

NBA

Risto Hinno

Friday, June 26, 2015

Jaanus "Sops" Kolmene käib jõusaalis kolm korda nädalas. Järgmisel kuul sisustab ta ka reede õhtupooliku raskusi tõstes, sest tema rinnalihas on statistiku jaoks liiga nõrk. Nimelt otsib korvpalliklubi Brooklyn Nets oma ridadesse statistikut, kes analüüsiks korvpalliandmeid ja aitaks seeläbi optimiseerida meeskonna mängistrateegiat. Kandidaadi nõuete seas on ka 50 kg rinnalt surumine. 1

Kuna rinnalihase nõue on ehk peagi täidetud, tahab Jaanus enne intervjuule minekut saada praktistikat kogemust NBA andmete analüüsimal. Ta teab, et NBA mängude kohta kogutakse SportVU tehnoloogia abil detailset statistikat.

Tutvumine andmestikuga

Kasutame NBA 2013/14 hooaja andmeid. Andmestikus on järgnevad tunnused:

- name - korvpalluri nimi
- team_name - korvpalluri meeskonna nimi
- opponent - vastasmeeskonna nimi
- game_date - mängu kuupäev
- shot_made_flag - kas vise läks korvi
- shot_type - kas oli 2-punkti või 3-punkti vise
- x - viske x-koordinaat
- y - viske y-koordinaat

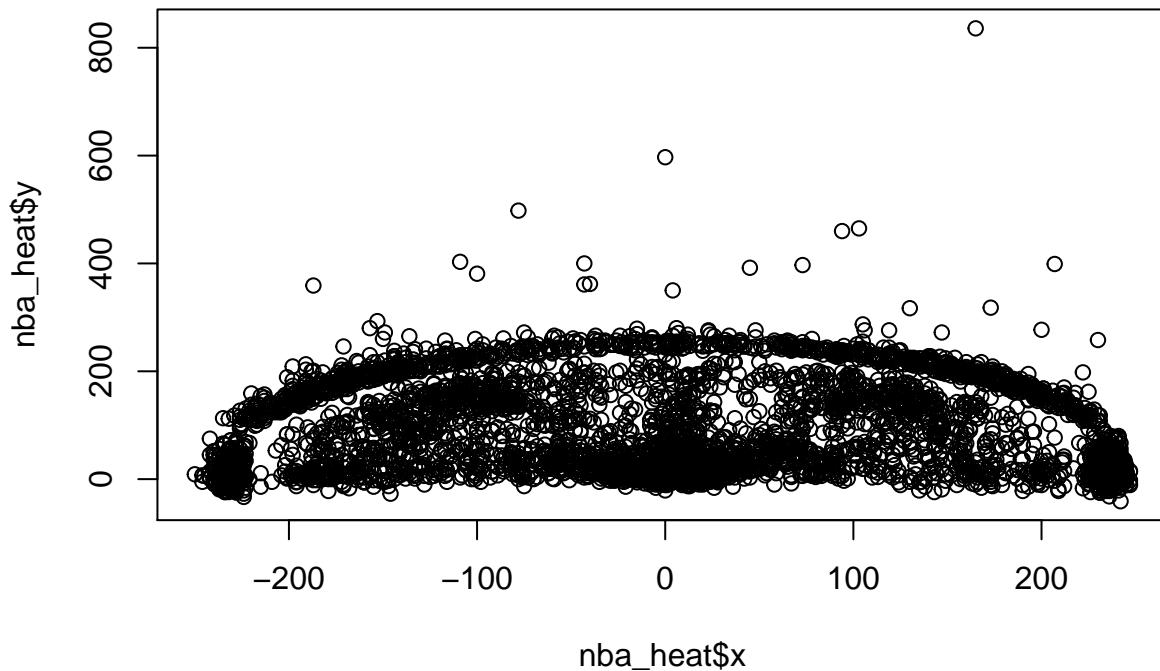
Laadi alla andmestik nba_2013.csv ja loe töökeskkonda.

```
nba=read.csv2("./data/nba_2013.csv")
```

Ülesanne 1 (1 punkt)

Visualiseeri NBA meeskonna Miami Heat viskekohti hajuvusdiagrammi abil kasutades tunnuseid x ja y. Iga sooritatud vise näita täpikesena. Kasuta R-i baasgraafika funktsiooni plot.

```
nba_heat=subset(nba, team_name=="Miami Heat")
plot(nba_heat$x, nba_heat$y)
```



Ülesanne 2 (2 punkti)

Diskretiseeri mänguväljak (jaga mänguväljak tükkideks).

Tükid moodustada nii, et ümarda x- ja y-koordinaat künnelisteni. Iga kastikese kohta arvuta:

- visete arv
- tabavusprotsent

Näpunäited:

- uuri, mida teeb käsk round(21:30, -1).
- kasuks tuleb paketi dplyr funktsionaalsus (mutate, group_by, summarise)

```
nba$x_round=round(nba$x, -1)
nba$y_round=round(nba$y, -1)

library(dplyr)
nba2=nba%>%
  group_by(x_round, y_round)%>%
  mutate(count = n(),
        tabavusprotsent=sum(shot_made_flag)/count)
```

Ülesanne 3 (2 punkti)

Visualiseeri eelmise ülesande tulemusena saadud andmeid. Tee kaks joonist:

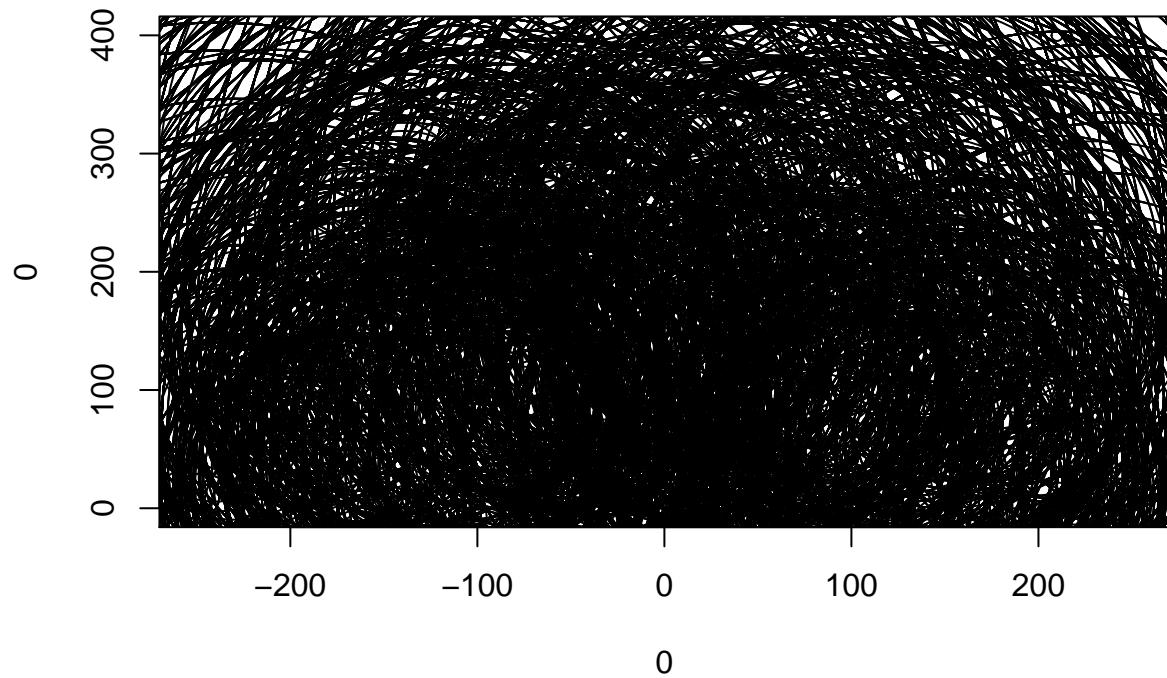
- Iga diskretiseeritud kastikese kohta joonista ring. Ringi pindala olgu proportsionaalne visete arvuga.
- Iga diskretiseeritud kastikese kohta joonista ruut. Ruudu pindala olgu proportsionaalne visete arvuga.

Näpunäited:

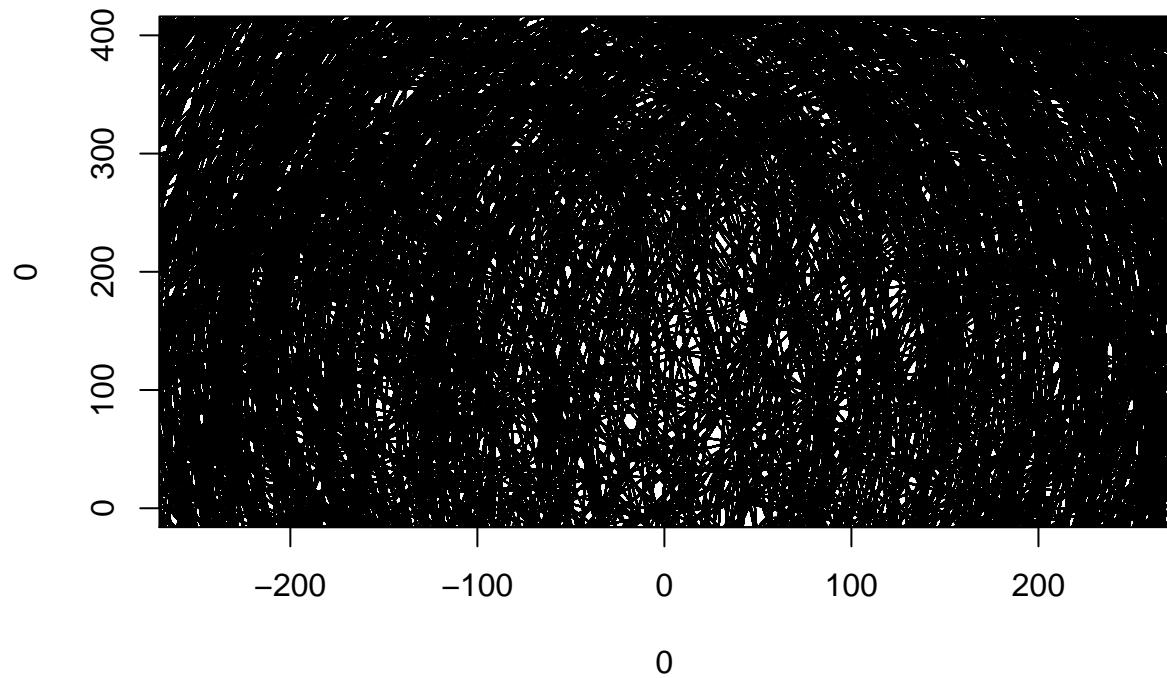
- Esmalt tekita tühi aken, kuhu hakkame joonistama: `plot(0, 0, type = "n", xlim=c(-250, 250), ylim=c(0, 400))`
- Ringikesed kanna joonisele funktsiooni `symbols` abil.
- Kuigi `symbols` töötab ka vektoritega, on edasise huvides lihtsam kasutada for tsüklit ning kanda ringid/ruudud joonisele ükshaaval.
- Funksiooni `symbols` korral on kasulikud argumendid `inches=FALSE`, `add=TRUE`. Esimene neist on vajalik selleks, et sümboli suurus skaleeruks vastavalt etteantud raadiusele, `add=TRUE` selleks, et viimati tehtud joonisele lisada uus sümbol.

```
nba2_heat=subset(nba2, team_name=="Miami Heat")

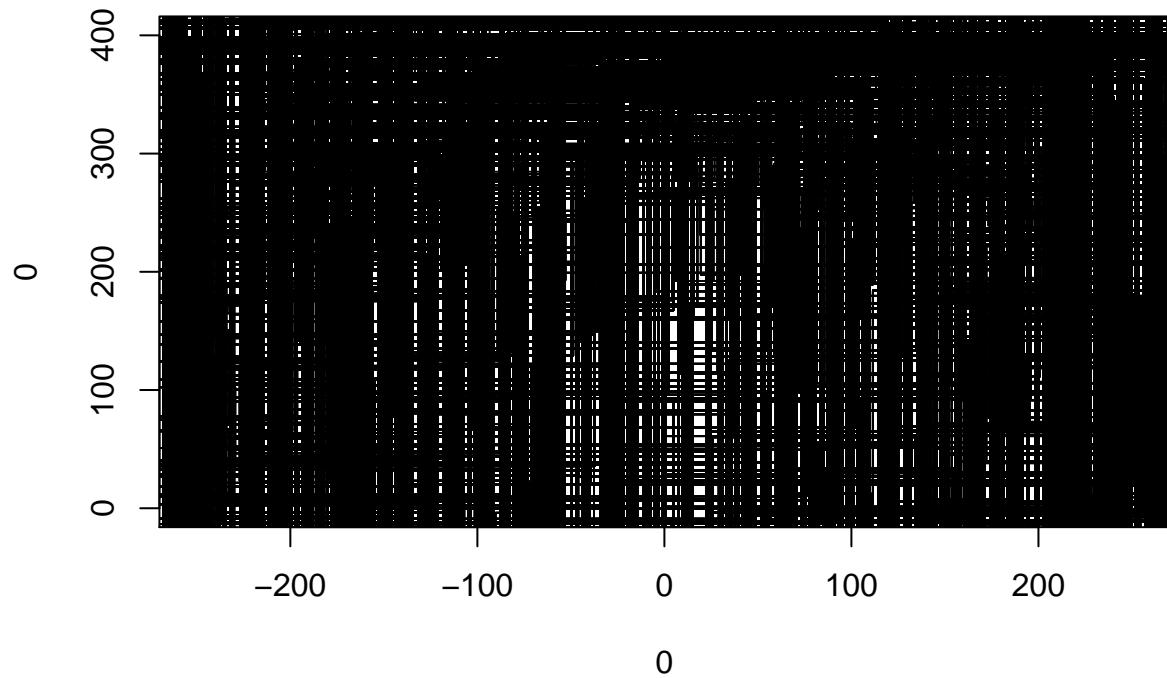
plot(0, 0, type = "n", xlim=c(-250, 250), ylim=c(0, 400))
#symbols(200, 100, circles=2, inches=FALSE, add=TRUE)
for (i in length(nba2_heat)) {
  symbols(nba2_heat$x_round, nba2_heat$y_round, circles=nba2_heat$count,
          inches=FALSE, add=TRUE)
}
```



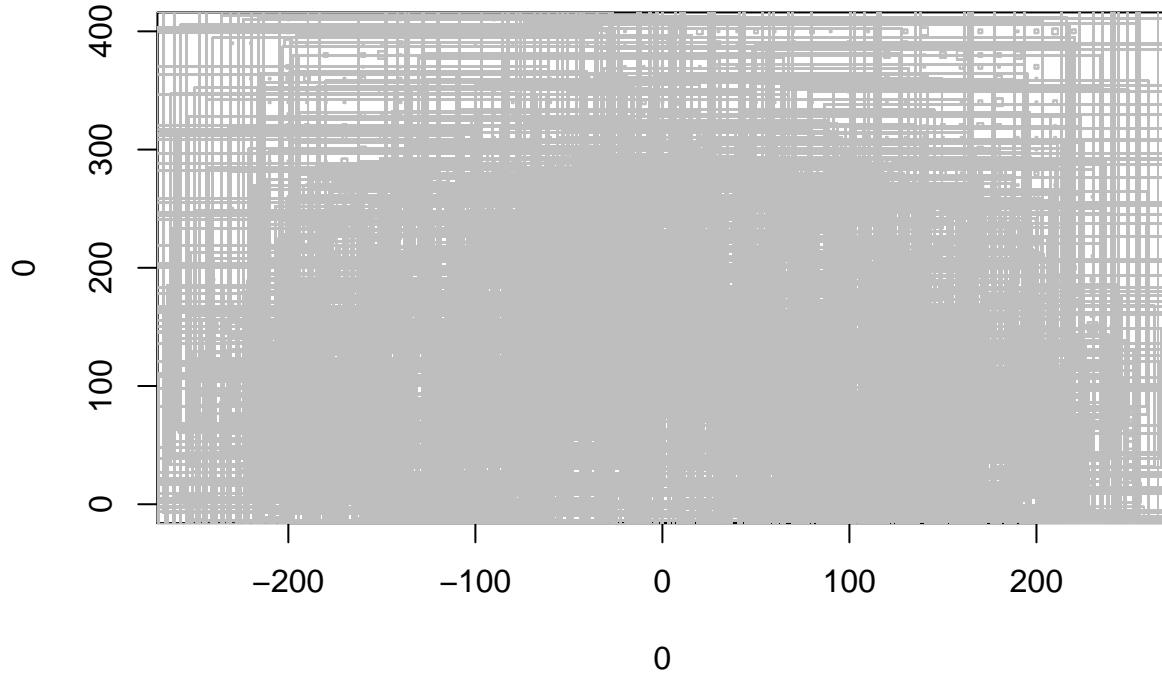
```
#kasutame points käsku
plot(0, 0, type = "n", xlim=c(-250, 250), ylim=c(0, 400))
for (i in length(nba2)) {
  points(nba2$x_round, nba2$y_round, cex=nba2$count)
}
```



```
#sama asi ruutudega
plot(0, 0, type = "n", xlim=c(-250, 250), ylim=c(0, 400))
for (i in length(nba2)) {
  points(nba2$x_round, nba2$y_round, cex=nba2$count, pch=22)
}
```



```
#kasutame symbols käsku
plot(0, 0, type = "n", xlim=c(-250, 250), ylim=c(0, 400))
for (i in length(nba2)) {
  symbols(nba2$x_round, nba2$y_round, squares=nba2$count,
          inches=F, add=TRUE, fg="grey")
}
```



Ülesanne 4 - suuruse muutmine (2 punkti)

Näeme, et meie joonisel on korvialuste kastikeste pindala palju suurem kui NY Timesi joonisel. NY Timesi graafikaosakond on jaganud visete arvu kolme kategooriasse: low, medium, high. Mida arvad, kas on hea mõte jagada kastid kolme kategooriasse?

- a.) Vali välja piirid, mille põhjal jaotatakse visete arv kolme kategooriasse (low, medium, high). Iga kategooria jaoks vali sobiv pindala suurus (kontrolli, et tulemus oleks visuaalselt kena). Kirjuta funktsioon get_radius, mis visete arvu sisestamisel tagastab vastava kategooria raadiuse.
- b.) Visualiseeri nüüd viskepositsioone sarnaselt eelmisele ülesandele, kui ringidel/ruutudel on vaid 3 erinevat suurust.

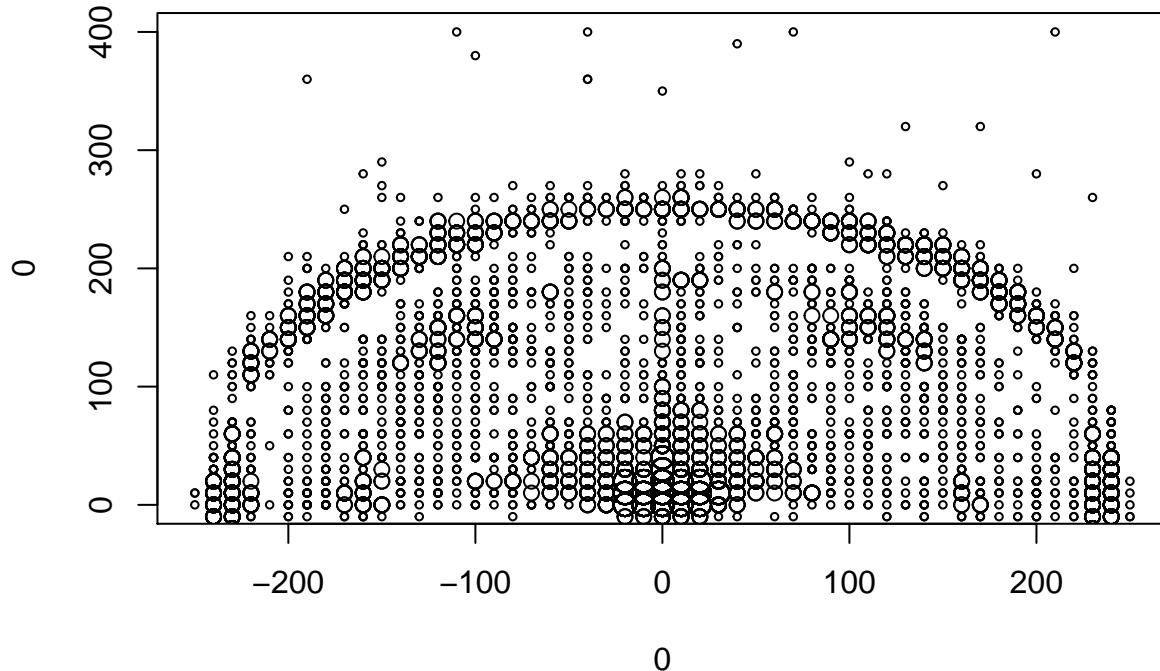
```
get_radius = function(visete_arv){
  tulem=list()
  for (i in 1:length(visete_arv)) {
    if (visete_arv[i]<150) {tulem[i]=2}
    else if (visete_arv[i]<=1000) {tulem[i]=4}
    else if (visete_arv[i]>1000) {tulem[i]=6}
  }
  tulem
}

nba2_heat$count_grupp=get_radius(nba2_heat$count)
#visualiseerime
plot(0, 0, type = "n", xlim=c(-250, 250), ylim=c(0, 400))
```

```

#symbols(200, 150,circles=10,inches=FALSE, add=TRUE)
for (i in length(nba2_heat)) {
  symbols(nba2_heat$x_round, nba2_heat$y_round,circles=nba2_heat$count_grupp,
          inches=FALSE, add=TRUE)
}

```



Ülesanne 5 - värvid (2 punkti)

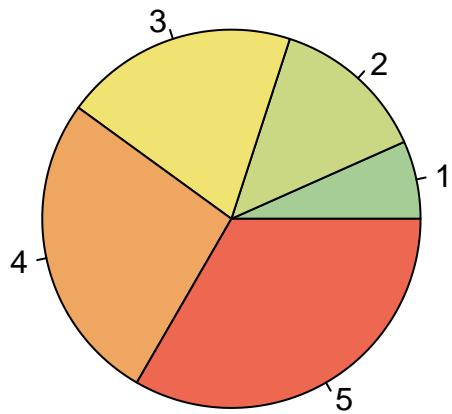
Uuri, mida tähistavad NY Times'i graafikul punane ja roheline värv.

Lisa nüüd ruutudele/ringidele värv vastavalt visketabavusele. Selleks kirjuta esmalt funktsioon get_color, mis tagastab antud visketabamusele vastava värvikoodi.

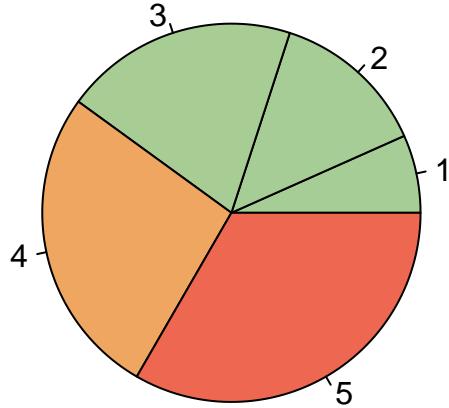
```

pal = colorRampPalette(c("#A6CD95", "#F1E471", "#ED6751"))
# colorRampPalette tagastab funktsiooni, mis võtab sisendiks täisarvu ja
# väljastab vastava arvu värvue
colors = pal(100)
# näide
pie(1:5, col=colors[c(1, 25, 50, 75, 100)])

```



```
pie(1:5, col=colors[c(1, 1.2, 2, 75, 100)])
```



```

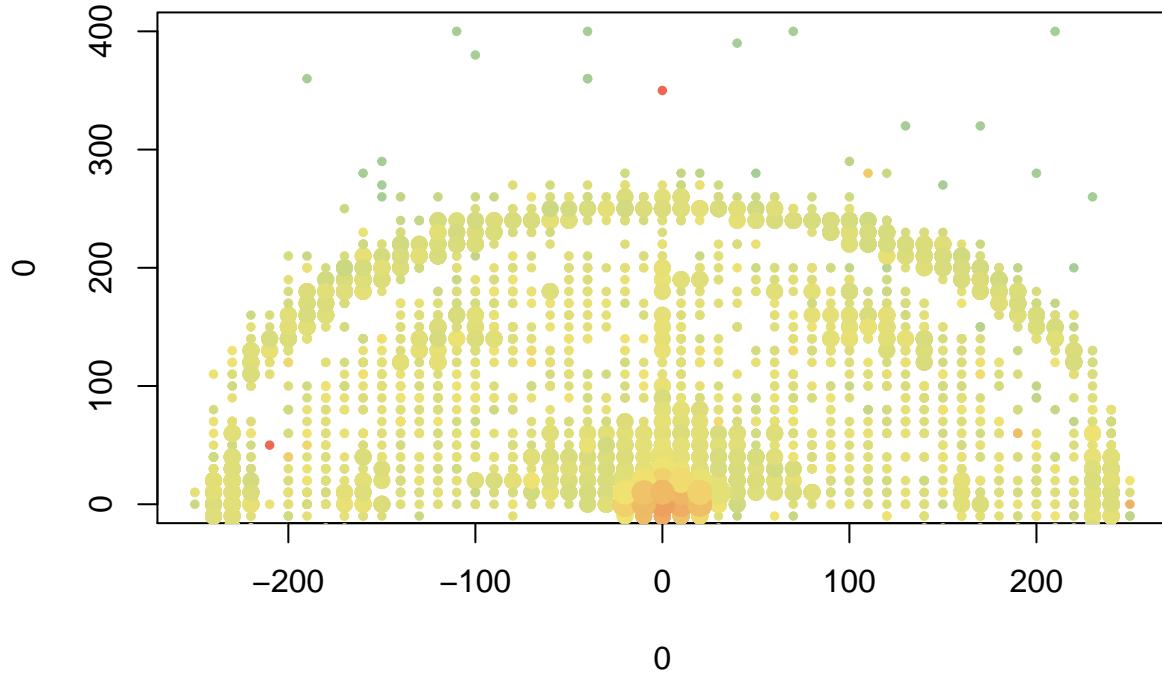
#teeme täisarvuks tabavusprotsendi
nba2_heat$tabavusprotsent2=round(nba2_heat$tabavusprotsent*100)
#asendame nullid 1-ga, muidu ei anna colorit
nba2_heat$tabavusprotsent3=ifelse(nba2_heat$tabavusprotsent2==0 ,1,
                                    nba2_heat$tabavusprotsent2)

#teeme funktsiooni värvide arvutamiseks
get_color = function(visketabamus){
  pal = colorRampPalette(c("#A6CD95", "#F1E471", "#ED6751"))
  colors = pal(100)
  varv=colors[visketabamus]
}

#arvutame väärised
nba2_heat$color=get_color(nba2_heat$tabavusprotsent3)

#paneme plotile
plot(0, 0, type = "n", xlim=c(-250, 250), ylim=c(0, 400))
for (i in length(nba2_heat)) {
  symbols(nba2_heat$x_round, nba2_heat$y_round,circles=nba2_heat$count_grupp,
          inches=FALSE, add=TRUE, bg=nba2_heat$color, fg=nba2_heat$color)
}

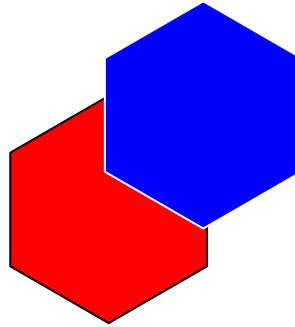
```



Ülesanne 6 - kuusnurgad (2 punkti)

Kasuta nüüd ringide ja ruutude asemel kuusnurki. Oleme ette andnud funktsiooni `plot_hexagon`, mis sisestades keskpunkti koordinaadid ja pindala, joonistab kuusnurga.

```
plot_hexagon = function(x, y, area, col="red", ...){
  r = sqrt(2*sqrt(3)/9*area)
  x_id = x + c(0, sqrt(3)/2*r, sqrt(3)/2*r, 0, -sqrt(3)/2*r, -sqrt(3)/2*r)
  y_id = y + c(r, r/2, -r/2, -r, -r/2, r/2)
  polygon(x_id, y_id, col=col, ...)
}
#näide
plot(0, 0, type = "n", axes = FALSE, xlim=c(-5, 5),
      ylim=c(-5, 5), xlab="", ylab "", asp=1)
plot_hexagon(1, 1, 15)
plot_hexagon(3, 3, 15, col="blue", border="white")
```



```

#teeme reaalsete andmetega
#enne teeme numericuks
nba2_heat$count_grupp=as.numeric(nba2_heat$count_grupp)
#teeme countgruppi suuremaks, et paistaks välja plotilt
get_radius_hexagon = function(visete_arv){
  tulem=list()
  for (i in 1:length(visete_arv)) {
    if (visete_arv[i]<150) {tulem[i]=500}
    else if(visete_arv[i]<=1000) {tulem[i]=600}
    else if (visete_arv[i]>1000) {tulem[i]=700}
  }
  tulem
}
nba2_heat$count_grupp_hex=get_radius_hexagon(nba2_heat$count)
nba2_heat$count_grupp_hex=as.numeric(nba2_heat$count_grupp_hex)

#plotime, kuid millegipäras t kuvab ainult osad kuusnurgad
plot(0, 0, type = "n", xlim=c(-250, 250), ylim=c(0, 400))
for (i in 1:length(nba2_heat)) {
  plot_hexagon(x=nba2_heat$x_round[i], y=nba2_heat$y_round[i],
                area=nba2_heat$count_grupp_hex[i],
                col=nba2_heat$color[i], border="white")
}

```

