```
# Radarchart function, slightly modified from the original “fmsb” R package
(http://cran.r-project.org/web/packages/fmsb/index.html)

##################################################################
radarchartcentre <- function (df, axistype = 0, seg = 4, pty = 16, pcol = 1:8, plty = 1:6,
                plwd = 1, pdensity = NULL, pfcol = NA, cglty = 3, cglwd = 1,
                cglcol = "navy", axislabcol = "blue", title = "", subtitle = "", size = "", maxmin = TRUE,
                na.itp = TRUE, ...)
{
 if (!is.data.frame(df)) {
  cat("The data must be given as dataframe.\n")
  return()
 }
 if ((n <- length(df)) < 3)
  return()
 if (maxmin == FALSE) {
  dfmax <- apply(df, 2, max)
  dfmin <- apply(df, 2, min)
  df <- rbind(dfmax, dfmin, df)
 }
 plot(c(-1.2, 1.2), c(-1.2, 1.2), type = "n", frame.plot = FALSE,
    axes = FALSE, xlab = "", ylab = "", main = title, asp = 1)
 mtext(subtitle, side=3, cex=0.5, font=3)
 legend(-1.5, 1.4,c("Obs.","Exp."), lty=c(1,1), lwd=c(2,2), col=c("#ff000099","#0000ff70"), cex=0.8,
bty ="n")
 legend(0.1, 1.4,paste("n=",size), cex=0.9, bty="n")

 theta <- seq(90, 450, length = n + 1) * pi/180
 theta <- theta[1:n]
 xx <- cos(theta)
 yy <- sin(theta)
 points(0,0, pch=20, cex=0.5, pty = cglty, col = cglcol)
 for (i in 0:seg) {
  polygon(xx * (i + 1)/(seg + 1), yy * (i + 1)/(seg + 1), lty = cglty, lwd = cglwd, border = cglcol)

  if (axistype == 1 | axistype == 3)
   text(-0.05, (i + 1)/(seg + 1), paste(i/seg * 100, "(%)"), col = axislabcol)

  if (axistype == 4 | axistype == 5)
   text(-0.05, (i + 1)/(seg + 1), sprintf("%3.2f", i/seg), col = axislabcol)
 }

 arrows(0, 0, xx*1, yy * 1, lwd = cglwd, lty = cglty, length = 0, col = cglcol)
 if (axistype == 2 | axistype == 3 | axistype == 5) {
  text(xx[1:n], yy[1:n], df[1, 1:n], col = axislabcol)
 }
 text(xx * 1.2, yy * 1.2, colnames(df), cex=0.85)
 series <- length(df[[1]])
```

```
if (length(pty) < (series - 2)) {
  ptys <- rep(pty, series - 2)
  pcols <- rep(pcol, series - 2)
  pltys <- rep(plty, series - 2)
  plwds <- rep(plwd, series - 2)
  pdensities <- rep(pdensity, series - 2)
  pfcols <- rep(pfcol, series - 2)
}
else {
  ptys <- pty
  pcols <- pcol
  pltys <- plty
  plwds <- plwd
  pdensities <- pdensity
  pfcols <- pfcol
}
for (i in 3:series) {
  xxs <- xx
  yys <- yy
  scale <- (df[i, ] - df[2, ])/(df[1, ] - df[2, ])

  if (sum(!is.na(df[i, ])) < 3) {
    cat(sprintf("[DATA NOT ENOUGH] at %d\n%g\n", i, df[i,
                           ]))
  }
  else {
    for (j in 1:n) {
      if (is.na(df[i, j])) {
        if (na.itp) {
          left <- ifelse(j > 1, j - 1, n)
          while (is.na(df[i, left])) {
            left <- ifelse(left > 1, left - 1, n)
          }
          right <- ifelse(j < n, j + 1, 1)
          while (is.na(df[i, right])) {
            right <- ifelse(right < n, right + 1, 1)
          }
          xxleft <- xx[left] + (df[i,left] - df[2, left])/(df[1, left] - df[2,left])
          yyleft <- yy[left] + (df[i,left] - df[2, left])/(df[1, left] - df[2,left])
          xxright <- xx[right] + (df[i,right] - df[2, right])/(df[1, right] - df[2, right])
          yyright <- yy[right] + (df[i,right] - df[2, right])/(df[1, right] - df[2, right])
        if (xxleft > xxright) {
            xxtmp <- xxleft
            yytmp <- yyleft
            xxleft <- xxright
            yyleft <- yyright
            xxright <- xxtmp
            yyright <- yytmp
      }
      xxs[j] <- xx[j] * (yyleft * xxright - yyright * xxleft)/(yy[j]*(xxright - xxleft) - xx[j] * (yyright - yyleft))
      yys[j] <- (yy[j]/xx[j]) * xxs[j]
     }
     else {
      xxs[j] <- 0
      yys[j] <- 0
     }
    }
    else {
     xxs[j] <- xx[j] * ((df[i, j] - df[2, j])/(df[1, j] - df[2, j]))
     yys[j] <- yy[j] * ((df[i, j] - df[2, j])/(df[1, j] - df[2, j]))
    }
   }
   polygon(xxs, yys, lty = pltys[i - 2], lwd = plwds[i - 2], border = pcols[i - 2], density = pdensities[i - 2], col = pfcols[i - 2])
   points(xx * scale, yy * scale, pch = ptys[i - 2],col = pcols[i - 2])
  }
 }
}
```