post02-anant-kota

Anant Kota

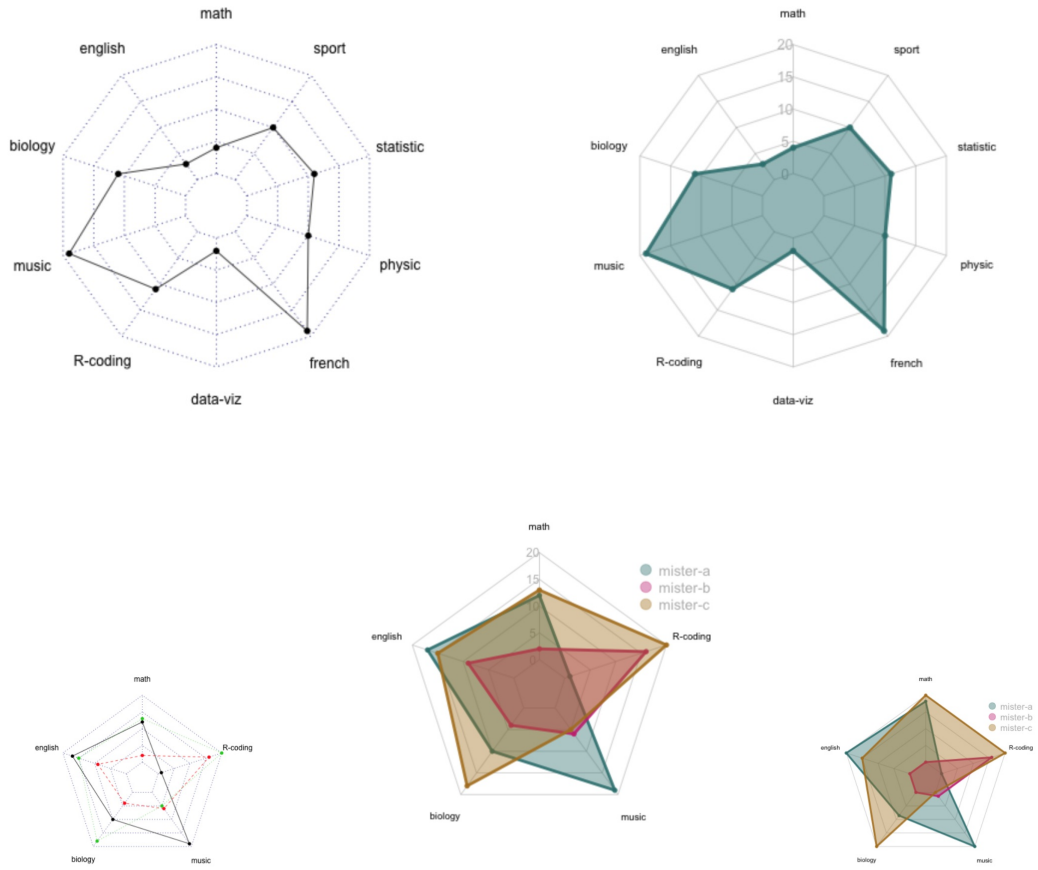
12/3/2017

Radar Plots: Constructing Sports Profiles

Motivation

In the age of social media, mass data analytics, and enhanced communication, we are a world of profiles. Constructing profiles is an incredibly useful skill for those wishing to enter the field of Human Resources to asses workplace performance, members of advertising agencies that are now targeting their pop-ads towards the web behavior of internet users (much to our chagrin), and even now the blossoming field of data nalytics in sports. Baseball, the heavy hitter of sports analytics due to its low injury rate and slow pace, has been the most active user of data analytics. In fact, data analytics has become so sophisticated in the baseball world that profiles of players are constructed in such a manner that these profiles are used to make personnel decisions such as whether to bench a player or not. In addition, profiles are even made for certain *situations* (depending on number of players on base, pitcher at the mound, etc.). Combining a player profile with a situational profile is a lethal combination in giving teams a winning edge in the sport. Fortunately, radar plots are an incredible way of visualizing player, team, or situations that R can accomplish and that's what we are going to be learning today.

Here are some examples of radar plots:



Radar plots are useful, not only for depicting profiles, but also for comparative analysis. Being a brilliant data scientist is insufficient in today's world. One can be have the most brilliant regression model but if they can clearly and efficiently convey their findings to those not well versed in statistical concepts. In our setting of being part of a data analytics team in a sports franchise, one would need to quickly compare different players and convey differences to a General Manager or a Coach. Radar plots achieve *clear* and *efficient* communication of finding by not leaving the reader with cumbersome numbers and mentally track differences as they make their way down a basic statstics chart. For baseball, in which coaches and general managers need to make momentary decision quickly before sending a player onto the field, it could be problematics to take the time to read every player profile in time.

Preliminary package and data loading

First we need to download the appropriate packages and data from www.pro-football-reference.com. We will be working, first, with quarterback profiles. You can get the data from here. Due to recent changes in the website, you will need to follow these instructions to export the data. In this post, we will be naming the data "qb-stats.csv"

```
# Load dplyr for manipulating provided data
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
```

```
# Load plotly for the function of actually creating radar plots
library(fmsb)
```

```
## Warning: package 'fmsb' was built under R version 3.3.2
```

```
# Read in quarterback stats into data frame
qbs <- read.csv('post02-data/qb-stats.csv', stringsAsFactors = F)
```

Data Cleaning for radarchart

For our purposes, we will only be looking at a select number of the stats provided. Namely, completion percentage, yards, touchdowns, interceptions, qbr, yards per attempt, yards per game, and sacks.

```
# Select required statistics
qbs <- qbs %>% select(X, Cmp., Yds, TD, Int, Y.A, Y.G, QBR, Sk)

# Change to more intuitive names
colnames(qbs)[c(1,9)] <- c('Player','Sacks')
```

Because we don't want to deal with the asterisks and other symbols we will remove them as well

```
# remove erronious symbols
for (i in 1:nrow(qbs)) {
  # grepl tests whether a character is in a string
  if (grepl('+',qbs[i,'Player'], fixed = T)) {
    qbs[i, 'Player'] <- substring(qbs[i, 'Player'], 1, nchar(qbs[i, 'Player'])-1)
  }
  if (grepl('*',qbs[i,'Player'], fixed = T)) {
    qbs[i, 'Player'] <- substring(qbs[i, 'Player'], 1, nchar(qbs[i, 'Player'])-1)
  }
}

# Replace NA with 0's
qbs[is.na(qbs)] <- 0

# Inspect data
str(qbs)
```

```
## 'data.frame':    96 obs. of  9 variables:
##  $ Player: chr  "Drew Brees" "Joe Flacco" "Blake Bortles" "Aaron Rodgers" ...
##  $ Cmp.  : num  70 64.9 58.9 65.7 62.4 67 63 61 65.3 60.4 ...
##  $ Yds   : int  5208 4317 3905 4428 3782 4917 4027 4233 4327 4386 ...
##  $ TD    : int  37 20 23 40 16 25 26 26 24 33 ...
##  $ Int   : int  15 15 16 7 14 12 16 14 10 21 ...
##  $ Y.A   : num  7.7 6.4 6.2 7.3 6.2 8.1 6.7 7.1 7.3 7.6 ...
##  $ Y.G   : num  326 270 244 277 236 ...
##  $ QBR   : num  72 58.4 49.2 76.9 52.8 71.7 51.8 58.9 70.5 64.5 ...
##  $ Sacks : int  27 33 34 35 33 23 21 40 37 36 ...
```

```
head(qbs)
```

```
##          Player Cmp.  Yds TD Int Y.A   Y.G  QBR Sacks
## 1    Drew Brees 70.0 5208 37  15 7.7 325.5 72.0    27
## 2    Joe Flacco 64.9 4317 20  15 6.4 269.8 58.4    33
## 3 Blake Bortles 58.9 3905 23  16 6.2 244.1 49.2    34
## 4 Aaron Rodgers 65.7 4428 40   7 7.3 276.8 76.9    35
## 5  Carson Wentz 62.4 3782 16  14 6.2 236.4 52.8    33
## 6  Kirk Cousins 67.0 4917 25  12 8.1 307.3 71.7    23
```

```
summary(qbs)
```

```
##     Player               Cmp.             Yds               TD        
##  Length:96          Min.   :  0.00   Min.   :   0.00   Min.   : 0.000  
##  Class :character   1st Qu.: 52.17   1st Qu.:  16.75   1st Qu.: 0.000  
##  Mode  :character   Median : 60.60   Median : 263.00   Median : 1.000  
##                     Mean   : 55.10   Mean   :1363.17   Mean   : 8.188  
##                     3rd Qu.: 67.00   3rd Qu.:3117.50   3rd Qu.:16.250  
##                     Max.   :100.00   Max.   :5208.00   Max.   :40.000  
##       Int              Y.A              Y.G               QBR        
##  Min.   : 0.000   Min.   : 0.000   Min.   :  0.000   Min.   :  0.00  
##  1st Qu.: 0.000   1st Qu.: 4.375   1st Qu.:  2.125   1st Qu.:  5.15  
##  Median : 1.000   Median : 6.800   Median : 87.150   Median : 44.90  
##  Mean   : 4.323   Mean   : 7.291   Mean   :118.133   Mean   : 41.14  
##  3rd Qu.: 7.000   3rd Qu.: 7.625   3rd Qu.:230.175   3rd Qu.: 64.75  
##  Max.   :21.000   Max.   :50.000   Max.   :325.500   Max.   :100.00  
##      Sacks      
##  Min.   : 0.00  
##  1st Qu.: 0.00  
##  Median : 3.00  
##  Mean   :11.64  
##  3rd Qu.:23.00  
##  Max.   :42.00
```

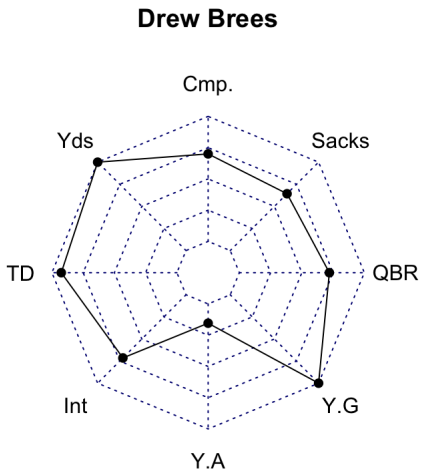
The function that we will be using to display the radar chart is (surprisngly) 'radarchart()'. However, in order to use this particular function, one will need to do some minor data cleaning before beginning. First off, because the radar chart functions is seeking to make comparative displays in information instead of absolute, it needs to know how to scale each variables in qbs. Otherwise, it would have one line with length 1000 for, say, yards while another line is 15 for touchdowns. Therefore, radarchart expects that each variable in qbs should have two additional rows above with the maximum of that column and the minimum of that column, for each column. In addition, radarchart works much better with an atomic structure of numbers only. So lets also name the rows after the players.

```
# name rows after players
rownames(qbs) <- qbs[,'Player']

# remove player column
qbs <- qbs[,-1]

# ========================================
# Add minimum and maximum of each variable
# ========================================
# Initialize a data frame
anex <- data.frame(c(1,1))

# For each column in qbs add a 2x1 vector with the min/max of that row
for (i in 1:ncol(qbs)) {
  add <- c(max(qbs[,i]), min(qbs[,i]))
  anex <- cbind(anex, add)
}
#Subtract the intialized part of the data frame
anex <- anex[,-1]

# Ensure thenames match for qbs and the part to annexed
colnames(anex) <- colnames(qbs)

# Add the anex data frame to the existing qbs
qbs <- rbind(anex,qbs)
```

Radar charts: first base

Because we are trying to construct a single profile for a single player, lets look at one qb, anmely, Drew Brees. Remeber we must always include the top two rows so that radarchart knows how to scale aech variable.

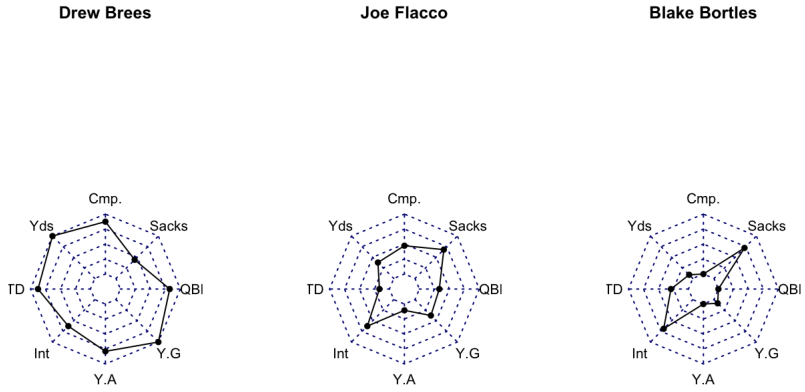
```
brees <- qbs[c(1,2,'Drew Brees'),]
radarchart(brees, title = 'Drew Brees')
```



This is the most basic form of the radarchart as it serves as a profile of Drew Brees. One interesting aspect of radarchart and the preliminary data cleaning that we did before hand for it is that each axis is already scaled towards the maximum and minimum of the respective variable in our data frame. For example, looking at our data frame for Drew Brees, because the point for Yards is on the perimeter of our polygon, we know that he is the leader ofr that respective stat. This makes sense because our original data frame was ranked by passing yards and Drew Brees was first on that list. However, by inspecting the Y.A (Yards per attempt) point, we know he is towards the bottom of yards per pass attempt. Therefore, we know all those yards are coming from more *attempts* rather than actual greater yards per attempt. Already, we see the power of using radar charts for quickly analyzing player profiles.

However, we also want to be able to analyze players side by side in a clear manner. We also, in a realistic scenario, want to analyze the relevant players and not count backups (sorry backups quarterbacks). One route we could take is to simply place multiple quarterback plots side by side and take the Top 15 qbs in passing yards. Since we're subsetting, we'll also have to rescale.

```
# Subset top 10 qbs including excluding first two min/max rows
top <- qbs[3:17,]

# Initialize a data frame
anex <- data.frame(c(1,1))

# For each column in top add a 2x1 vector with the min/max of that row
for (i in 1:ncol(top)) {
  add <- c(max(top[,i]), min(top[,i]))
  anex <- cbind(anex, add)
}
#Subtract the intialized part of the data frame
anex <- anex[,-1]

# Ensure thenames match for top and the part to annexed
colnames(anex) <- colnames(top)

# Add the anex data frame to the existing top
top <- rbind(anex,top)

# Plot top 3 qbs in new subset
par(mfrow = c(1,3))
for (i in 1:3) {
  radarchart(top[c(1,2,i+2),], title = rownames(top)[i+2])
}
```

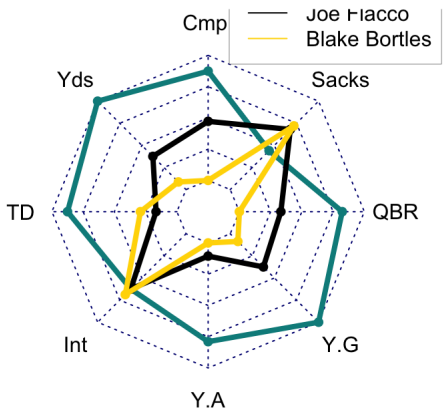



```
dev.off()
```

```
## null device
##           1
```

However, there's a more efficient manner that radar includes and that is to plot all the players on one chart. Suppose you want to compare three players on one single plot. Luckily, radarchart includes this automatically by simply passing those rows directly into the function as such. To distinguish between the lines, provide a call to pcol and supply a vector argument of the colors to be used. In addition, plwd and plty follow suit in a similiar manner by controlling the plot line aesthetics. plwd is the width of the plot lines while plty is the type of line provided (the dictionary for line types can be found here. To provide a legend, one can use the legend() function as we learned in the graphics package in class.

```
# Notice the subsetting within the function radarchart()
radarchart(top[1:(3+2),], pcol = c('cyan4','black','gold'), plty = 1, plwd = 4)
legend(x=0.135, y=1.6, legend = rownames(top[3:(3+2),]), col = c('cyan4','black','gold'), lty = 1,lwd = 1, box.lwd
= .25)
```

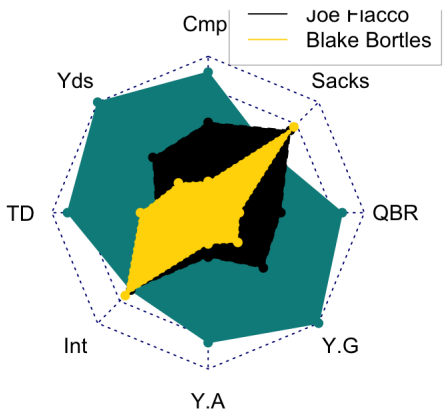


Addition customization

Adjust fill

You may choose, however, to add a fill in the radarchart like the examples provided at the beggining of the post. This can be accomplished by supplying an argument for pfcol in a similiar manner as pcol, with a vector of the colors to be used.

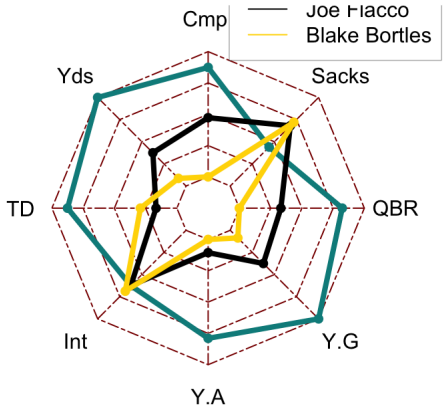
```
radarchart(top[1:(3+2),], pcol = c('cyan4','black','gold'), pfcol = c('cyan4', 'black', 'gold'))
legend(x=0.135, y=1.6, legend = rownames(top[3:(3+2),]), col = c('cyan4','black','gold'), lty = 1,lwd = 1, box.lwd
= .25)
```



Adjust grid line

The grid of the chart can also be further enhanced according to preferences by making calls to cglcol, cglwd, cglty which follow in the same fashion as pcol, plwd, plty:

```
radarchart(top[1:(3+2),], pcol = c('cyan4','black','gold'), plty = 1, plwd = 4,
           cglcol = 'red4',cglty = 6, cglwd = 1)
legend(x=0.135, y=1.6, legend = rownames(top[3:(3+2),]), col = c('cyan4','black','gold'), lty = 1,lwd = 1, box.lwd
= .25)
```



Conclusion

In todays world, being able to compare one thing to another is at the heart of many data visualization efforts. When one can expediate the interpretation process of data, especially in the case of readers that may not be familiar with statistical concepts, the use of visuals to immediately convey points is essential. Radar charts are a powerful technique that one can use to distinguish and compare characteristics of various objects whether they be players, cars, companies, etc. When one can convey certain aspects in an efficient manner then the value of one's work is heightened because the power of data anlytics is only as powerful as the universality of its interpretation.

Resources

1. https://www.forbes.com/forbes/welcome/?toURL=https://www.forbes.com/sites/leighsteinberg/2015/08/18/changing-the-game-the-rise-of-sports-analytics/&refURL=https://www.google.com/&referrer=https://www.google.com/
2. https://moderndata.plot.ly/radar-charts-in-r-using-plotly/
3. https://cran.r-project.org/web/packages/fmsb/fmsb.pdf
4. http://www.r-graph-gallery.com/142-basic-radar-chart/
5. https://rdrr.io/cran/fmsb/man/radarchart.html

6. https://stackoverflow.com/questions/13386656/adding-legends-to-a-radarchart
7. http://www.r-graph-gallery.com/143-spider-chart-with-saveral-individuals/