

```

#####
# April 2020
#
# The following R code allows to replicate all the analyses and examples
# in the book "Basketball Data Science" (by P. Zuccolotto and M. Manisera),
# forthcoming as a CRC Press publication.
# It is based on the "BasketballAnalyzer" package developed with M. Sandri.
# See
# https://bdsports.unibs.it/basketballanalyzer/
# for further explanations and updates
#####

#####
#Warning: If you want to reproduce the figures
#contained in the book 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
#####

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

#####
#####
# CHAPTER 2 #
# Data and Basic Statistical Analyses #
#####
#####
#data(package="BasketballAnalyzer")
#PbP <- PbPmanipulation(PbP.BDB)

#####
# 2.2 BASIC STATISTICAL ANALYSES
#####

#####
# 2.2.1 Pace, Ratings, Four Factors
#####
rm(list=ls())

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

plot(FF.sel)

FF <- fourfactors(Tbox,Obox)
listPlots <- plot(FF)

library(gridExtra)
grid.arrange(grobs=listPlots[1:2], ncol=1)

#####
# 2.2.2 Bar-line plots
#####

```

```

rm(list=ls())

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

Pbox.HR <- subset(Pbox, Team=="Houston Rockets" &
                 MIN>=500)
barline(data=Pbox.HR, id="Player",
        bars=c("P2p", "P3p", "FTp"), line="MIN",
        order.by="PM", labels.bars=c("2P%", "3P%", "FT%"),
        title="Houston Rockets")

#####
# 2.2.3 Radial plots
#####
rm(list=ls())

Pbox.PG <- subset(Pbox, Player=="Russell Westbrook" |
                 Player=="Stephen Curry" |
                 Player=="Chris Paul" |
                 Player=="Kyrie Irving" |
                 Player=="Damian Lillard" |
                 Player=="Kyle Lowry" |
                 Player=="John Wall" |
                 Player=="Rajon Rondo" |
                 Player=="Kemba Walker")
attach(Pbox.PG)
X <- data.frame(P2M, P3M, FTM, REB=OREB+DREB, AST,
                STL, BLK)/MIN
detach(Pbox.PG)
radialprofile(data=X, title=Pbox.PG$Player, std=FALSE)

radialprofile(data=X, title=Pbox.PG$Player, std=TRUE)

#####
# 2.2.4 Scatter plots
#####
rm(list=ls())

Pbox.sel <- subset(Pbox, MIN>= 500)
attach(Pbox.sel)
X <- data.frame(AST, TOV, PTS)/MIN
detach(Pbox.sel)
mypal <- colorRampPalette(c("blue", "yellow", "red"))
scatterplot(X, data.var=c("AST", "TOV"), z.var="PTS",
            labels=1:nrow(X), palette=mypal)

SAS <- which(Pbox.sel$Team=="San Antonio Spurs")
scatterplot(X, data.var=c("AST", "TOV"), z.var="PTS",
            labels=Pbox.sel$Player, palette=mypal,
            subset=SAS)

SAS <- which(Pbox.sel$Team=="San Antonio Spurs")
scatterplot(X, data.var=c("AST", "TOV"), z.var="PTS",

```

```

labels=Pbox.sel$Player, palette=mypal,
subset=SAS, zoom=c(0.20,0.325,0.05,0.10))

#####
# 2.2.5 Bubble plots
#####
rm(list=ls())

attach(Tbox)
X <- 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(X, id="T", x="P2p", y="P3p", col="FTp",
           size="AS", labels=labs)

Pbox.GSW.CC <- subset(Pbox,
                    (Team=="Golden State Warriors" |
                     Team=="Cleveland Cavaliers") &
                    MIN>=500)
attach(Pbox.GSW.CC)
X <- data.frame(ID=Player, Team, V1=DREB/MIN, V2=STL/MIN,
                V3=BLK/MIN, V4=MIN)
detach(Pbox.GSW.CC)
labs <- c("Defensive Rebounds", "Steals", "Blocks",
          "Total minutes played")
bubbleplot(X, id="ID", x="V1", y="V2", col="V3",
           size="V4", text.col="Team", labels=labs,
           title="GSW and CC during the regular season",
           text.legend=TRUE, text.size=3.5, scale=FALSE)

#####
# 2.2.6 Variability analysis
#####
rm(list=ls())

Pbox.OKC <- subset(Pbox, Team=="Oklahoma City Thunder"
                  & MIN>=500)
vrbl <- variability(data=Pbox.OKC, data.var="P3p",
                  size.var="P3A")

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

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

plot(vrbl, title="Variability diagram - OKC")

#####
# 2.2.7 Inequality analysis
#####

```

```

rm(list=ls())

Pbox.BN <- subset(Pbox, Team=="Brooklyn Nets")
ineqBN <- inequality(Pbox.BN$PTS, nplayers=8)
Pbox.MB <- subset(Pbox, Team=="Milwaukee Bucks")
ineqMB <- inequality(Pbox.MB$PTS, nplayers=8)
library(gridExtra)
p1 <- plot(ineqBN, title="Brooklyn Nets")
p2 <- plot(ineqMB, title="Milwaukee Bucks")
grid.arrange(p1, p2, nrow=1)

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

#####
rm(list=ls())
PbP <- PbPmanipulation(PbP.BDB)

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

lineup <- c("Stephen Curry", "Kevin Durant",
           "Klay Thompson", "Draymond Green",
           "Zaza Pachulia")
filt5 <- apply(PbP.GSW[, 4:13], 1,
              function(x) {
                x <- as.character(x)
                sum(x %in% lineup)==5
              })

subPbP.GSW <- PbP.GSW[filt5, ]
PTS5 <- sapply(lineup,
              function(x) {
                filt <- subPbP.GSW$player==x
                sum(subPbP.GSW$points[filt], na.rm=T)
              })

inequality(PTS5, nplayer=5)

#####
rm(list=ls())
PbP <- PbPmanipulation(PbP.BDB)

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

lineup <- c("Stephen Curry", "Kevin Durant",
           "Klay Thompson", "Draymond Green",

```

```

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

subPbP.GSW.DET <- PbP.GSW.DET[filt5, ]
PTS5 <- sapply(lineup,
              function(x) {
                filt <- subPbP.GSW.DET$player==x
                sum(subPbP.GSW.DET$points[filt], na.rm=T)
              })

inequality(PTS5,nplayer=5)

#####
# 2.2.8 Shot charts
#####
rm(list=ls())
PbP <- PbPmanipulation(PbP.BDB)

subdata <- subset(PbP, player=="Kevin Durant")
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)

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")

#####
#####
# CHAPTER 3 #
# Discovering Patterns in Data #
#####
#####

#####
# 3.1 DETECTING ASSOCIATIONS BETWEEN VARIABLES
#####

#####
# 3.1.1 Statistical dependence
#####
rm(list=ls())
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)

library(vcd)
assocstats(T)

#####
# 3.1.2 Mean dependence
#####
rm(list=ls())
library(dplyr)
library(lsr)
library(tibble)
FF <- fourfactors(Tbox, Obox)
attach(Tbox)
attach(FF)
X <- data.frame(PTS, P2M, P3M, FTM, REB=OREB+DREB, AST,
               STL, BLK, ORtg, DRtg)
detach(Tbox)
detach(FF)
Playoff <- Tadd$Playoff
eta <- sapply(X, 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')

#####
# 3.1.3 Correlation
#####
rm(list=ls())
data <- subset(Pbox, MIN>=500)
attach(data)
X <- data.frame(AST, TOV)/MIN
detach(data)

cor(X$AST, X$TOV)
cor(rank(X$AST), rank(X$TOV))
cor(X$AST, X$TOV, method="spearman")
cor(X)

#####
# 3.2 ANALYZING PAIRWISE LINEAR CORRELATION AMONG VARIABLES
#####
rm(list=ls())
data <- merge(Pbox, Tadd, by="Team")

```

```

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

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

corrmatrix <- corranalysis(X[,1:8], threshold=0.5)
plot(corrmatrix)

scatterplot(X, data.var=1:8, z.var="Playoff",
            diag=list(continuous="blankDiag"))

#####
# 3.3 DISPLAYING INDIVIDUAL CASES ACCORDING TO THEIR SIMILARITY
#####
rm(list=ls())

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

mds <- MDSmap(data)
plot(mds, labels=id)

selp <- which(id=="Al Horford" | id=="Kyle Korver" |
             id=="Myles Turner" | id=="Kyle Kuzma" |
             id=="Andrew Wiggins")

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

plot(mds, labels=id, subset=selp, col.subset="tomato",
     zoom=c(0,3,0,2))

plot(mds, z.var=c("P2M", "P3M", "AST", "REB"),
     level.plot=FALSE, palette=topo.colors)

plot(mds, z.var=c("P2M", "P3M", "AST", "REB"),
     contour=TRUE, palette=topo.colors)

#####
# 3.4 ANALYZING NETWORK RELATIONSHIPS
#####
rm(list=ls())
PbP <- PbPmanipulation(PbP.BDB)
PbP.GSW <- subset(PbP, team=="GSW")

netdata <- assistnet(PbP.GSW)
netdata
#RNGkind(sample.kind="Rounding")
set.seed(7)
plot(netdata)

plot(netdata, layout="circle", edge.thr=20)

```

```

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$points)
p10 <- mean(PbP.GSW.DG0$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$points)
p11 <- mean(PbP.GSW.DG1$playlength)

plot(netdata, layout="circle", edge.thr=20,
     node.col="FGPTS_AST", node.size="ASTPTS")
plot(netdata, layout="circle", edge.thr=20,
     node.col="FGPTS", node.size="FGPTS_ASTp")

#####
TAB <- netdata$assistTable
X <- netdata$nodeStats

names(X)[1] <- "Player"
data <- merge(X, Pbox, by="Player")

mypal <- colorRampPalette(c("blue", "yellow", "red"))
scatterplot(data, data.var=c("FGM", "FGM_ASTp"),
            z.var="MIN", labels=data$Player,
            palette=mypal, repel_labels=TRUE)

#####
sel <- which(data$MIN > 984)
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)
}

library(gridExtra)
grid.arrange(grobs=pM, nrow=2)
grid.arrange(grobs=pR, nrow=2)

```

```

#####
library(vcd)
assocstats(tab)

#####
XX <- data.frame(X[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)

#####
library(tidygraph)
library(igraph)
library(CINNA)
net1 <- as_tbl_graph(netdata$assistNet)
class(net1) <- "igraph"
centr_degree(net1)
alpha_centrality(net1)
closeness(net1, mode="all")
betweenness(net1)
calculate_centralities(net1)

#####
# 3.5 ESTIMATING DENSITY OF EVENTS
#####

#####
# 3.5.1 Density with respect to a concurrent variable
#####

rm(list=ls())
PbP <- PbPmanipulation(PbP.BDB)

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

densityplot(data=data.team, shot.type="2P",
            var="periodTime", best.scorer=TRUE)
densityplot(data=data.team, shot.type="2P",
            var="totalTime", best.scorer=TRUE)
densityplot(data=data.team, shot.type="2P",
            var="playlength", best.scorer=TRUE)
densityplot(data=data.team, 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)
densityplot(data=data.opp, shot.type="2P",
            var="shot_distance", best.scorer=TRUE)

KD <- subset(PbP, player=="Kevin Durant" & result!="")

```

```

SC <- subset(PbP, player=="Stephen Curry" & result!="")
densityplot(data=KD, shot.type="field",
  var="playlength")
densityplot(data=KD, shot.type="field",
  var="shot_distance")
densityplot(data=SC, shot.type="field",
  var="playlength")
densityplot(data=SC, shot.type="field",
  var="shot_distance")

#####
# 3.5.2 Density in space
#####
rm(list=ls())
PbP <- PbPmanipulation(PbP.BDB)

PbP$xx <- PbP$original_x/10
PbP$yy <- PbP$original_y/10 - 41.75

KT <- subset(PbP, player=="Klay Thompson")

shotchart(data=KT, x="xx", y="yy",
  type="density-polygons")
shotchart(data=KT, x="xx", y="yy", type="density-raster")
shotchart(data=KT, x="xx", y="yy", type="density-hexbin")

shotchart(data=KT, x="xx", y="yy",
  type="density-polygons", scatter=TRUE)
shotchart(data=KT, x="xx", y="yy", type="density-raster",
  scatter=TRUE, pt.col="tomato", pt.alpha=0.1)
shotchart(data=KT, x="xx", y="yy", type="density-hexbin",
  nbins=50, palette="bwr")

#####
# 3.5.3 Joint density of two variables
#####
rm(list=ls())
data <- subset(Pbox, MIN>=500)
attach(data)
X <- data.frame(PTS, P3M, P2M, REB=OREB+DREB, AST)/MIN
detach(data)

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

#####
#####
# CHAPTER 4 #
# Findings Groups in Data #
#####
#####

#####
# 4.2 CLUSTER ANALYSIS
#####

```

```
#####
# 4.2.1 k-means clustering of NBA teams
#####
rm(list=ls())

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)

#RNGkind(sample.kind="Rounding")
set.seed(29)
kclu1 <- kclustering(data)
plot(kclu1)

set.seed(29)
kclu2 <- kclustering(data, labels=Tbox$Team, k=5)
plot(kclu2)

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)

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)

#####
# 4.2.1 k-means clustering of Golden State Warriors shots
#####
rm(list=ls())
PbP <- PbPmanipulation(PbP.BDB)

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

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

```

#RNGkind(sample.kind="Rounding")
set.seed(1)
kclu1 <- kclustering(data, algorithm="MacQueen",
                    nclumax=15, iter.max=500)
plot(kclu1)

set.seed(1)
kclu2 <- kclustering(data, algorithm="MacQueen",
                    iter.max=500, k=6)
plot(kclu2)

#####
cluster <- as.factor(kclu2$Subjects$Cluster)
shots <- data.frame(shots, cluster)
shots$xx <- shots$original_x/10
shots$yy <- shots$original_y/10 - 41.75

no.clu <- 6
p1 <- p2 <- vector(no.clu, mode="list")
for (k in 1:no.clu) {
  shots.k <- subset(shots, 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)
}

library(gridExtra)
grid.arrange(grobs=p1, nrow=3)
grid.arrange(grobs=p2, nrow=3)

#####
shots.pl <- table(shots$player, shots$cluster)
Xineq <- as.data.frame.matrix(shots.pl)

no.clu <- 6
p <- vector(no.clu, mode="list")
for (k in 1:no.clu) {
  ineqC <- inequality(Xineq[,k], npl=nrow(Xineq))
  title <- paste("Cluster", k)
  p[[k]] <- plot(ineqC, title=title)
}

library(gridExtra)
grid.arrange(grobs=p, nrow=3)

#####
shots.perc <- shots.pl/rowSums(shots.pl)
Xbar <- 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, id="player", line="FGA",

```

```

bars=c("X1", "X2", "X3", "X4", "X5", "X6"),
order.by="FGA", label.line="Field goals attempted",
labels.bars=labclusters)

#####
# 4.2.3 Hierarchical clustering of NBA players
#####
rm(list=ls())

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

data <- subset(data, Pbox$MIN>=1500)
ID <- Pbox$Player[Pbox$MIN>=1500]

hclu1 <- hclustering(data)
plot(hclu1)

hclu2 <- hclustering(data, labels=ID, k=9)
plot(hclu2, profiles=TRUE)

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

####
Pbox.subset <- subset(Pbox, MIN>=1500)
MIN <- Pbox.subset$MIN
X <- data.frame(hclu2$Subjects, scale(data), 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)
}
library(gridExtra)
grid.arrange(grobs=p, ncol=3)

#####
#####
# CHAPTER 5 #
# Modeling Relationships in Data #

```

```
#####
#####

#####
# 5.1 LINEAR MODELS
#####

#####
# 5.1.1 Simple linear regression model
#####
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=X, y=Y, type="lin")
xtitle <- "AST per minute"
ytitle <- "TOV per minute"
plot(out, xtitle=xtitle, ytitle=ytitle)

selp <- which(Pl=="Damian Lillard")
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)

#####
# 5.2 NONPARAMETRIC REGRESSION
#####

#####
# 5.2.1 Polynomial local 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=X, y=Y, type="lin")
xtitle <- "REB per minute"
ytitle <- "P3M per minute"
plot(out, xtitle=xtitle, ytitle=ytitle)

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

#####
# 5.2.2 Gaussian kernel smoothing
```

```
#####
rm(list=ls())

data <- subset(Pbox, MIN>=500)
attach(data)
X <- data.frame(PTS, P3M, P2M, REB=OREB+DREB, AST)/MIN
detach(data)

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

#####
# 5.2.2.1 Estimation of scoring probability
#####
rm(list=ls())
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)
library(gridExtra)
grid.arrange(p1, p2, ncol=2)

p11 <- c("Kevin Durant", "Draymond Green", "Klay Thompson")
p1 <- scoringprob(data=PbP.GSW, shot.type="2P",
players=p11, var="shot_distance",
col.team="gray")
p12 <- c("Kevin Durant", "Draymond Green")
p2 <- scoringprob(data=PbP.GSW, shot.type="2P",
players=p12, var="totalTime", bw=1500,
col.team="gray")
library(gridExtra)
grid.arrange(p1, p2, ncol=2)

#####
# 5.2.2.2 Estimation of expected points
#####
rm(list=ls())
PbP <- PbPmanipulation(PbP.BDB)

PbP.GSW <- subset(PbP, team=="GSW")
pl <- c("Stephen Curry", "Kevin Durant")
mypal <- colorRampPalette(c("red", "green"))
expectedpts(data=PbP.GSW, players=pl,
col.team="gray", palette=mypal,
col.hline="gray")

Pbox.GSW <- subset(Pbox, PTS>=500 &
Team=="Golden State Warriors")
pl <- Pbox.GSW$Player
mypal <- colorRampPalette(c("red", "green"))
expectedpts(data=PbP.GSW, players=pl,
col.team="gray", palette=mypal,
col.hline="gray")
```

```

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

#####
rm(list=ls())
PbP <- PbPmanipulation(PbP.BDB)

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.GSW <- rbind(bot_top(PbP, "top"),
                 bot_top(PbP, "bot"))
pl <- c("Stephen Curry top", "Stephen Curry bot",
        "Kevin Durant top", "Kevin Durant bot")
mypal <- colorRampPalette(c("red", "green"))
expectedpts(data=PbP.GSW, bw=1200, players=pl,
             col.team="gray", palette=mypal,
             col.hline="gray", var="totalTime",
             xlab="Total time", x.range=NULL)

#####
# CHAPTER 6 #
# The R package BasketballAnalyzer #
#####

#####
# 6.2 PREPARING DATA
#####
rm(list=ls())

dts <- read.csv(file="2012-18_teamBoxScore.csv")
dts$gmDate <- as.Date(as.character(dts$gmDate))
year <- as.numeric(format(dts$gmDate, "%Y"))
month <- as.numeric(format(dts$gmDate, "%m"))
dts$season <- ifelse(month<5, paste0(year-1, "-", year),
                    paste0(year, "-", year+1))

library(dplyr)
Tbox2 <- dts %>%
  group_by(season, teamAbbr) %>%
  summarise(GP=n(), MIN=sum(round(teamMin/5)),
            PTS=sum(teamPTS),
            W=sum(teamRslt=="Win"), L=sum(teamRslt=="Loss"),
            P2M=sum(team2PM), P2A=sum(team2PA), P2p=P2M/P2A,
            P3M=sum(team3PM), P3A=sum(team3PA), P3p=P3M/P3A,
            FTM=sum(teamFTM), FTA=sum(teamFTA), FTp=FTM/FTA,
            OREB=sum(teamORB), DREB=sum(teamDRB), AST=sum(teamAST),

```

```

    TOV=sum(teamTO), STL=sum(teamSTL), BLK=sum(teamBLK),
    PF=sum(teamPF), PM=sum(teamPTS-opptPTS) %>%
  rename(Season=season, Team=teamAbbr) %>%
  as.data.frame()

Obox2 <- dts %>%
  group_by(season, teamAbbr) %>%
  summarise(GP=n(), MIN=sum(round(opptMin/5)),
    PTS=sum(opptPTS),
    W=sum(opptRslt=="Win"), L=sum(opptRslt=="Loss"),
    P2M=sum(oppt2PM), P2A=sum(oppt2PA), P2p=100*P2M/P2A,
    P3M=sum(oppt3PM), P3A=sum(oppt3PA), P3p=100*P3M/P3A,
    FTM=sum(opptFTM), FTA=sum(opptFTA), FTp=100*FTM/FTA,
    OREB=sum(opptORB), DREB=sum(opptDRB), AST=sum(opptAST),
    TOV=sum(opptTO), STL=sum(opptSTL), BLK=sum(opptBLK),
    PF=sum(opptPF), PM=sum(teamPTS-opptPTS) %>%
  rename(Season=season, Team=teamAbbr) %>%
  as.data.frame()

dts <- read.csv(file="2012-18_playerBoxScore.csv",
  encoding="UTF-8")
dts$gmDate <- as.Date(as.character(dts$gmDate))
year <- as.numeric(format(dts$gmDate,"%Y"))
month <- as.numeric(format(dts$gmDate,"%m"))
dts$season <- ifelse(month<5, paste0(year-1,"-",year),
  paste0(year,"-",year+1))

Pbox2 <- dts %>%
  group_by(season, teamAbbr, playDispNm) %>%
  summarise(GP=n(), MIN=sum(playMin), PTS=sum(playPTS),
    P2M=sum(play2PM), P2A=sum(play2PA), P2p=100*P2M/P2A,
    P3M=sum(play3PM), P3A=sum(play3PA), P3p=100*P3M/P3A,
    FTM=sum(playFTM), FTA=sum(playFTA), FTp=100*FTM/FTA,
    OREB=sum(playORB), DREB=sum(playDRB), AST=sum(playAST),
    TOV=sum(playTO), STL=sum(playSTL), BLK=sum(playBLK),
    PF=sum(playPF)) %>%
  rename(Season=season, Team=teamAbbr,
    Player=playDispNm) %>%
  as.data.frame()

#####
# 6.3 CUSTOMIZING PLOTS
#####
rm(list=ls())

Pbox.sel <- subset(Pbox, MIN>=500)
attach(Pbox.sel)
X <- data.frame(AST, TOV, PTSpm=PTS)/MIN
detach(Pbox.sel)
mypal <- colorRampPalette(c("blue","yellow","red"))
p1 <- scatterplot(X, data.var=c("AST","TOV"),
  z.var="PTSpM", palette=mypal)

print(p1)
class(p1)

p2 <- p1 +
  labs(title="Scatter plot", x="Assists",
    y="Turnovers") +
  scale_x_continuous(breaks=seq(0,0.35,0.05),
    limits=c(0,0.35)) +
  theme(panel.background=element_rect(fill="#FFCCCC20",

```

```

        colour="red", size=3)) +
  guides(color=FALSE)
print(p2)

p3 <- p1 +
  geom_segment(x=0.225, y=0.025, xend=X$AST[143]+0.005,
              yend=X$TOV[143]-0.001, size=1,
              color="red",
              arrow=arrow(length=unit(0.25, "cm"),
                          type="closed", angle=20)) +
  annotate("text", x=0.225, y=0.025,
         label=Pbox.sel[143,"Player"],
         color="red", fontface=2, hjust=0)
print(p3)

p3 + geom_rect(xmin=0.2, ymin=0.075,
              xmax=Inf, ymax=Inf,
              fill="#DDDDDDAA", color=NA)

p3$layers <- c(geom_rect(xmin=0.2, ymin=0.075,
                       xmax=Inf, ymax=Inf,
                       fill="#DDDDDDAA", color=NA),
              p3$layers)
print(p3)

library(grid)
library(jpeg)
URL <- "https://goo.gl/WGk6J1"
download.file(URL,"NBALogo.jpg", mode="wb")
NBALogo <- readJPEG("NBALogo.jpg", native=TRUE)
grb <- rasterGrob(NBALogo, interpolate=TRUE)
p4 <- p2 +
  annotation_custom(grb, xmin=0.025, xmax=0.05,
                  ymin=0.1, ymax=0.15) +
  guides(color=FALSE)
print(p4)

library(cowplot)
ggdraw() +
  draw_plot(p1) +
  draw_plot(p2, x=0.55, y=0.06, width=0.3, height=0.325)

plot_grid(p1, NULL, p2, p4, nrow=2,
         labels = c("A", "", "B", "C"))

ggdraw() +
  draw_plot(p3, x=0, y=0.5, width=1, height=0.5) +
  draw_plot(p2, x=0, y=0, width=0.5, height=0.5) +
  draw_plot(p4, x=0.5, y=0, width=0.5, height=0.5)

q1 <- ggplot_build(p1)
q1$data[[1]]$shape <- 17
q1$data[[1]]$size <- 3
p1b <- ggplot_gtable(q1)
plot(p1b)

str(q1$data[[1]])

#####
# 6.4 BUILDING INTERACTIVE GRAPHICS

```

```
#####  
library(plotly)  
Pbox.sel <- subset(Pbox, MIN>=500)  
attach(Pbox.sel)  
X <- data.frame(AST,TOV, PTSpm=PTS)/MIN  
detach(Pbox.sel)  
mypal <- colorRampPalette(c("blue","yellow","red"))  
p5 <- scatterplot(X, data.var=c("AST","TOV"),  
                  z.var="PTSpM", palette=mypal)  
ggplotly(p5, tooltip="text")  
  
data <- Pbox[1:64, c("PTS","P3M","P2M","OREB","Team")]  
p6 <- scatterplot(data, data.var=1:4, z.var="Team")  
ggplotly(p6)
```