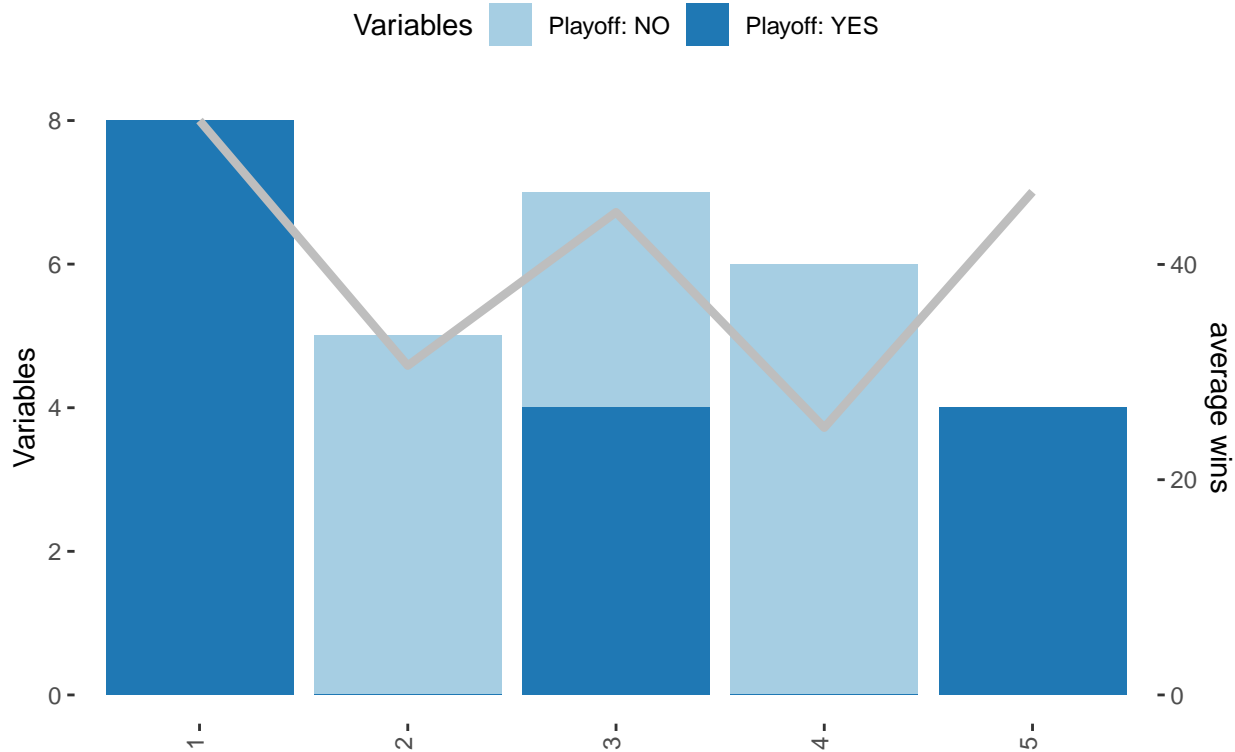


```
kclu2.PO <- table(kclu2$Subjects$Cluster, Tadd$Playoff)

kclu2.W <- tapply(Tbox$W, kclu2$Subjects$Cluster, mean)

Xbar <- data.frame(cluster=c(1:5), N=kclu2.PO[,1],
                   Y=kclu2.PO[,2], W=kclu2.W)

barline(data=Xbar, id="cluster", bars=c("N","Y"),
        labels.bars=c("Playoff: NO","Playoff: YES"),
        line="W", label.line="average wins",
        decreasing=FALSE)
```



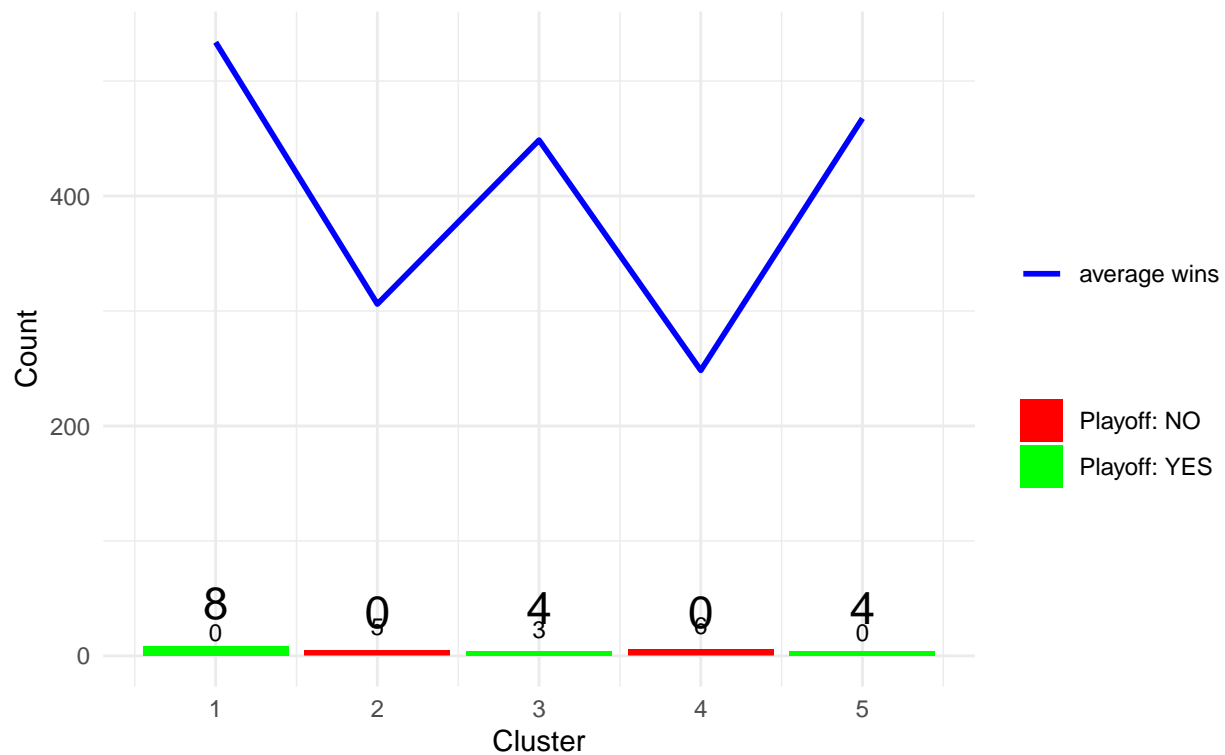
Bars ordered by cluster

```
ggplot(data = Xbar, aes(x = cluster)) +
  geom_bar(aes(y = N, fill = "Playoff: NO"), stat = "identity", position = "dodge") +
  geom_bar(aes(y = Y, fill = "Playoff: YES"), stat = "identity", position = "dodge") +
  geom_line(aes(y = W * 10, group = 1, color = "average wins"), size = 1) +
  geom_text(aes(y = N + 10, label = N), vjust = -0.2, size = 3, position = position_dodge(width = 0.9)) +
  geom_text(aes(y = Y + 10, label = Y), vjust = -0.5, size = 6, position = position_dodge(width = 0.9)) +
  scale_fill_manual(values = c("Playoff: NO" = "red", "Playoff: YES" = "green")) +
  scale_color_manual(values = c("average wins" = "blue")) +
  labs(
    x = "Cluster",
    y = "Count",
    fill = "",
    color = "",
    title = "Barline Chart with Labels",
    subtitle = "Comparison of No, Yes, and Average Wins by Cluster"
  ) +
  theme_minimal()
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

## Barline Chart with Labels

Comparison of No, Yes, and Average Wins by Cluster



*# The number with a bigger size are for "Yes" and smaller size are for "No"*

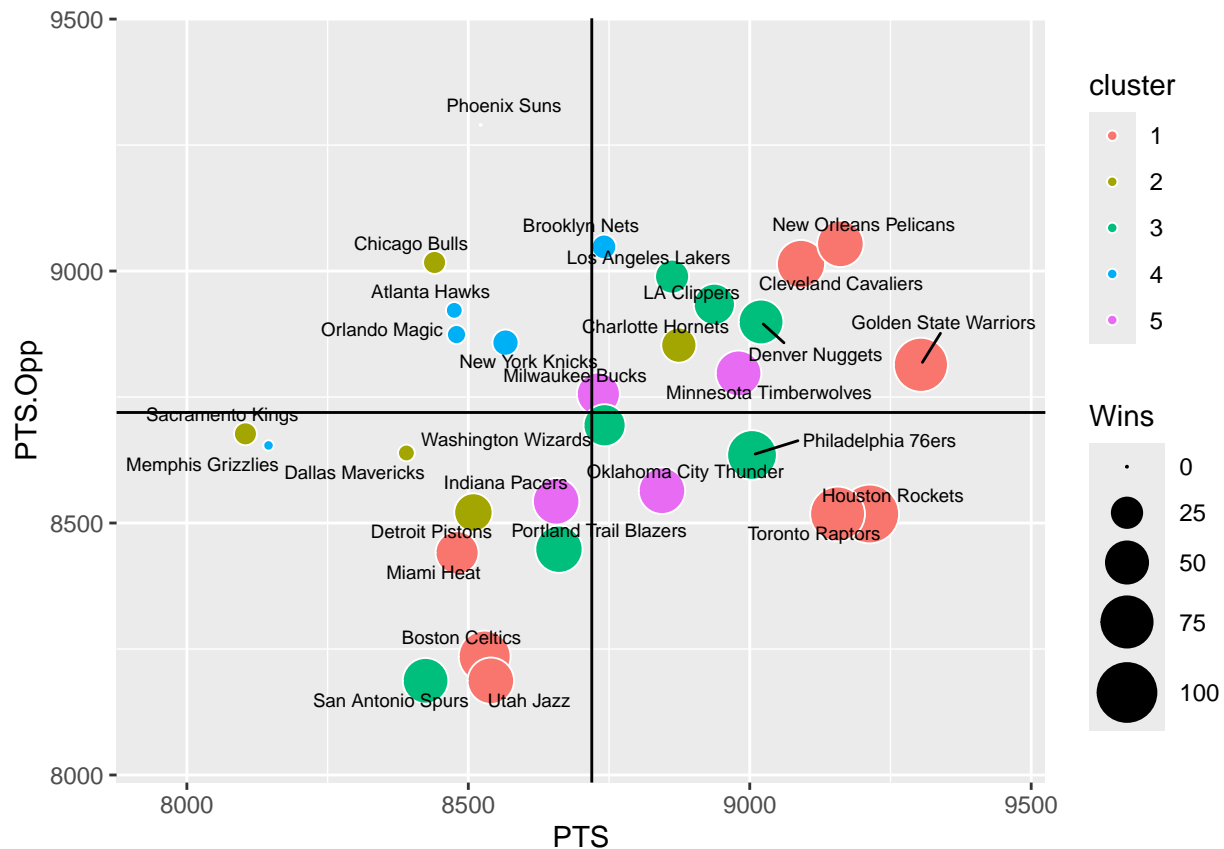
```
cluster <- as.factor(kclu2$Subjects$Cluster)

Xbubble <- data.frame(Team=Tbox$Team, PTS=Tbox$PTS,
                      PTS.Opp=Obox$PTS, cluster,
                      W=Tbox$W)

labs <- c("PTS", "PTS.Opp", "cluster", "Wins")

bubbleplot(Xbubble, id="Team", x="PTS", y="PTS.Opp",
            col="cluster", size="W", labels=labs)
```





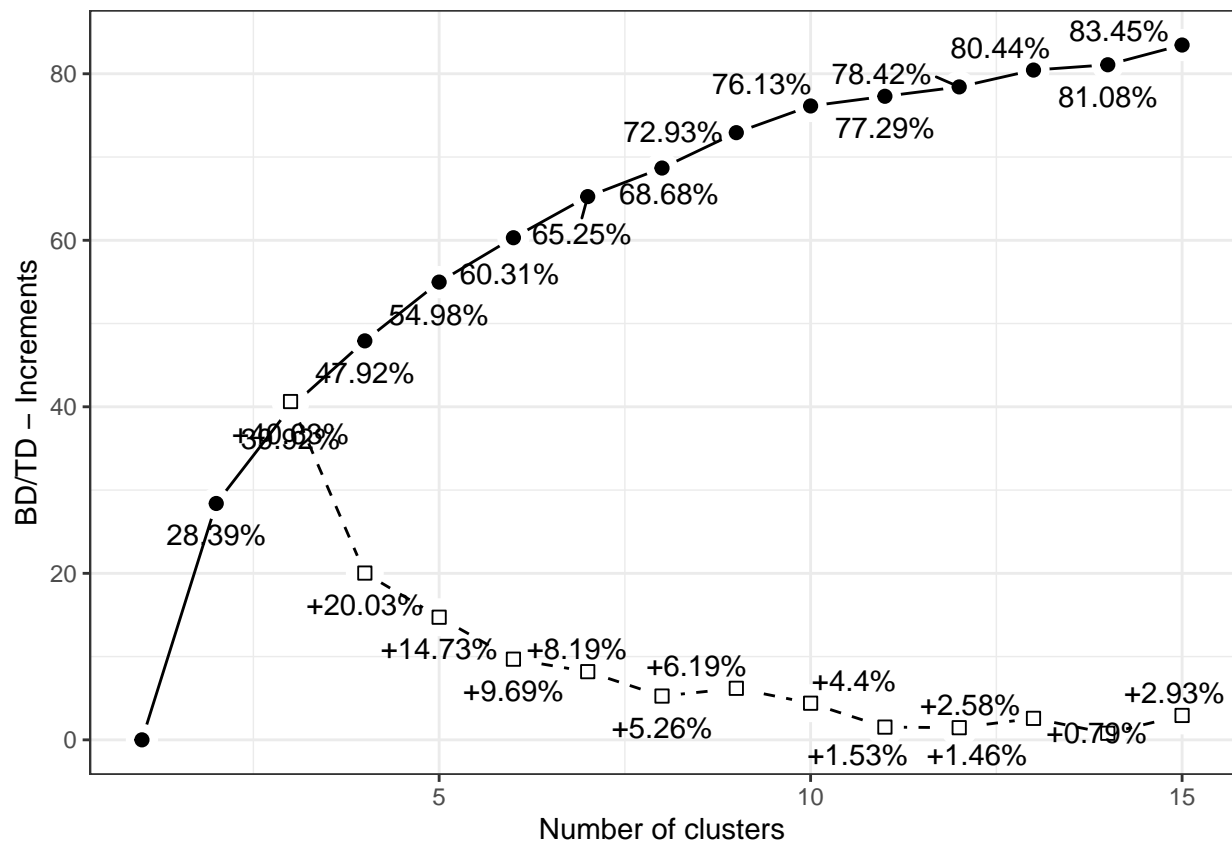
## K-means clustering of Toronto Raptors shots

```
PbP <- PbPmanipulation(PbP.BDB)

shots.TOR <- subset(PbP,
  !is.na(PbP$shot_distance) &
  PbP$team=="TOR")
shots.TOR <- mutate_if(shots.TOR, is.factor, droplevels)

attach(shots.TOR)
data.kclu <- data.frame(PTS=points, DIST=shot_distance,
  TIMEQ=periodTime, PL=playlength)
detach(shots.TOR)

set.seed(1)
kclu.TOR <- kclustering(data.kclu, algorithm="MacQueen",
  nclumax=15, iter.max=500)
plot(kclu.TOR)
```

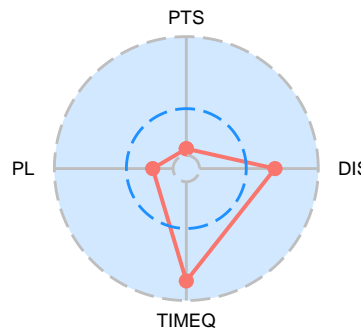


```
set.seed(246)

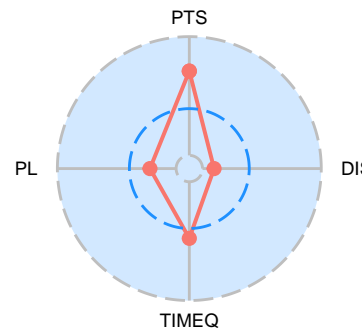
kclu.TOR2 <- kclustering(data.kclu, algorithm="MacQueen",
                        iter.max=500, k=6)

plot(kclu.TOR2)
```

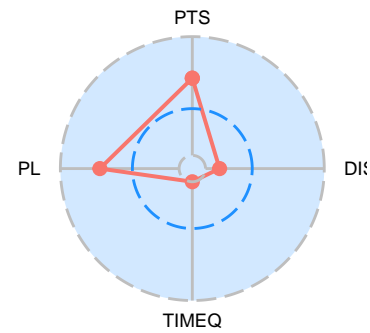
Cluster 1 – CHI = 0.56



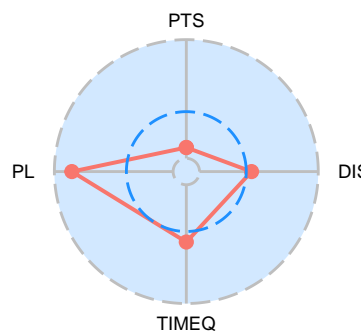
Cluster 2 – CHI = 0.3



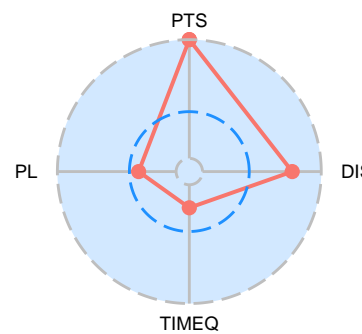
Cluster 3 – CHI = 0.3



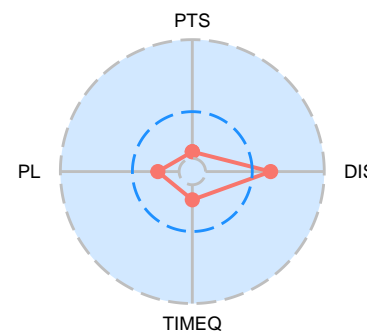
Cluster 4 – CHI = 0.45



Cluster 5 – CHI = 0.43



Cluster 6 – CHI = 0.34



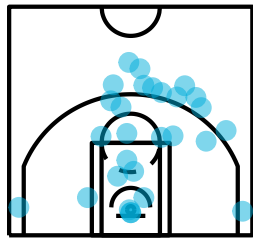
```
cluster <- as.factor(kclu.TOR2$Subjects$Cluster)
shots.TOR <- data.frame(shots.TOR, cluster)
shots.TOR$xx <- shots.TOR$original_x/10
shots.TOR$yy <- shots.TOR$original_y/10 - 41.75

no.clu.TOR <- 6

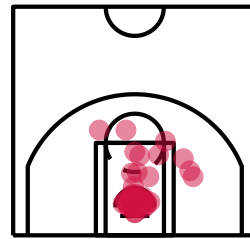
p1 <- p2 <- vector(no.clu.TOR, mode="list")

for (k in 1:no.clu.TOR) {
  shots.k <- subset(shots.TOR, cluster==k)
  p1[[k]] <- shotchart(data=shots.k, x="xx", y="yy",
    z="result", type=NULL,
    scatter = TRUE,
    drop.levels=FALSE)
  p2[[k]] <- shotchart(data=shots.k, x="xx", y="yy",
    z="periodTime",
    col.limits=c(0,720),
    result="result", num.sect=5,
    type="sectors", scatter=FALSE)
}
```

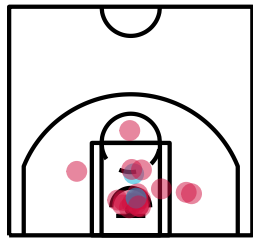
```
grid.arrange(grobs=p1, nrow=3)
```



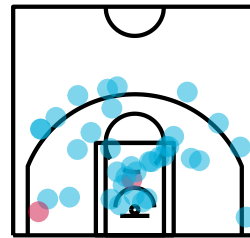
result  
made  
missed



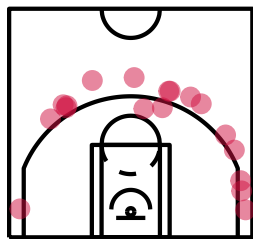
result  
made  
missed



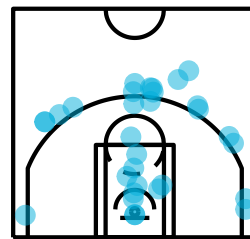
result  
made  
missed



result  
made  
missed



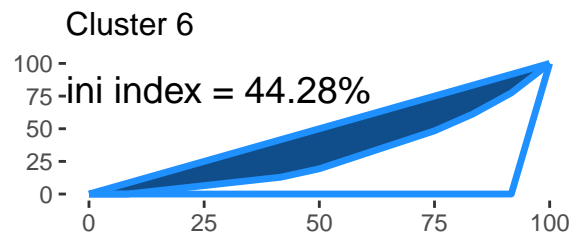
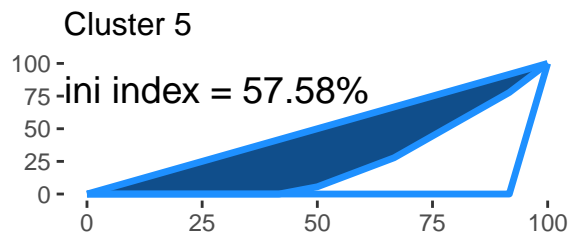
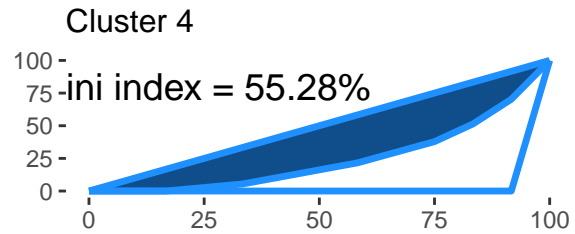
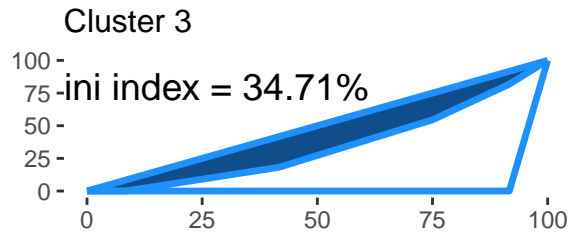
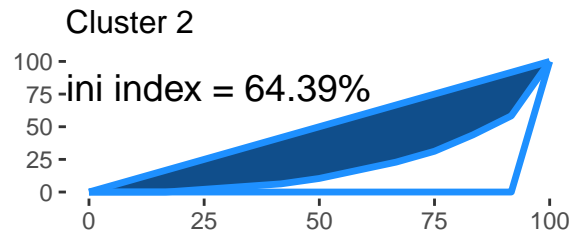
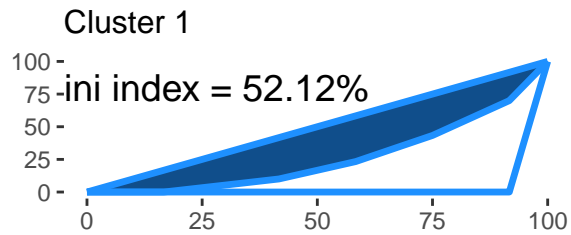
result  
made  
missed



result  
made  
missed

```
grid.arrange(grobs=p2, nrow=3)
```



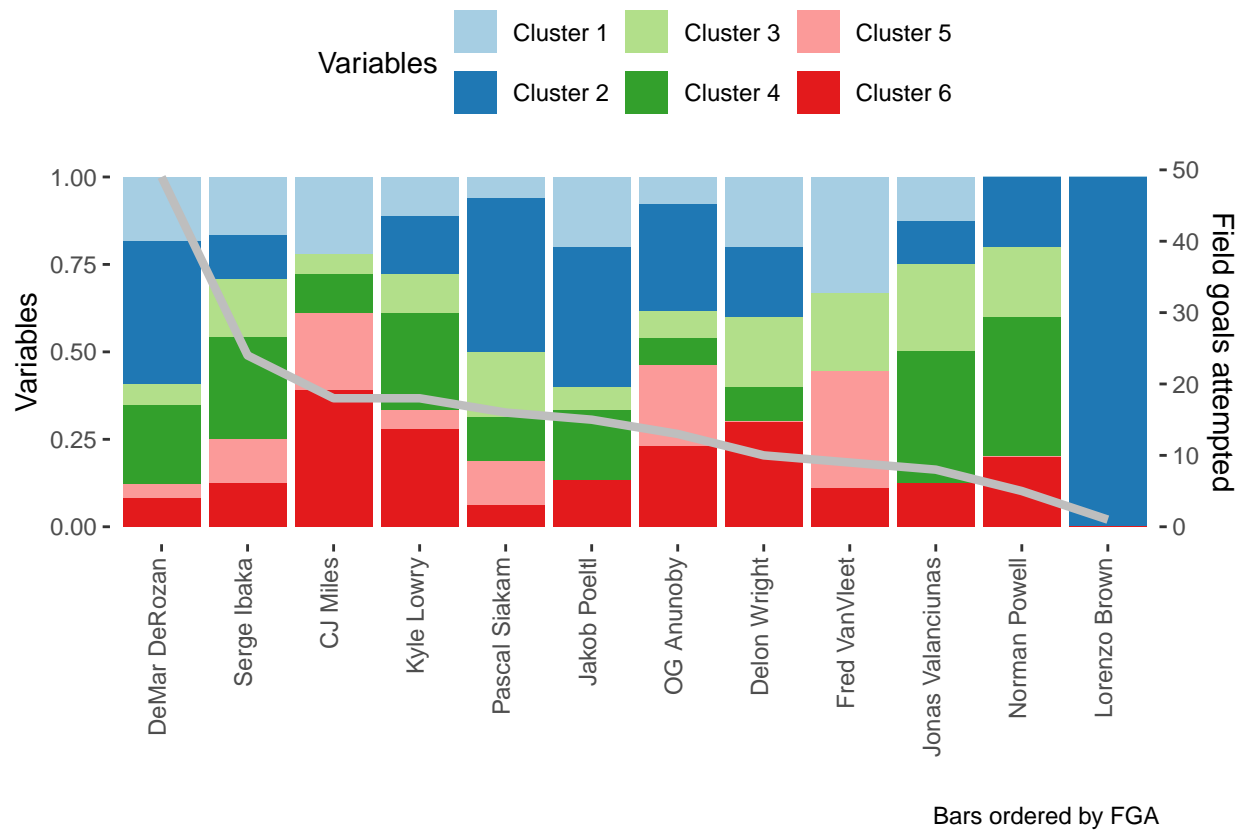


```
shots.perc <- shots.pl/rowSums(shots.pl)

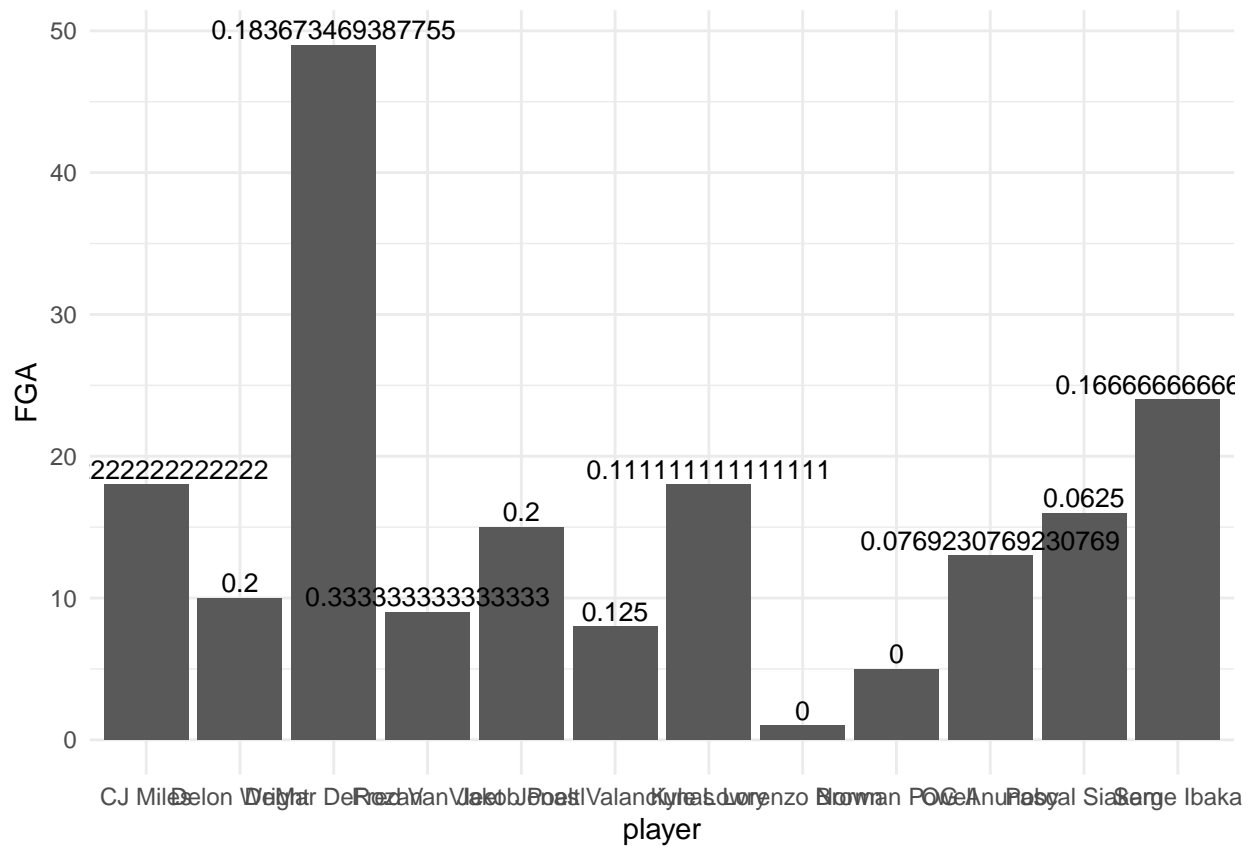
Xbar.TOR <- data.frame(player=rownames(shots.pl),
  rbind(shots.perc,
    FGA=rowSums(shots.pl))

labclusters <- c("Cluster 1","Cluster 2","Cluster 3",
  "Cluster 4","Cluster 5","Cluster 6")

barline(data=Xbar.TOR, id="player", line="FGA",
  bars=c("X1","X2","X3","X4","X5","X6"),
  order.by="FGA", label.line="Field goals attempted",
  labels.bars=labclusters)
```

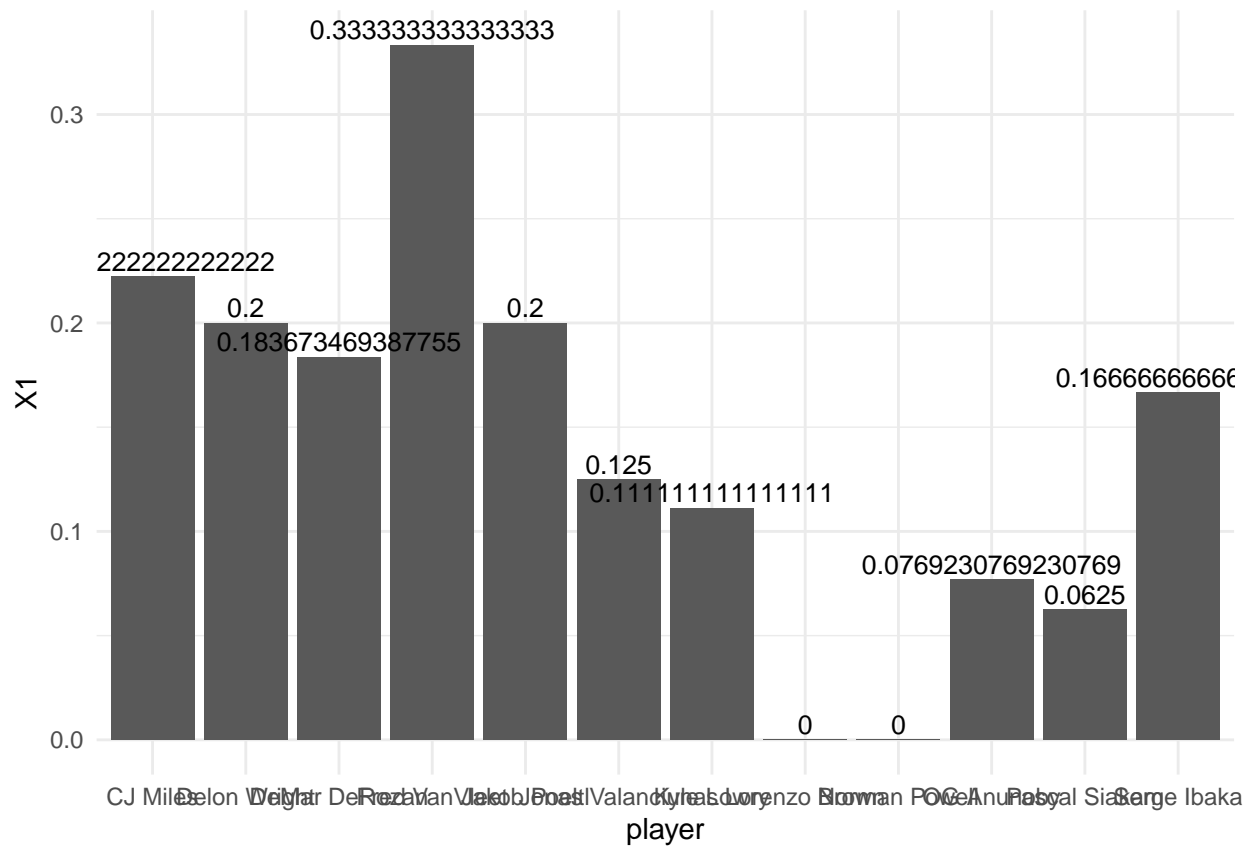


```
ggplot(Xbar.TOR, aes(x=player, y=FGA)) +
  geom_bar(stat="identity") +
  geom_text(aes(label=X1), vjust=-0.3, size=3.5) +
  theme_minimal()
```

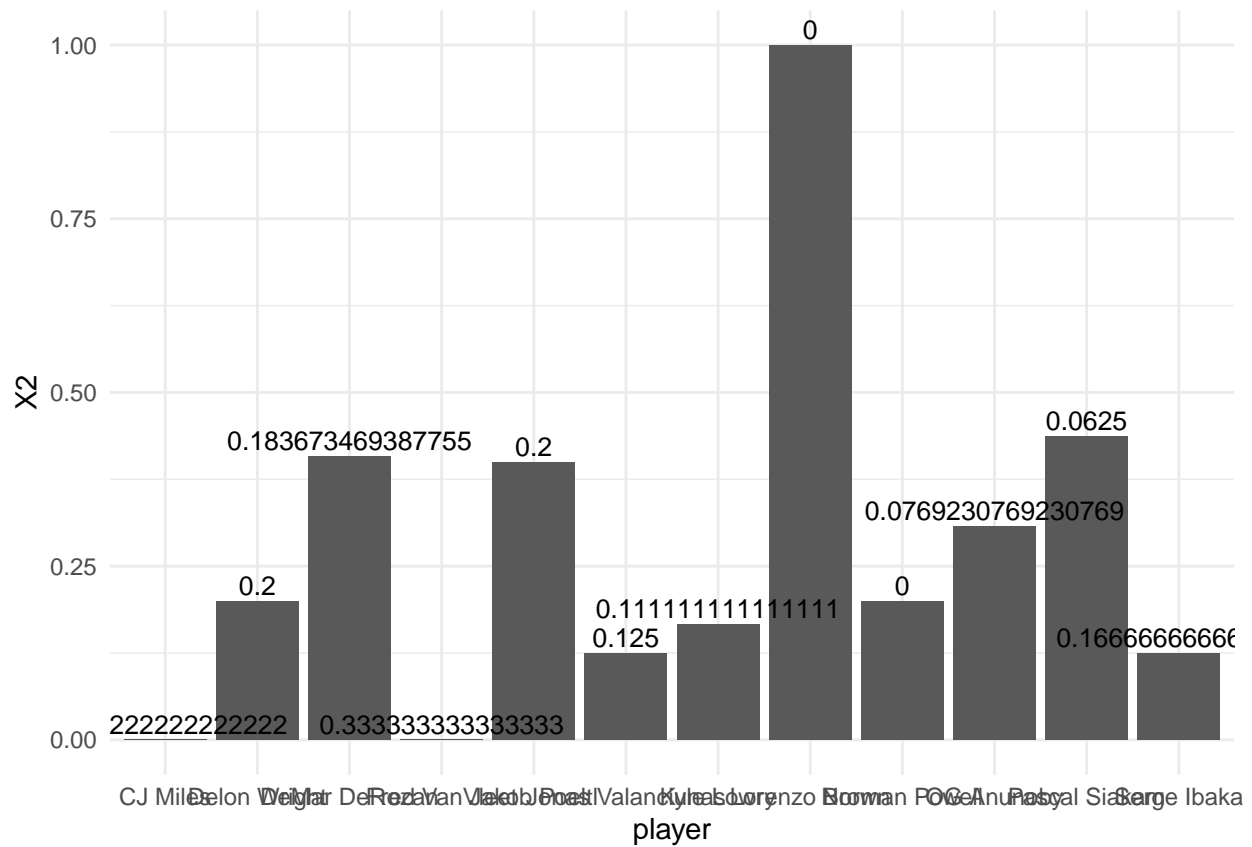


```
ggplot(Xbar.TOR, aes(x=player, y=X1)) +
  geom_bar(stat="identity") +
  geom_text(aes(label=X1), vjust=-0.3, size=3.5) +
  theme_minimal()
```

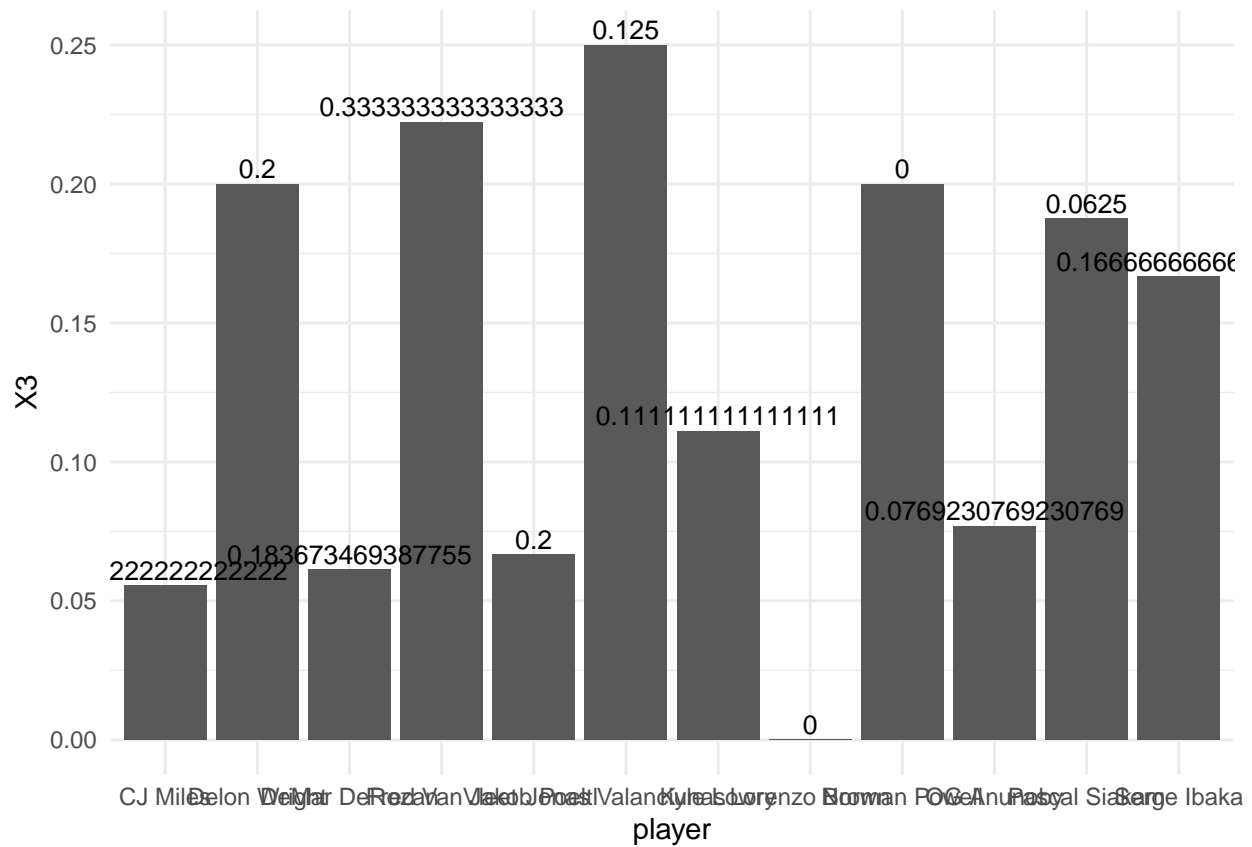




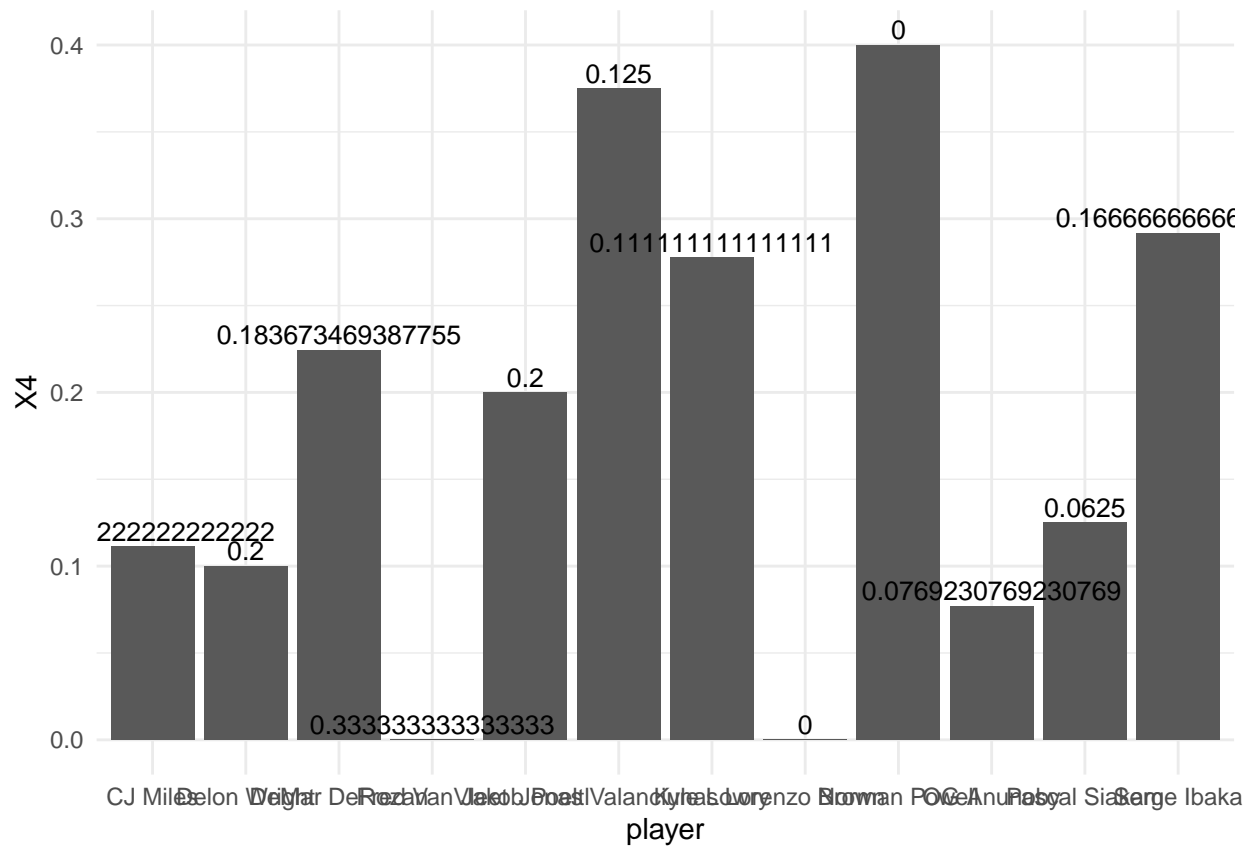
```
ggplot(Xbar.TOR, aes(x=player, y=X2)) +
  geom_bar(stat="identity") +
  geom_text(aes(label=X1), vjust=-0.3, size=3.5) +
  theme_minimal()
```



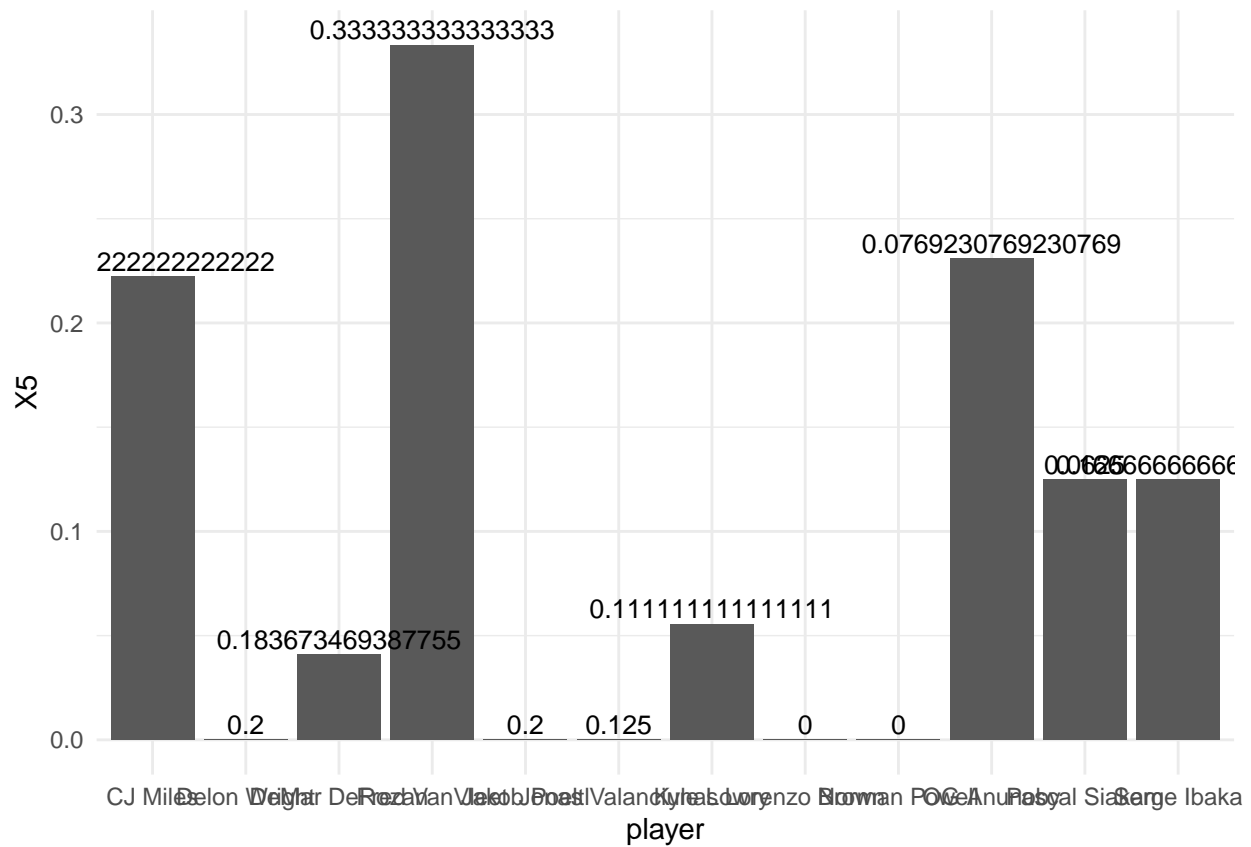
```
ggplot(Xbar.TOR, aes(x=player, y=X3)) +
  geom_bar(stat="identity") +
  geom_text(aes(label=X1), vjust=-0.3, size=3.5) +
  theme_minimal()
```



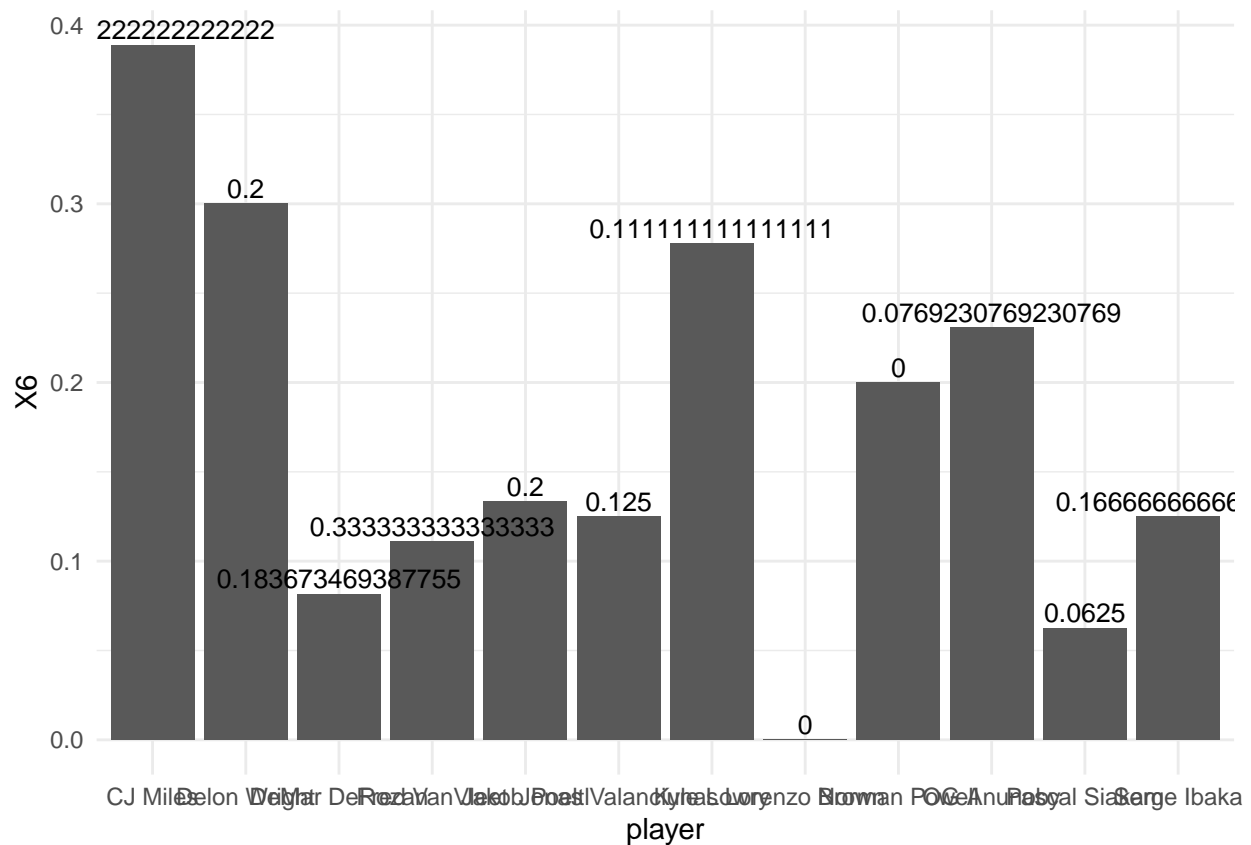
```
ggplot(Xbar.TOR, aes(x=player, y=X4)) +
  geom_bar(stat="identity") +
  geom_text(aes(label=X1), vjust=-0.3, size=3.5) +
  theme_minimal()
```



```
ggplot(Xbar.TOR, aes(x=player, y=X5)) +
  geom_bar(stat="identity") +
  geom_text(aes(label=X1), vjust=-0.3, size=3.5) +
  theme_minimal()
```



```
ggplot(Xbar.TOR, aes(x=player, y=X6)) +
  geom_bar(stat="identity") +
  geom_text(aes(label=X1), vjust=-0.3, size=3.5) +
  theme_minimal()
```



## Hierarchical clustering of NBA players

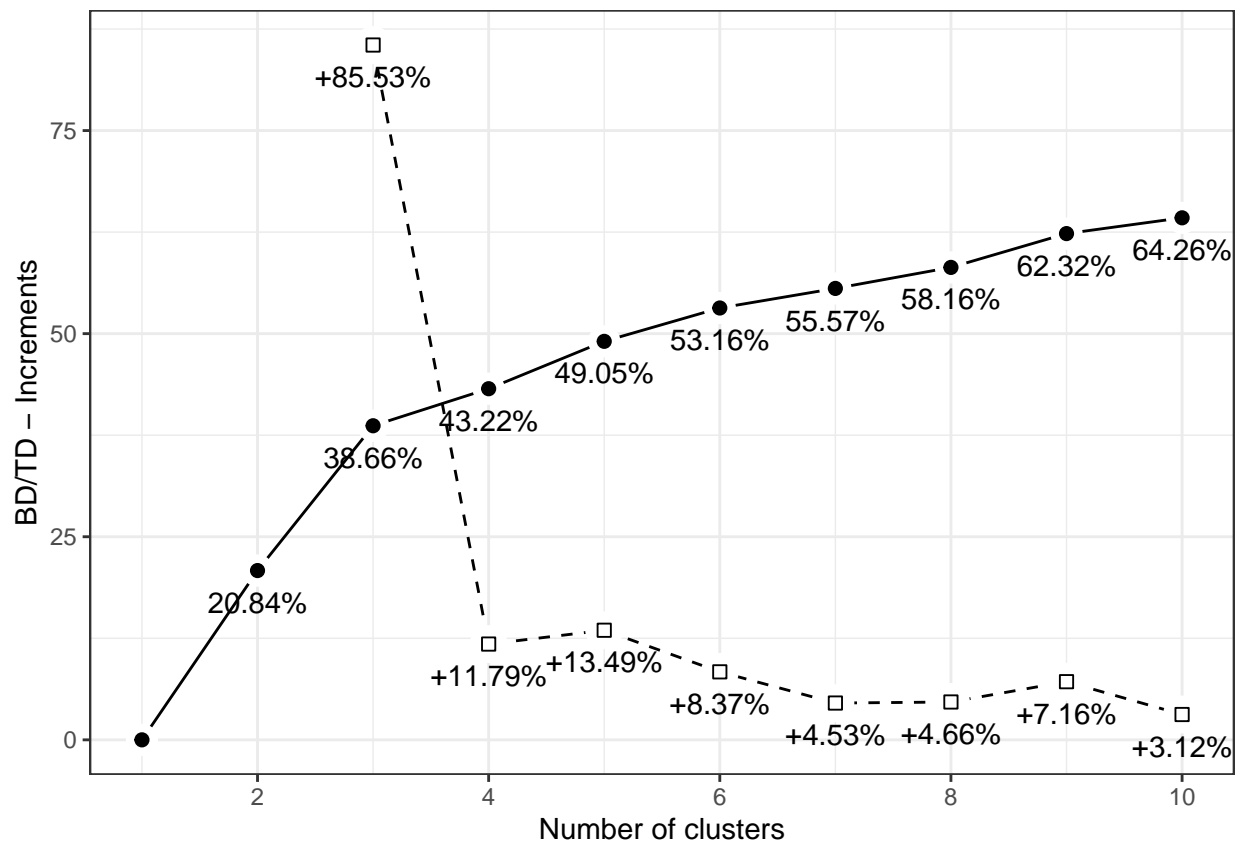
```
rm(list=ls())

attach(Pbox)
df <- data.frame(PTS, P3M, REB=OREB+DREB,
                 AST, TOV, STL, BLK, PF)
detach(Pbox)

df <- subset(df, Pbox$MIN>=1500)

ID <- Pbox$Player[Pbox$MIN>=1500]

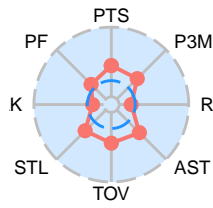
hclu <- hclustering(df)
plot(hclu)
```



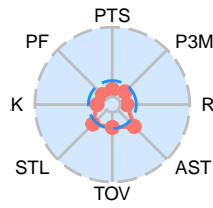
```
hclu2 <- hclustering(df, labels=ID, k=9)
```

```
plot(hclu2, profiles=TRUE)
```

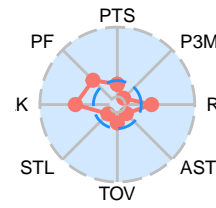
Cluster 1 – CHI = 0.5



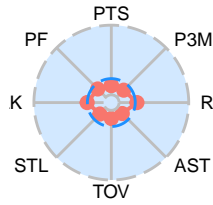
Cluster 2 – CHI = 0.32



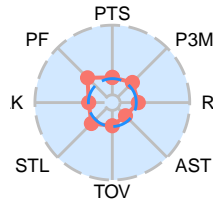
Cluster 3 – CHI = 0.



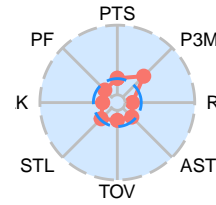
Cluster 4 – CHI = 0.3



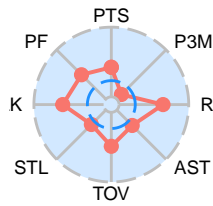
Cluster 5 – CHI = 0.33



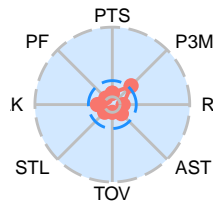
Cluster 6 – CHI = 0.



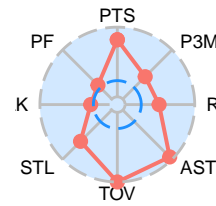
Cluster 7 – CHI = 0.95



Cluster 8 – CHI = 0.19



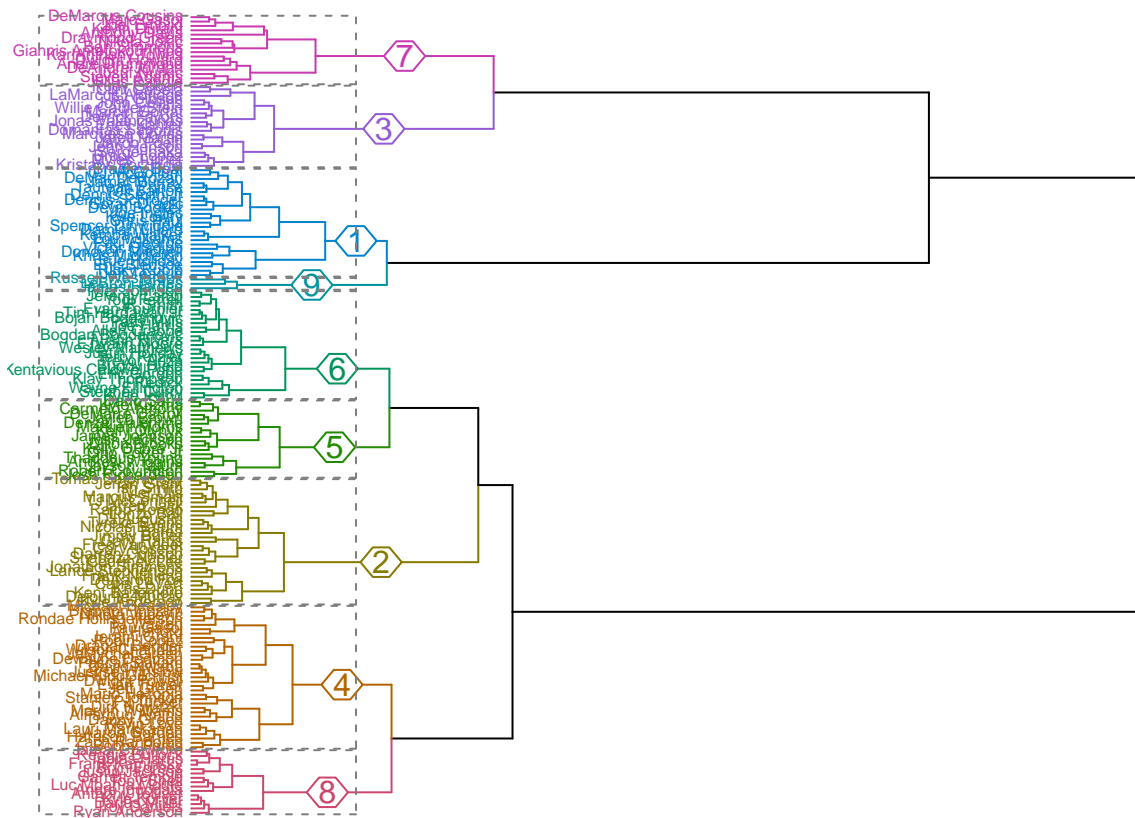
Cluster 9 – CHI = 0.



```
plot(hclu2, rect=TRUE, colored.branches=TRUE,
     cex.labels=0.5)
```

```
## Warning in par(oldmar): argument 1 does not name a graphical parameter
```





```
Pbox.subset <- subset(Pbox, MIN>=1500)

MIN <- Pbox.subset$MIN

X <- data.frame(hclu2$Subjects, scale(df), MIN)

dvar <- c("PTS", "P3M", "REB", "AST",
          "TOV", "STL", "BLK", "PF")
svar <- "MIN"

yRange <- range(X[,dvar])

sizeRange <- c(1500, 3300)

no.clu <- 9

p <- vector(no.clu, mode="list")

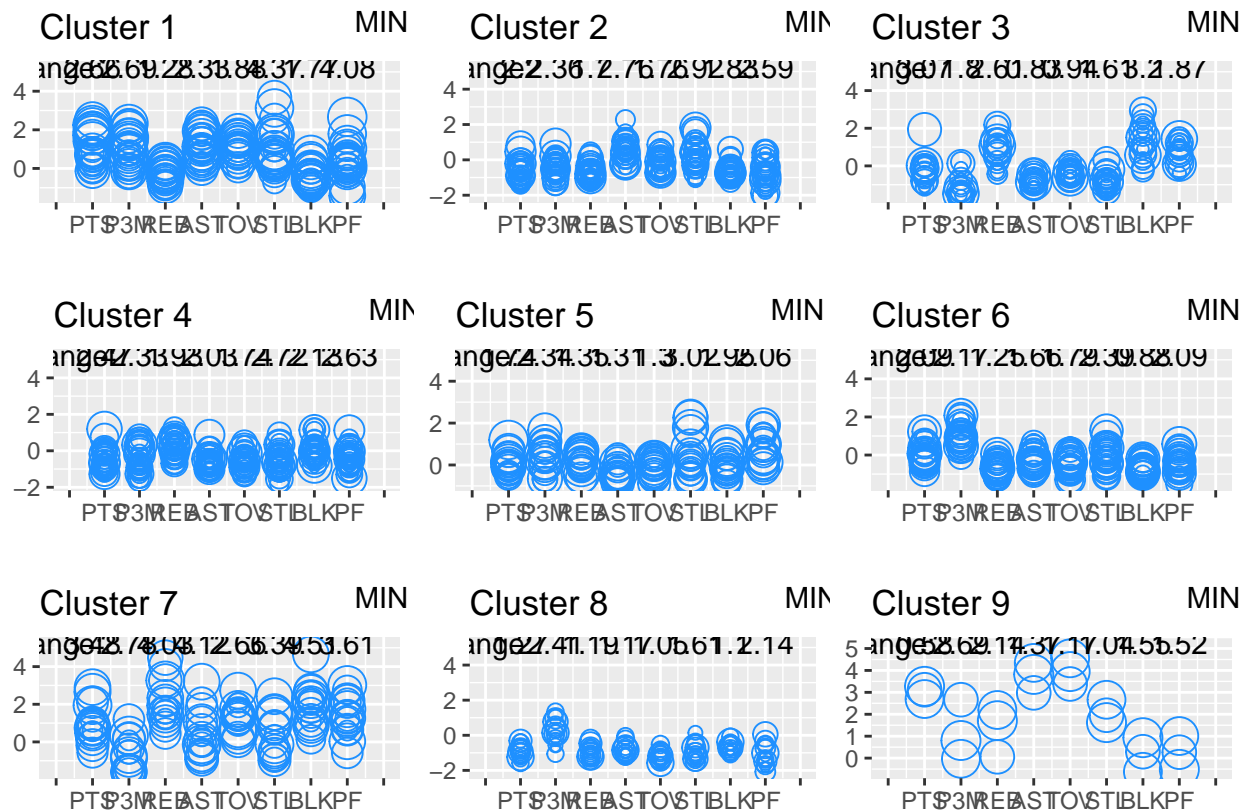
for (k in 1:no.clu) {
  XC <- subset(X, Cluster==k)
  vrb <- variability(XC[,3:11], data.var=dvar,
                    size.var=svar, weight=FALSE,
                    VC=FALSE)
  title <- paste("Cluster", k)
  p[[k]] <- plot(vrb, size.lim=sizeRange, ylim=yRange,
                title=title, leg.pos=c(0,1),
                leg.just=c(-0.5,0),
```

```

leg.box="vertical",
leg.brk=seq(1500,3000,500),
leg.title.pos="left", leg.nrow=1,
max.circle=7)
}

grid.arrange(grobs=p, ncol=3)

```



## LINEAR MODELS

### Simple linear regression

```

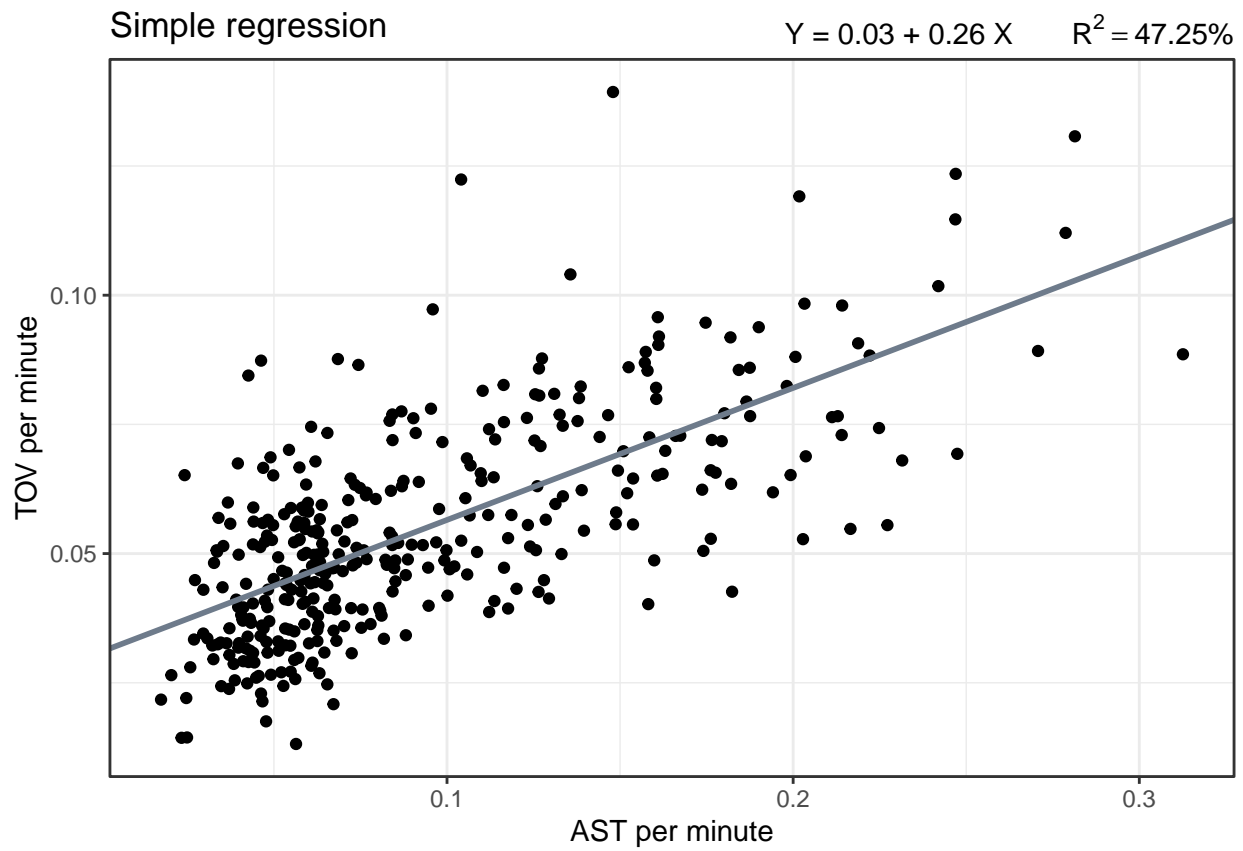
rm(list=ls())

Pbox.sel <- subset(Pbox, MIN>=500)

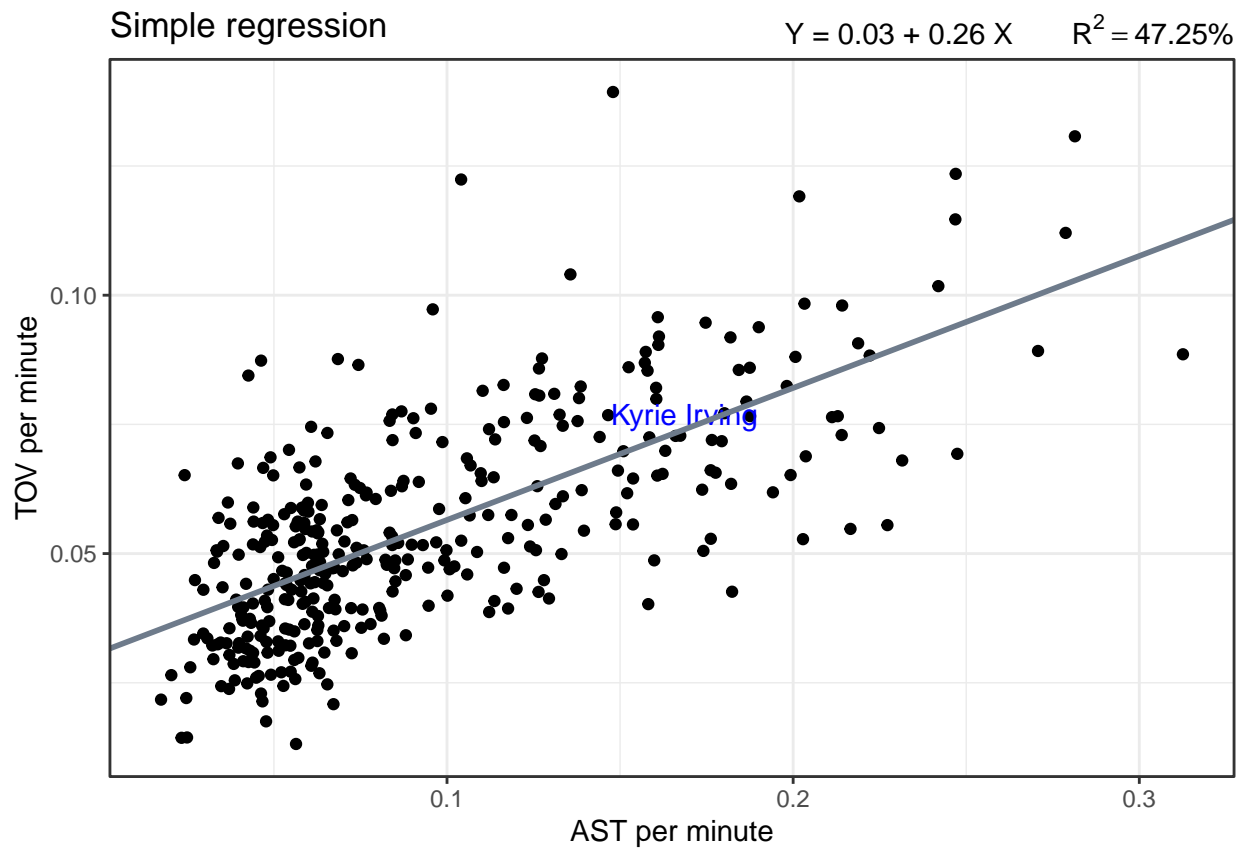
attach(Pbox.sel)
x <- AST/MIN
y <- TOV/MIN
pl <- Player
detach(Pbox.sel)

out <- simplereg(x, y, type="lin")
xtitle <- "AST per minute"
ytitle <- "TOV per minute"
plot(out, xtitle=xtitle, ytitle=ytitle)

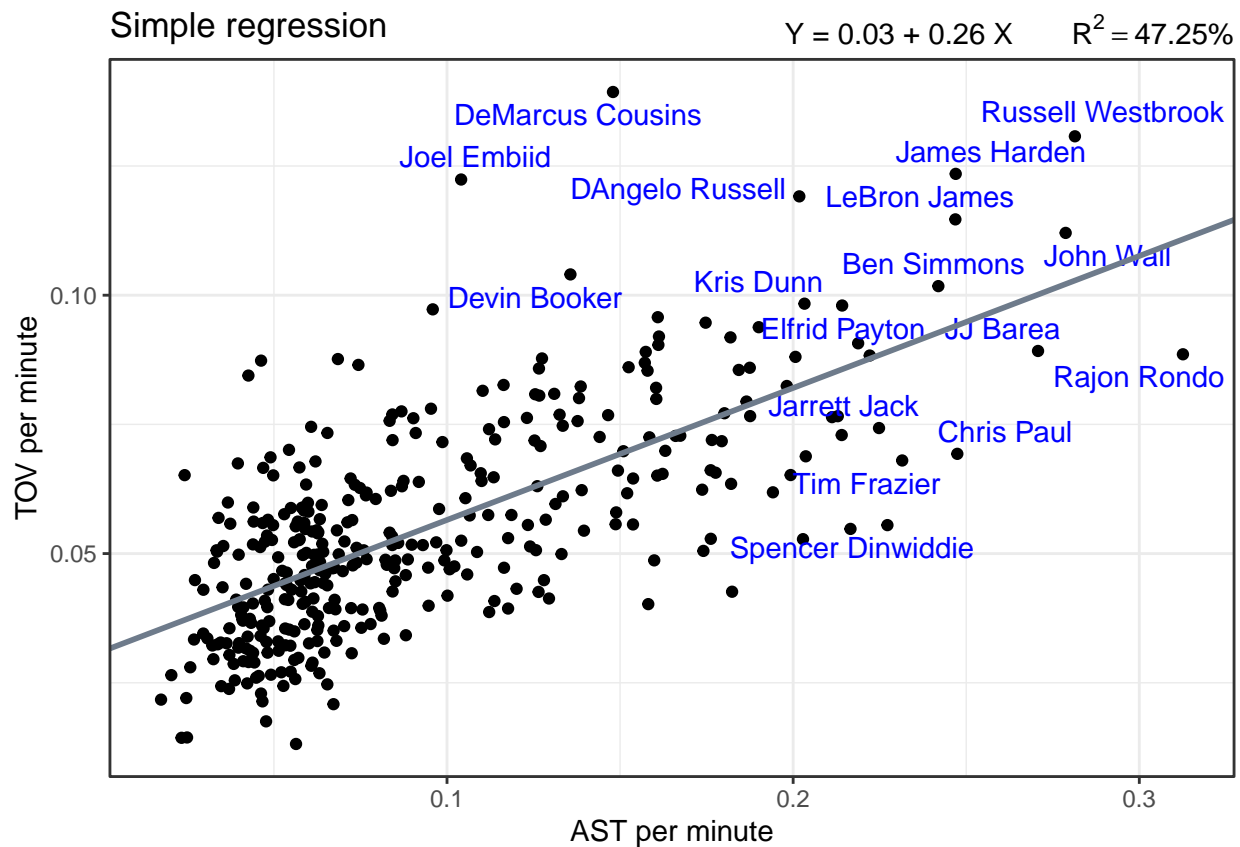
```



```
selp <- which(pl=="Kyrie Irving")
plot(out, labels=pl, subset=selp, xtitle=xtitle,
      ytitle=ytitle)
```



```
plot(out, labels=pl, subset="quant",  
      Lx=0, Ux=0.97, Ly=0, Uy=0.97,  
      xtitle=xtitle, ytitle=ytitle)
```



## Non Parametric Regression

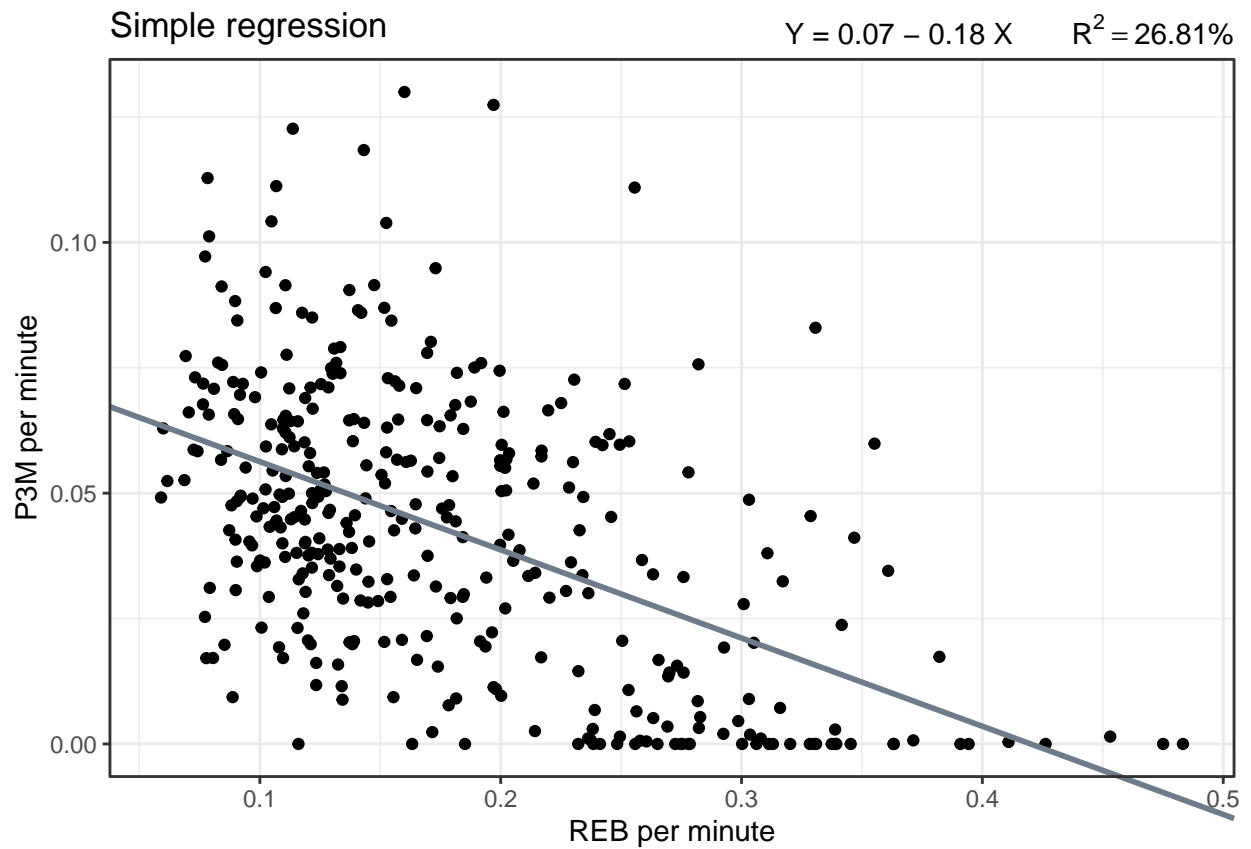
### Polynomial regression

```
rm(list=ls())

Pbox.sel <- subset(Pbox, MIN>=500)

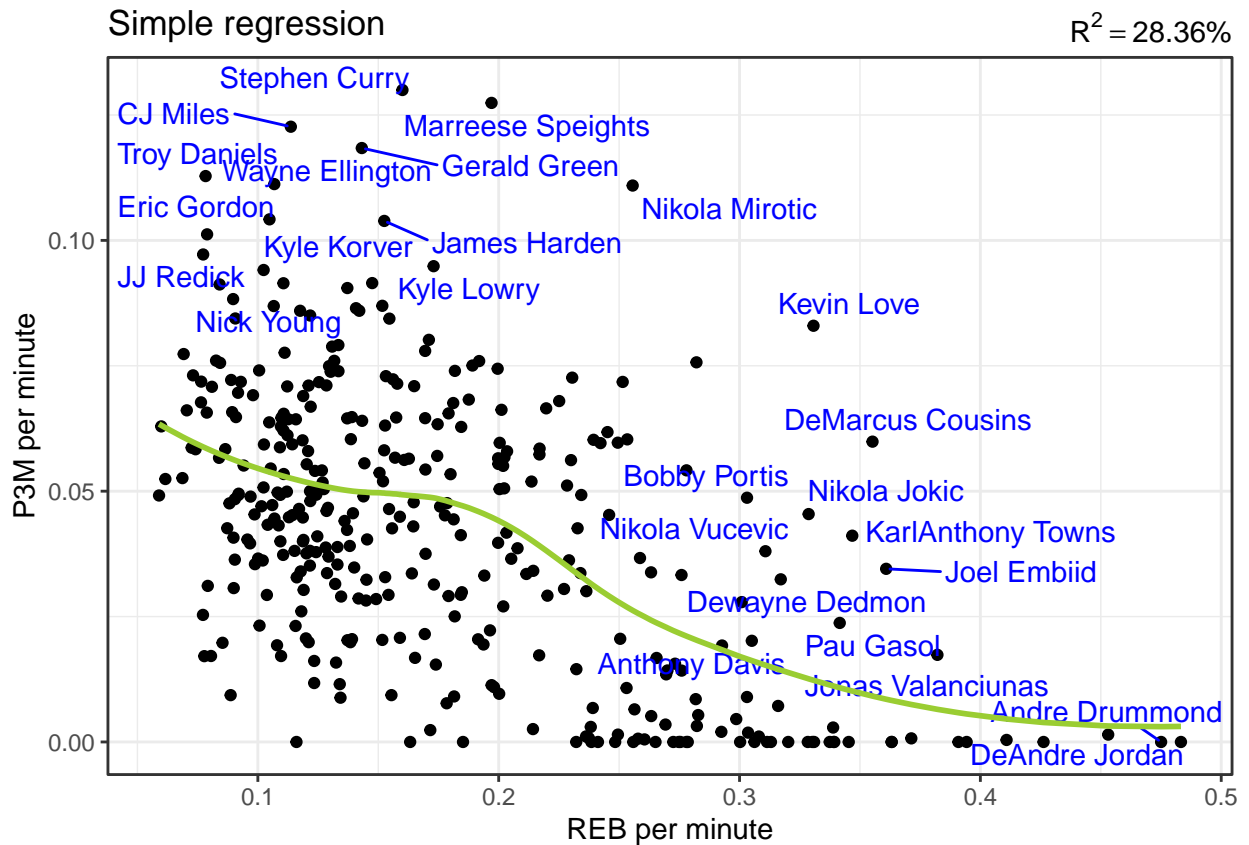
attach(Pbox.sel)
x <- (DREB+OREB)/MIN
y <- P3M/MIN
pl <- Player
detach(Pbox.sel)

out <- simplereg(x, y, type="lin")
xtitle <- "REB per minute"
ytitle <- "P3M per minute"
plot(out, xtitle=xtitle, ytitle=ytitle)
```



```
out <- simplereg(x, y, type="pol")
plot(out, labels=pl, subset="quant",
      Lx=0, Ux=0.90, Ly=0, Uy=0.95,
      xtitle=xtitle, ytitle=ytitle)
```

```
## Warning: ggrepel: 28 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



### Gaussian kernel smoothing

```
data <- subset(Pbox, MIN>=500)
```

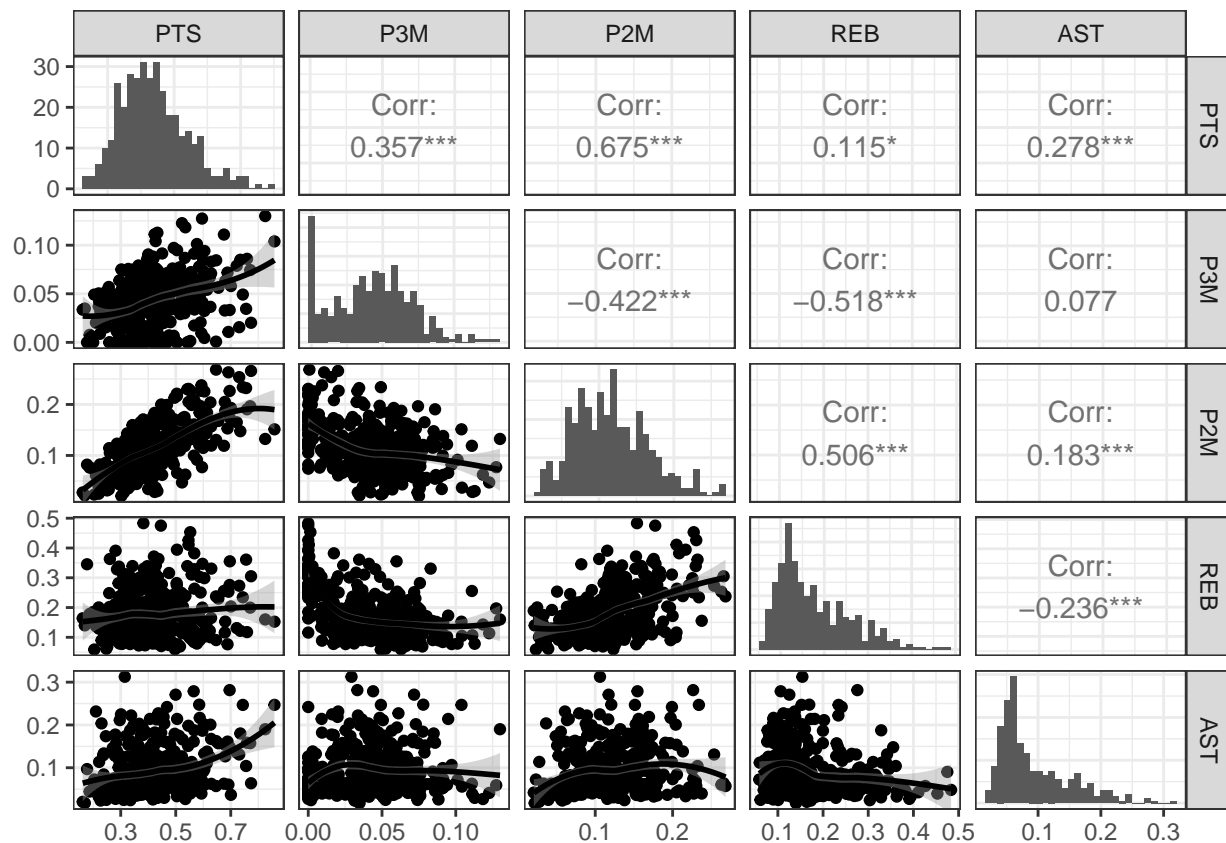
```
attach(data)
```

```
df <- data.frame(PTS, P3M, P2M, REB=OREB+DREB, AST)/MIN
```

```
detach(data)
```

```
scatterplot(df, data.var=1:5,
lower=list(continuous="smooth_loess"),
diag=list(continuous="barDiag"))
```

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



### Estimation of scoring probability

```
PbP <- PbPmanipulation(PbP.BDB)

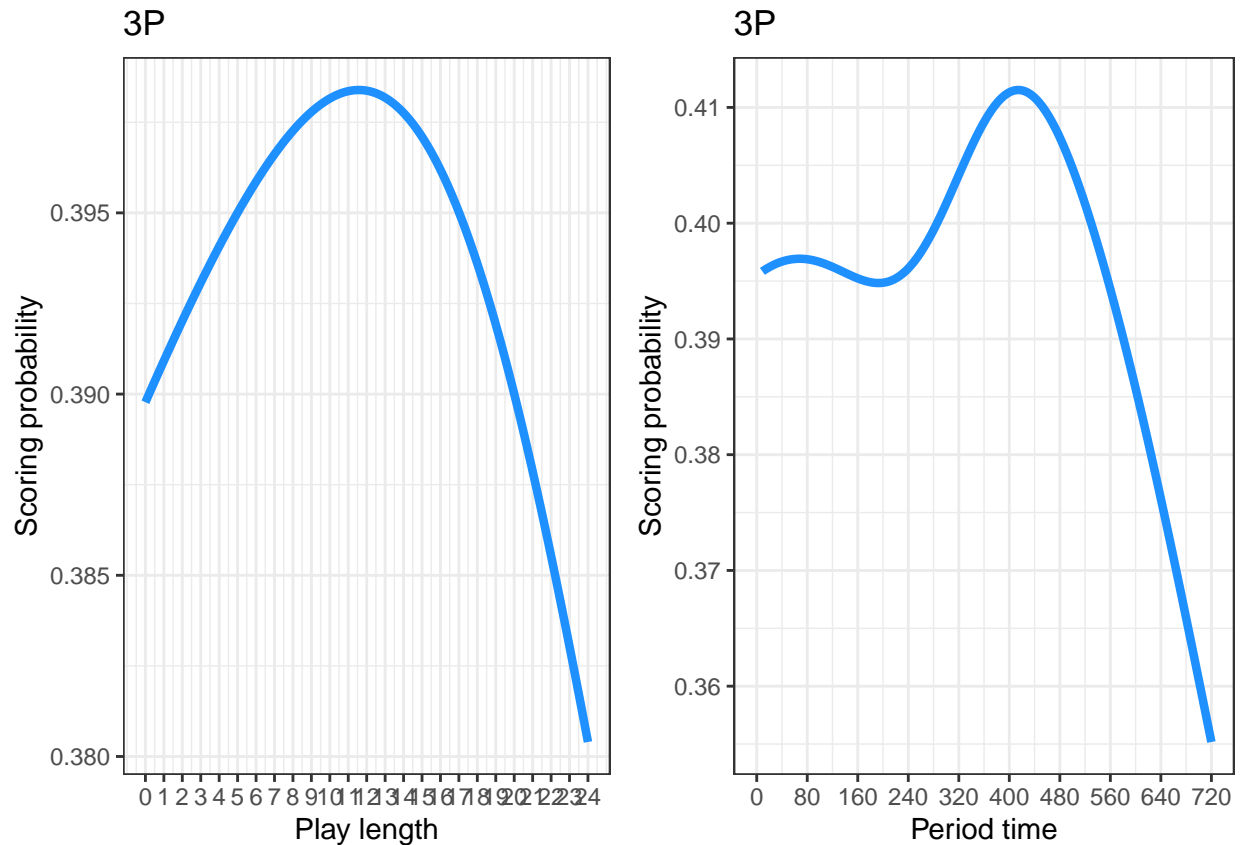
PbP.GSW <- subset(PbP, team=="GSW" & result!="")

p1 <- scoringprob(data=PbP.GSW, shot.type="3P",
                  var="playlength")

p2 <- scoringprob(data=PbP.GSW, shot.type="3P",
                  var="periodTime", bw=300)

grid.arrange(p1, p2, ncol=2)
```





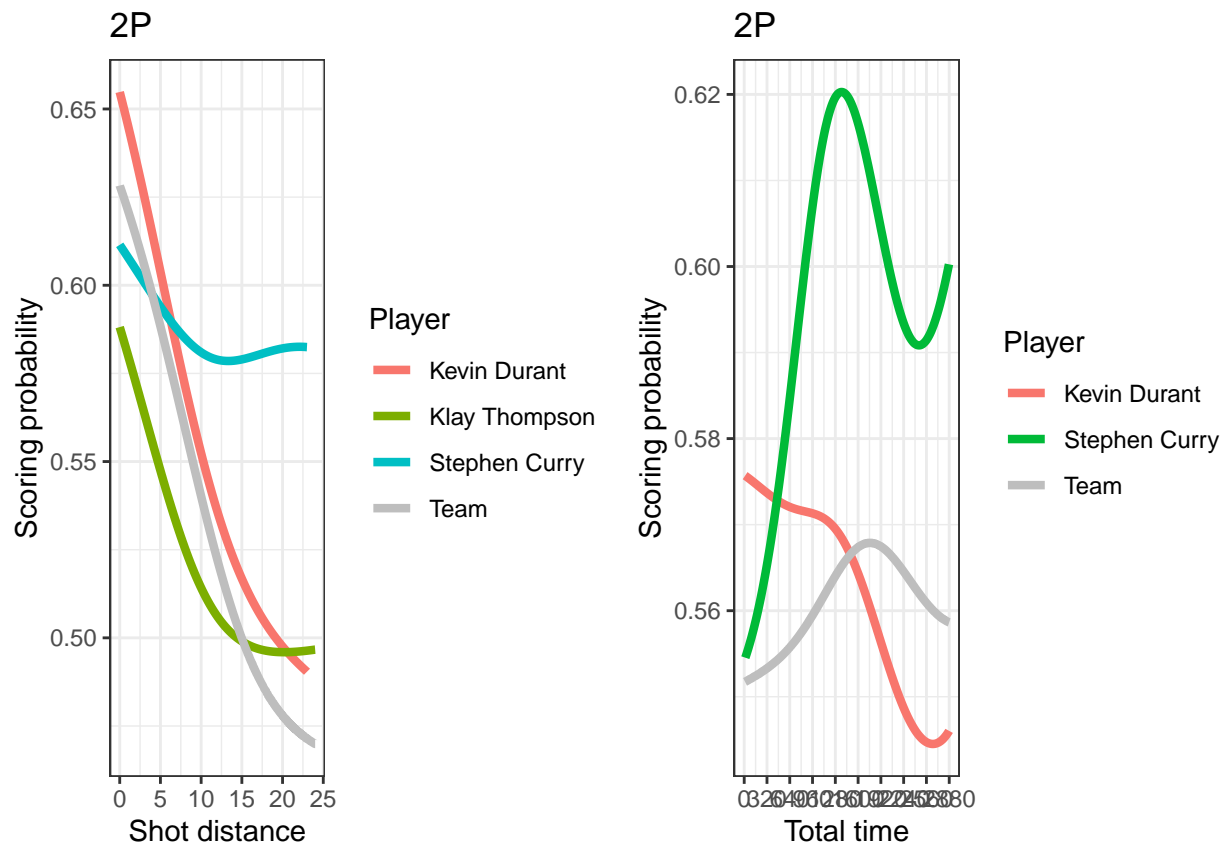
```
p11 <- c("Kevin Durant","Stephen Curry","Klay Thompson")

p1 <- scoringprob(data=PbP.GSW, shot.type="2P",
  players=p11, var="shot_distance",
  col.team="gray")

p12 <- c("Kevin Durant","Stephen Curry")

p2 <- scoringprob(data=PbP.GSW, shot.type="2P",
  players=p12, var="totalTime", bw=1500,
  col.team="gray")

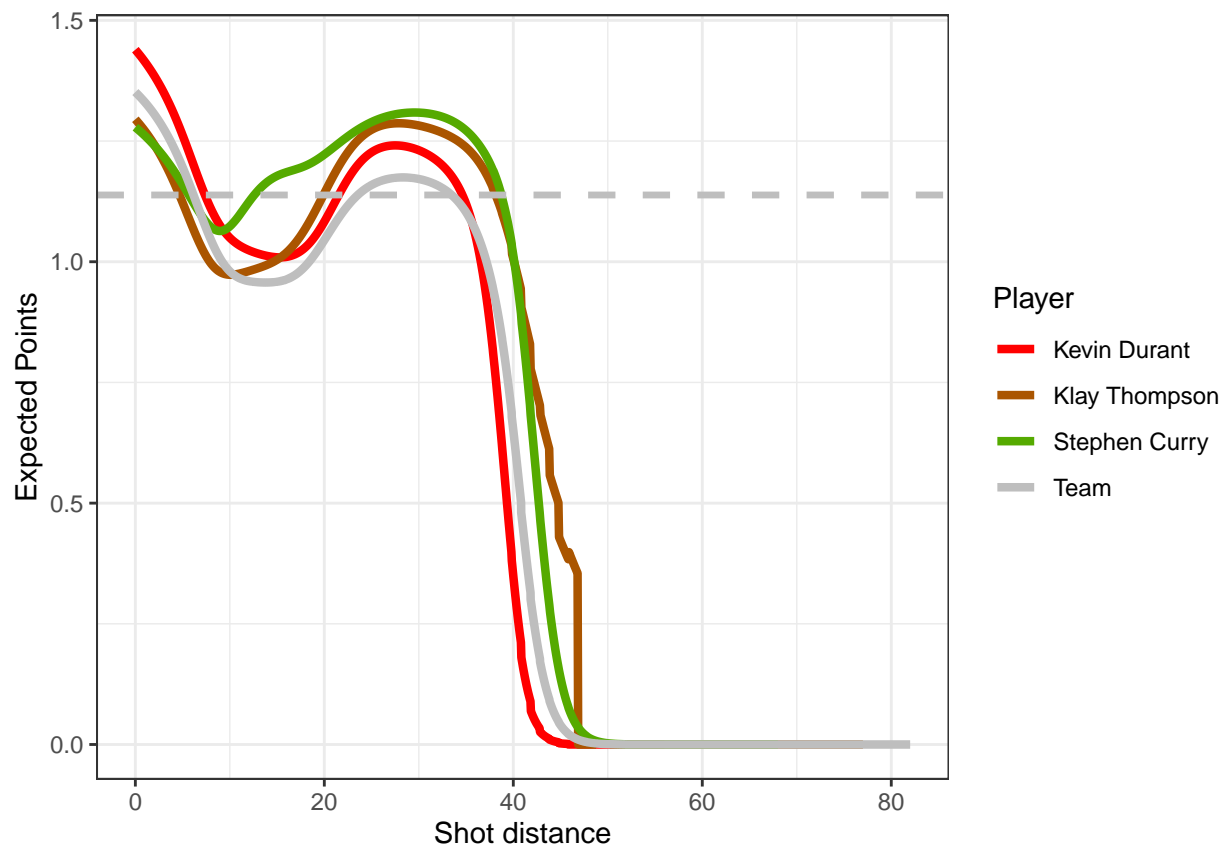
grid.arrange(p1, p2, ncol=2)
```



### Estimation of expected points

```
mypal <- colorRampPalette(c("red","green"))

expectedpts(data=PbP.GSW, players=pl1,
             col.team="gray", palette=mypal,
             col.hline="gray")
```

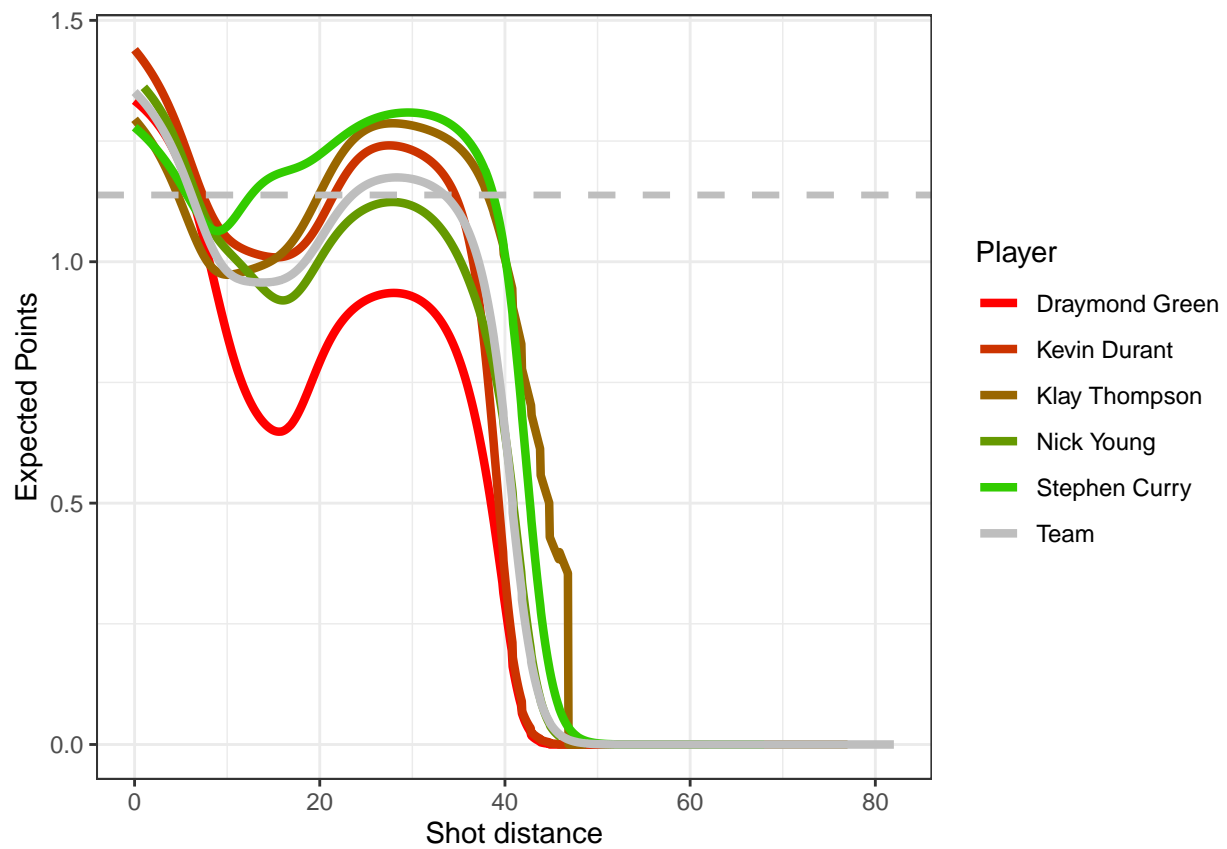


```
Pbox.GSW <- subset(Pbox, PTS>=500 &
  Team=="Golden State Warriors")

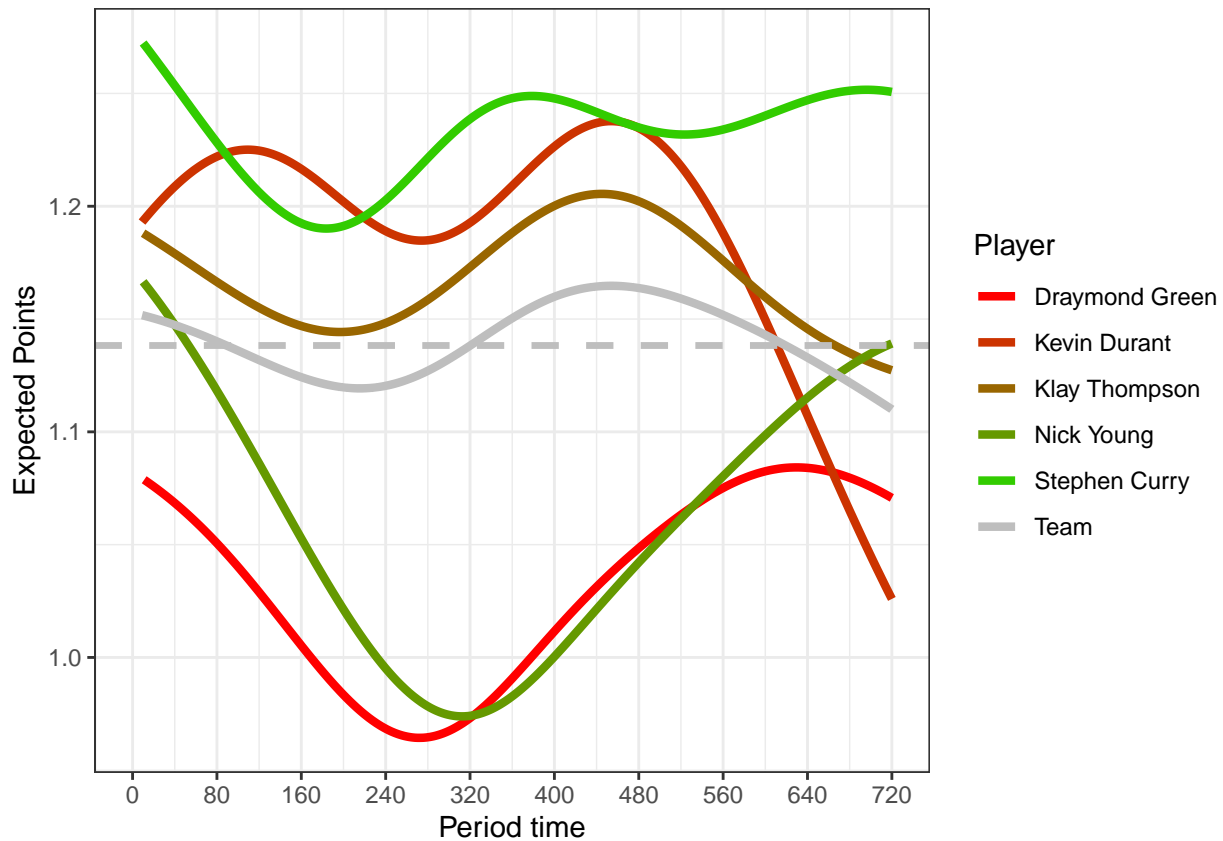
pl.gsw <- Pbox.GSW$Player

mypal <- colorRampPalette(c("red","green"))

expectedpts(data=PbP.GSW, players=pl.gsw,
  col.team="gray", palette=mypal,
  col.hline="gray")
```



```
expectedpts(data=PbP.GSW, bw=300, players=pl.gsw,
  col.team="gray", palette=mypal,
  col.hline="gray", var="periodTime",
  xlab="Period time")
```



```

top <- subset(Tadd, Playoff=="Y" & team!="GSW")$team
bot <- subset(Tadd, Playoff=="N")$team

bot_top <- function(X, k) {
  dts <- subset(subset(X, oppTeam %in% get(k)),
                team="GSW")
  dts$player <- paste(dts$player, k)
  return(dts)
}

PbP.GSW2 <- rbind(bot_top(PbP, "top"), bot_top(PbP, "bot"))

pl.gsw2 <- c("Stephen Curry top", "Stephen Curry bot",
            "Kevin Durant top", "Kevin Durant bot")

mypal <- colorRampPalette(c("red", "green"))

expectedpts(data=PbP.GSW2, bw=1200, players=pl.gsw2,
            col.team="gray", palette=mypal,
            col.hline="gray", var="totalTime",
            xlab="Total time", x.range=NULL)

```

