1. 基于 NBA 篮球数据的可视化实例

1.1 包的安装和加载

- 安装 packages

```
#install.packages("vcd")
#install.packages("ggpubr")
#install.packages("fmsb")
#install.packages("scatterplot3d")
#install.packages("wordcloud")
```

- 加载 packages

```
library(vcd)
library(ggpubr)
library(ggplot2)
library(cowplot)
library(fmsb)
library(corrplot)
library(scatterplot3d)
library(RColorBrewer)
library(wordcloud)
```

1.2 示例数据

本文采用的示例数据是 2016-2017 赛季 NBA 的数据

- 球员数据

  球员数据主要包括了球员 ID，球员名字等数据，其中，avg_pts 到 avg_play_time 分别表示球员的场均得分、助攻、进攻篮板、防守篮板、抢断、盖帽、失误、命中球数、出手次数、三分命中数、上场时间；position 表示球员在场上位置，1.5 表示该球员可以打 1 号位或者 2 号位。简单的将进攻篮板和防守篮板相加，作为篮板数据(avg_reb)

```
players = read.table('data/nba_players.csv',header=T, sep=",")
players$avg_reb = players$avg_oreb+players$avg_dreb
dim(players)
players[1:5,]
```

446 16

player_id	player_name	avg_pts	avg_ast	avg_oreb	avg_dreb	avg_stl	avg_blk	avg_tov	avg_fgm	avg_fga	avg_tpm	avg_play_time	team_name	position
1	凯尔-科沃尔	10.12	1.64	0.13	2.66	0.49	0.31	1.03	3.57	7.66	2.42	26.13	骑士	2.5
2	蒂亚戈-斯普利特	4.33	0.44	0.89	1.56	0.11	0.11	0.67	1.56	3.44	0.22	8.56	76人	5.0
3	保罗-米尔萨普	17.87	3.63	1.59	6.13	1.30	0.87	2.29	6.17	13.94	1.09	33.66	老鹰	4.0
4	萨博-塞福洛沙	7.16	1.73	0.84	3.52	1.48	0.50	0.95	2.81	6.35	0.66	25.73	老鹰	2.5
5	杰夫-蒂格	15.29	7.78	0.39	3.66	1.24	0.39	2.63	4.90	11.07	1.10	32.44	步行者	1.0

- 库里每场比赛数据

  记录了库里每场比赛的详细数据

```
currey_performance = read.table('data/nba_curry_performance.csv',header=T, sep=",")
currey_performance_sum = aggregate(currey_performance[,c(2:11)],list(currey_performance[,1]),sum)
currey_performance_sum[1:5,]
```

Group.1	pts	ast	oreb	dreb	stl	blk	tov	fgm	fga	tpm
2016/10/26	26	4	0	3	0	0	4	9	18	3
2016/10/29	23	7	1	0	1	0	4	8	19	4
2016/10/31	28	3	0	1	0	0	1	9	17	5
2016/11/10	24	6	0	3	1	0	5	8	12	4
2016/11/11	33	7	1	4	1	0	4	10	17	7

- 球队地理信息

lon 和 lat 表示球队的经纬度，由于洛杉矶湖人与快船同在一个城市，因此坐标稍作了一些修改。

```
team_info =read.table('data/nba_team.csv',header=T, sep=",")
team_info[1:5,]
```

team_id	team_name	division	lon	lat
1	老鹰	东部	-84.39215	33.75689
2	凯尔特人	东部	-71.06221	42.36620
3	鹈鹕	西部	-90.07153	29.95107
4	公牛	东部	-87.67418	41.88069
5	黄蜂	东部	-80.83934	35.22514

由于不同数据可视化图形需要不同格式，类型的数据，因此在绘制过程中，会使用pandas对数据进行更多的处理，以便图形的绘制

- 勇士对每场球员数据

记录了勇士球员每场比赛的详细数据

```
gs_player_performance = read.table('data/gs_player_performance.csv',header=T, sep=",")
gs_player_performance1 = aggregate(gs_player_performance[,c(4:13)],list(gs_player_performance[,3]),sum)
gs_player_performance1[1:5,]
```

Group.1	pts	ast	oreb	dreb	stl	blk	tov	fgm	fga	tpm
2015/10/28	111	29	21	35	8	7	20	41	96	9
2015/10/31	112	26	11	36	9	4	8	43	93	9
2015/11/1	134	26	3	35	10	4	10	46	84	17
2015/11/10	109	29	13	28	13	4	16	43	91	12
2015/11/12	100	22	7	29	12	8	23	36	75	11

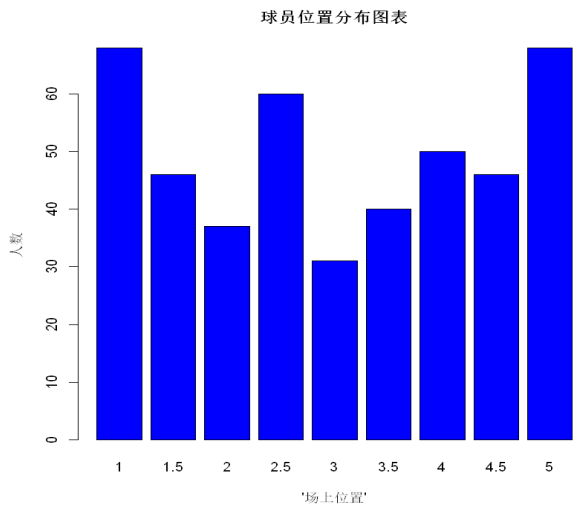
1.3 探索性数据分析

条形图

- 各个位置上球员数量分布

绘制基础的条形图

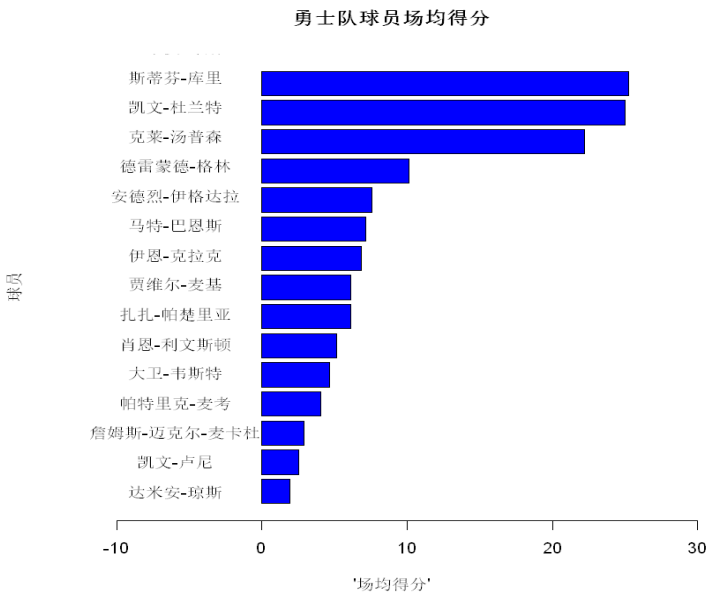
```
# 计算各个位置上球员数量
player_position_num = as.data.frame(table(players$position))
names(player_position_num)=c("position","count")
player_position_num
#绘制条形图
barplot(player_position_num$count,names.arg = player_position_num$position,xlab = "'场上位置'",ylab = "人数",col = "blue", main = "球员位置分布图表")
```



- 勇士队球员场均得分

绘制横向的条形图

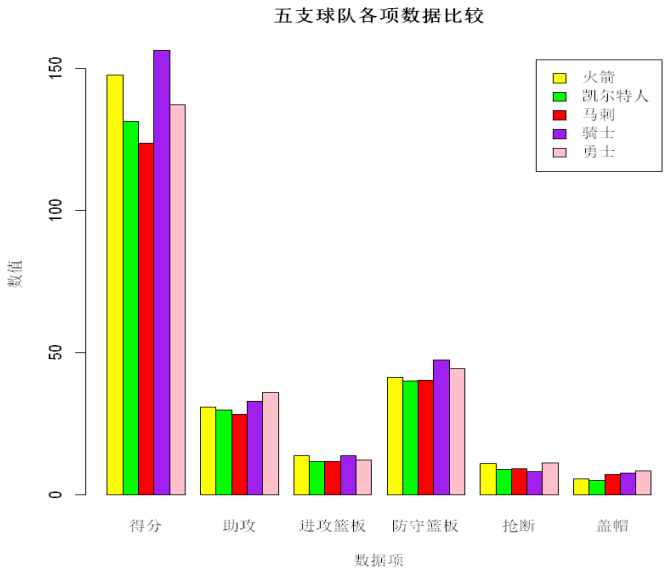
```
# 取出勇士球员数据
gs_players = players[which(players$team_name == "勇士"),]
# 按得分排序
gs_players=gs_players[order(gs_players$avg_pts,decreasing= F),]
#绘制横向条形图
barplot(gs_players$avg_pts,,xlim=c(-12,30),horiz=T,xlab = "'场均得分'",ylab = "球员",col = "blue", main = "勇士队球员场均得分")
text(seq(from=0.7,length.out=135,by=1.22),x=-6,label=gs_players$player_name)
```



- 五支球队各项数据比较

分组条形图是可视化数据对比的常用方法

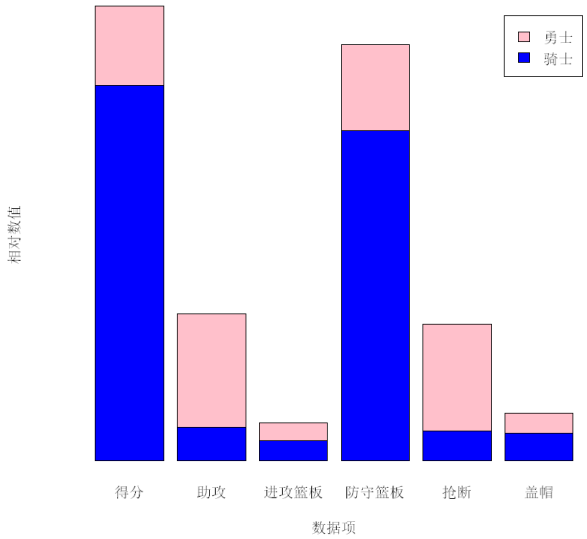
```
# 取出五支队伍球员的数据
sub_team_players = players[which((players$team_name == "勇士") | (players$team_name == "
火箭") | (players$team_name == "骑士") | (players$team_name == "凯尔特人") |
(players$team_name == "马刺")),]
# 计算五支队伍的人均数据
sub_team_players_stats =
aggregate(sub_team_players[,c(3:13)],list(sub_team_players[,14]),sum)
sub_team_players_stats
sub_team_players_stats1=sub_team_players_stats[,1:7]
sub_team_players_stats2=as.matrix(sub_team_players_stats1[,2:7])
barplot(sub_team_players_stats2,
      beside = TRUE,
      main = "五支球队各项数据比较",
      names.arg = c("得分","助攻","进攻篮板","防守篮板","抢断","盖帽"),
      xlab = "数据项",
      ylab = "数值",
      col=c("yellow","green","red","purple","pink"),
      legend=sub_team_players_stats1[,1]
      )
```



- 两支队伍数据对比

堆叠条形图可以增强对比度

```
gs_cle_stats = sub_team_players_stats1[which((sub_team_players_stats1[,1] == "勇士")
|(sub_team_players_stats1[,1] == "骑士")),]
gs_cle_stats
gs_cle_stats1=matrix(c(gs_cle_stats[1,2:7],gs_cle_stats[2,2:7]),2,6)
barplot(gs_cle_stats1,
        axes=FALSE,
        names.arg = c("得分","助攻","进攻篮板","防守篮板","抢断","盖帽"),
        xlab="数据项",
        ylab="相对数值",
        col=c("blue","pink"),
        legend=gs_cle_stats[,1])
```

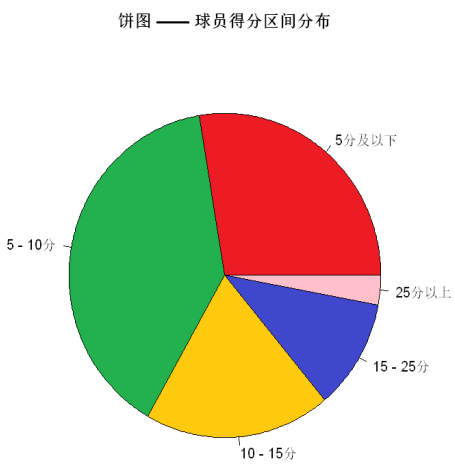


饼图

- 球员得分区间分布

```
#将球员得分进行封箱操作
names = c('5 分及以下', '5 - 10 分', '10 - 15 分', '15 - 25 分', '25 分以上')
break1=c(0, 5, 10, 15, 25, 100)
ta=table(cut(players$avg_pts,break1,labels=names,ordered_result = T))
data=as.data.frame(ta)
data
#涂色（可选）
cols = c("#ED1C24","#22B14C","#FFC90E","#3f48CC","PINK")
pie(data$Freq,labels = data$Var1,edges=500,col=cols,main="饼图 —— 球员得分区间分布")
```

Var1	Freq
5 分及以下	123
5 - 10 分	174
10 - 15 分	86
15 - 25 分	49
25 分以上	13

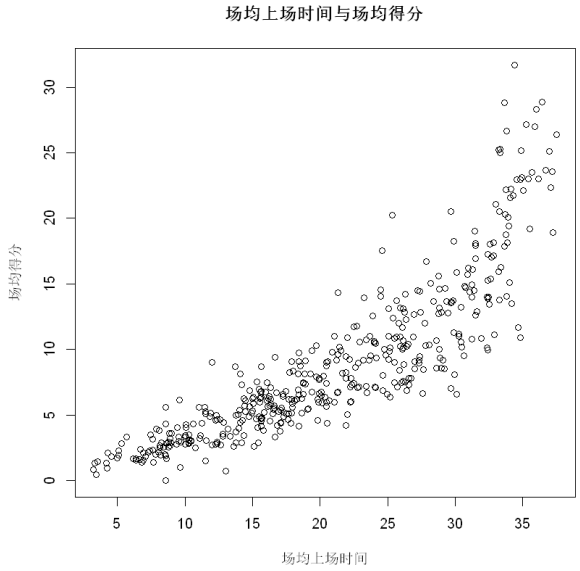


散点图

- 场均上场时间与场均得分

绘制基础散点图

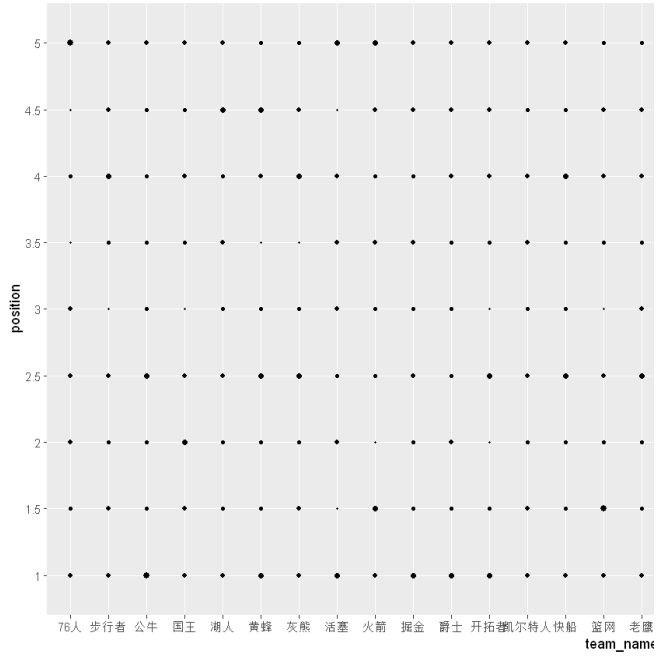
```
plot(players$avg_play_time, players$avg_pts, main="场均上场时间与场均得分", xlab="场均上场时间", ylab="场均得分", pch=1)
```



气泡图

- 各球队在每个位置上的球员数量(以大小表示)

```
players_tp_num=as.data.frame(table(players$team_name,players$position))
names(players_tp_num)=c("team_name","position","player_num")
g<-ggplot(players_tp_num, aes(x=team_name,y=position,size=player_num))  +
geom_point(colour = "black") + scale_size(rang=c(0.5,2))
ggdraw() + draw_plot(g, width = 2, height = 1)
```

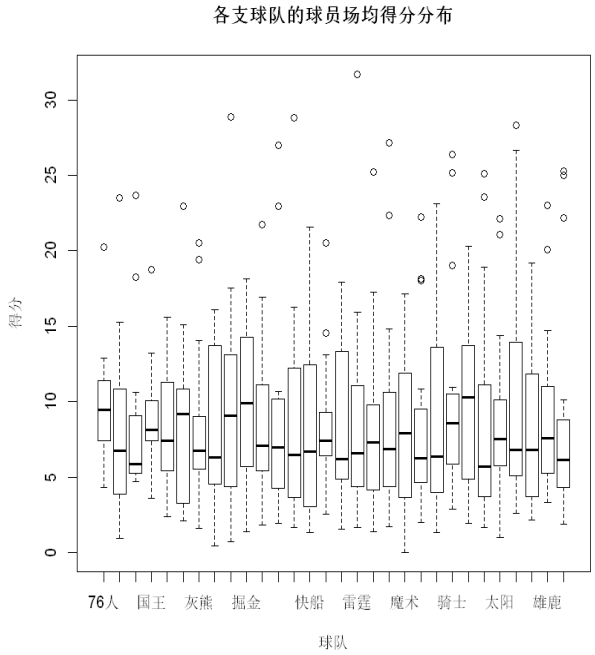


箱形图

- 各支球队的球员场均得分分布

  通过箱形图方便地观察球员得分分布情况，特别是数据中的几个关键统计量，如中位数等

```
boxplot(avg_pts ~ team_name, data = players, xlab = "球队",ylab = "得分", main = "各支球队的球员场均得分分布")
```



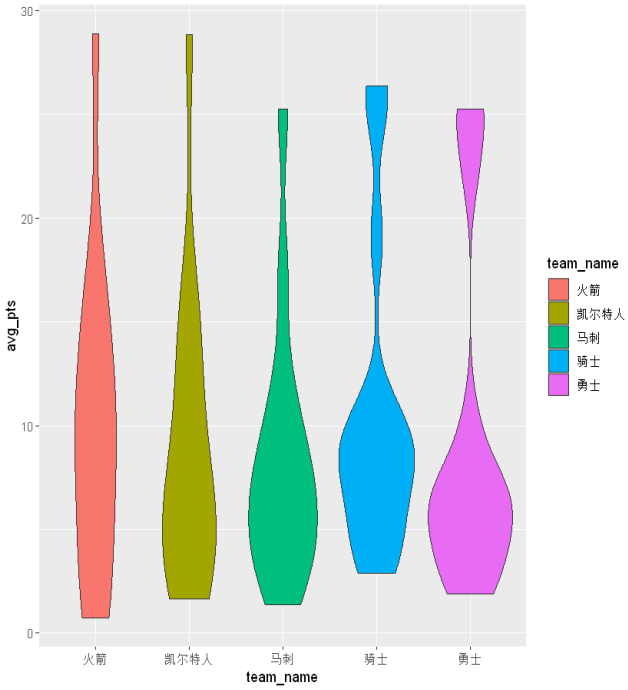
小提琴图

- 五支球队的球员场均得分分布

  小提琴图与箱形图有相似，但是能更清晰的看到整个数据的分布情况

```
# 取出五支队伍球员的数据
sub_team_players = players[which((players$team_name == "勇士") | (players$team_name == "
火箭") | (players$team_name == "骑士") | (players$team_name == "凯尔特人") |
(players$team_name == "马刺")),]
# 计算五支队伍的人均数据
sub_team_players[1:5,]
p<-ggplot(sub_team_players, aes(x = team_name, y = avg_pts))
p+geom_violin(aes(fill = team_name))
```

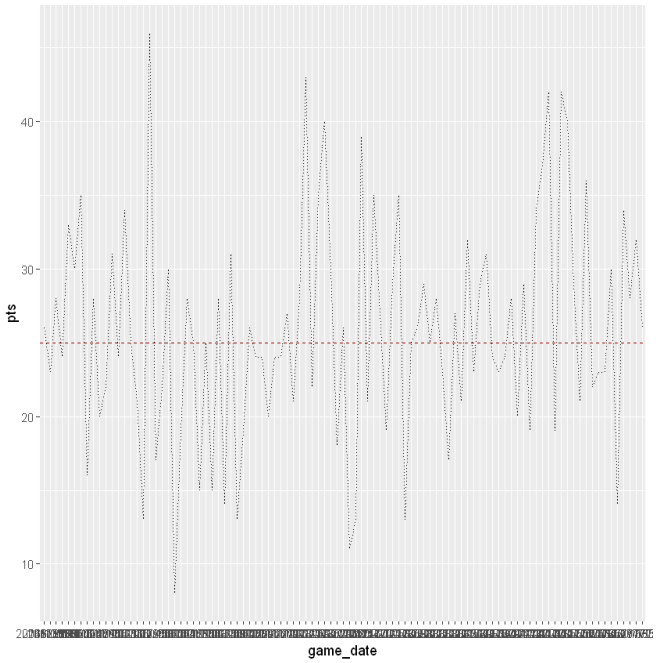
	player_id	player_name	avg_pts	avg_ast	avg_oreb	avg_dreb	avg_stl	avg_blk	avg_tov	avg_fgm	avg_fga	avg_tpm	avg_play_time	tean
1	1	凯尔-科沃尔	10.12	1.64	0.13	2.66	0.49	0.31	1.03	3.57	7.66	2.42	26.13	
9	10	埃迪-塔瓦雷斯	4.00	0.50	2.00	3.50	0.00	3.00	1.00	2.00	2.50	0.00	14.00	
11	14	阿米尔-约翰逊	6.50	1.76	1.45	3.09	0.64	0.79	0.98	2.66	4.61	0.34	20.11	
12	15	约纳斯-杰雷布科	3.83	0.92	0.77	2.71	0.35	0.23	0.50	1.46	3.35	0.58	15.83	
13	16	埃弗里-布拉德利	16.25	2.18	1.18	4.91	1.25	0.20	1.62	6.53	14.11	1.96	33.40	



折线图

- 库里比赛得分趋势

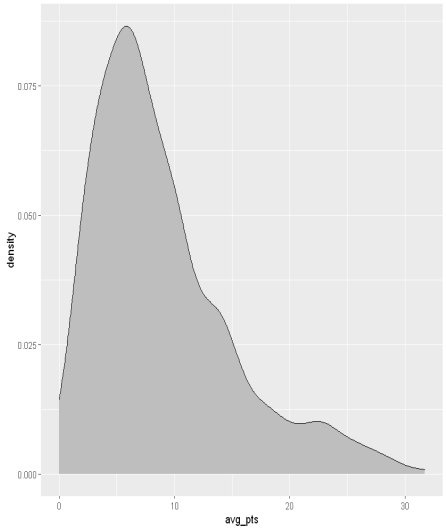
```
currey_performance1 <- currey_performance # 赋值数据
currey_performance1$game_date <- factor(currey_performance1$game_date)#变量因子化
g<-ggplot(currey_performance1, aes(x=game_date, y=pts, group=1)) +
geom_line(linetype="dotted")+geom_hline(aes(yintercept=25),colour="#990000",linetype="dashed")
# ggdraw() + draw_plot(g, width = 10, height = 1)
g
```



密度图

- 得分密度图

```
p<-ggplot(players, aes(x = avg_pts))
p + geom_density(color = "black", fill = "gray")
```

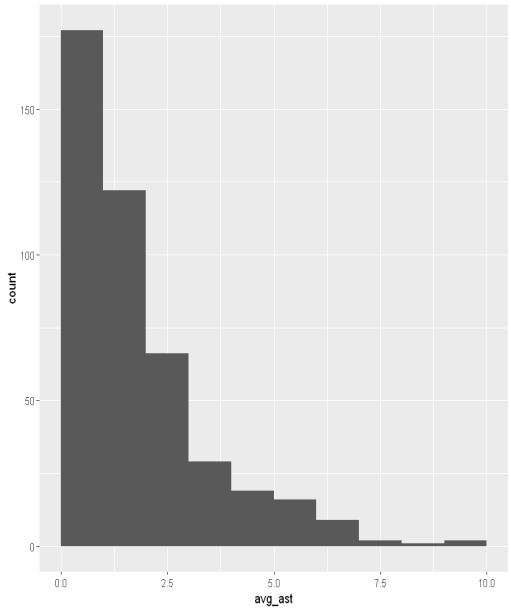


直方图

- 助攻直方图

```
p<-ggplot(players, aes(x=avg_ast)) + geom_histogram(breaks=seq(0,10,1))+ xlim(0,10)
p
```

```
Warning message:
"Removed 3 rows containing non-finite values (stat_bin)."
```

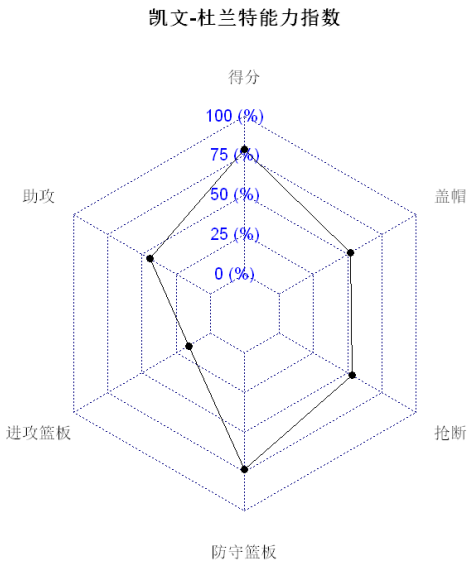


雷达图

- 凯文-杜兰特能力指数

```
players_f=players[,2:8]
dat1 <- data.frame(
  avg_pts = c(max(players_f$avg_pts), min(players_f$avg_pts)),
  avg_ast = c(max(players_f$avg_ast), min(players_f$avg_ast)),
  avg_oreb = c(max(players_f$avg_oreb), min(players_f$avg_oreb)),
  avg_dreb = c(max(players_f$avg_dreb), min(players_f$avg_dreb)),
  avg_stl = c(max(players_f$avg_stl), min(players_f$avg_stl)),
  avg_blk = c(max(players_f$avg_blk), min(players_f$avg_blk)))
```

```
#players_f=players_f[which((players_f$player_name=="凯文-杜兰特") |
(players_f$player_name=="斯蒂芬-库里") |(players_f$player_name=="勒布朗-詹姆斯")
|(players_f$player_name=="安东尼-戴维斯") |(players_f$player_name=="卡尔-安东尼-唐斯
")),]
players_f=players_f[which(players_f$player_name=="凯文-杜兰特"),]
labels = c('得分', '助攻', '进攻篮板', '防守篮板', '抢断', '盖帽')
data1=players_f[,2:7]
dat <- rbind(dat1, data1)
names(dat)=labels
radarchart(dat,axistype =1,title="凯文-杜兰特能力指数")
```

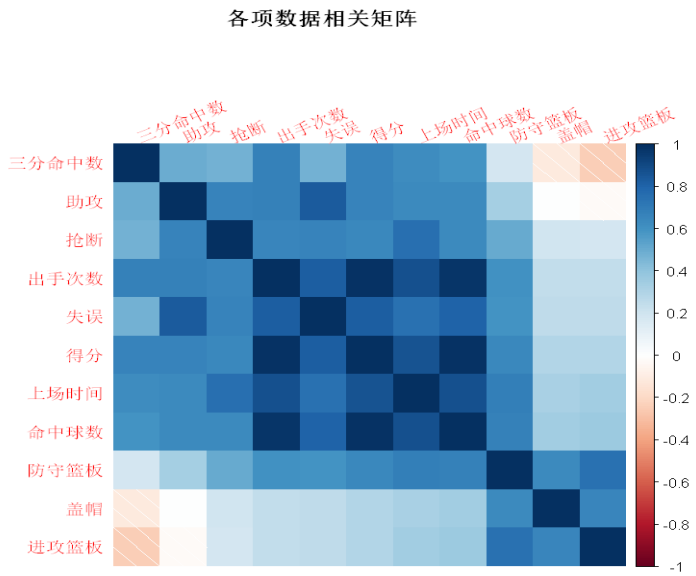


热力图

- 各项数据相关矩阵

```
item_data = players[, 3:13]
item_data_col= c('得分', '助攻', '进攻篮板', '防守篮板', '抢断', '盖帽', '失误', '命中球数', '出手
次数', '三分命中数', '上场时间')
names(item_data)=item_data_col
```

```
co = cor(item_data)
corrplot(co,method="shade",title="各项数据相关矩阵",tl.srt=25,order="AOE",mar = c(1,1,1,1))
```

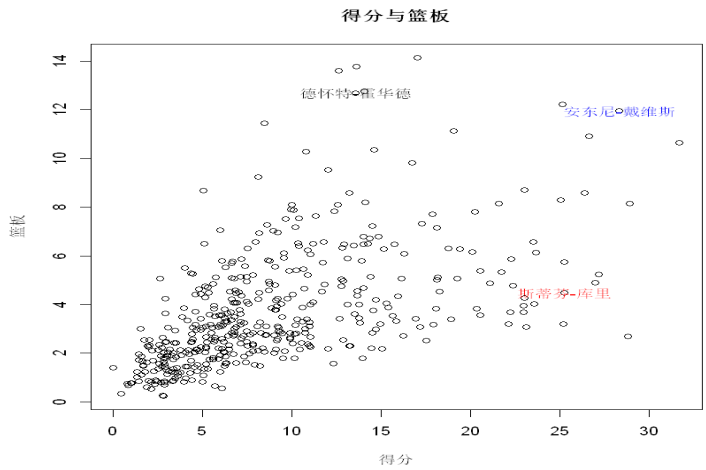


给图形增加注解

- 得分与篮板

给斯蒂芬-库里、德怀特-霍华德和安东尼-戴维斯三位球员增加注解

```
plot(players$avg_pts, players$avg_reb, main="得分与篮板", xlab="得分", ylab="篮板", pch=1)
players_node=players[which((players$player_name=="斯蒂芬-库里") |
(players$player_name=="德怀特-霍华德") |(players$player_name=="安东尼-戴维斯")),]
text(players_node[1,]$avg_pts,players_node[1,]$avg_reb,players_node[1,]$player_name,col="blue")
text(players_node[2,]$avg_pts,players_node[2,]$avg_reb,players_node[2,]$player_name,col="red")
text(players_node[3,]$avg_pts,players_node[3,]$avg_reb,players_node[3,]$player_name,col="black")
```

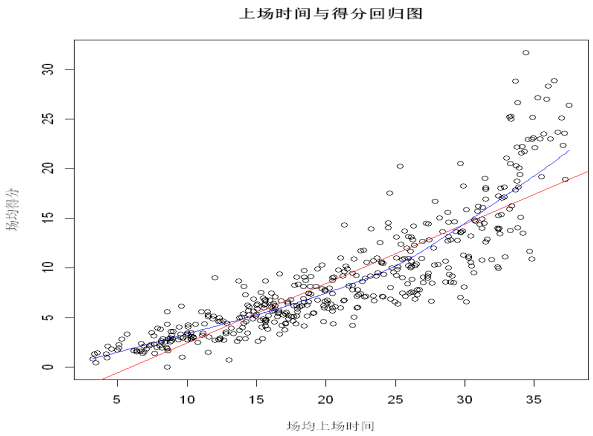


散点回归图

- 上场时间与得分回归图

画出上场时间与得分的散点图，添加回归线

```
plot(players$avg_play_time, players$avg_pts, main="上场时间与得分回归图", xlab="场均上场时间", ylab="场均得分", pch=1)
abline(lm(players$avg_pts~players$avg_play_time), col="red") # 添加拟合线，这里 lm() 是一个线性回归函数
lines(lowess(players$avg_pts~players$avg_play_time), col="blue") # LOWESS（局部加权散点图平滑化）
```

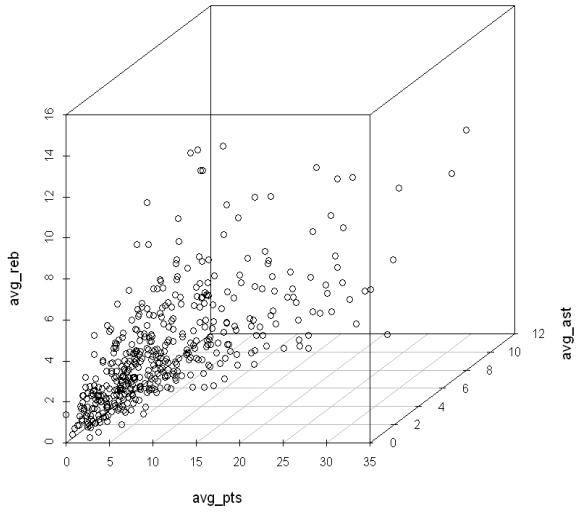


3D 图形

由于 3D 图形由于观察角度等原因很容易引起对图片的误解，因此数据可视化中 3D 图形并不是很常见，例如 ggplot 中并不包含 3D 绘制

- 得分、助攻与篮板

```
#scatterplot3d
attach(players)
p<-scatterplot3d(avg_pts,avg_ast,avg_reb)
```



词云图

词云图可以通过"关键字"大小、色彩等图片元素对数据进行有效的可视化

- 得分最高的 50 名球员

```
# 得分最高的 50 名球员
# 按得分排序
players_order=players[order(players$avg_pts,decreasing= T),]
players_order50=players_order[1:50,]
```

```
wordcloud(players_order50$player_name,round(players_order50$avg_pts),col = brewer.pal(8,
"Set1"),
        random.order=F,random.color=F,scale = c(2, 0.1),
       rot.per=F)
```