```
#### =========================
### Tornado Plots
## Test code:
# paramNames <-  c( "Param 1 [Low/High CI]",
#                   "Param 2 [Low/High CI]",
#                   "Param 3 [-/+ 15%]",
#                   "Param 4 [-/+ 15%]"
# )
#
# # data structure: ymean       ymin    ymax
# data <- matrix(c(100, 80,     120,
#                  100, 25,     150,
#                  100, 95,     120,
#                  100, 75, 160), nrow = 4, ncol = 3, byrow = TRUE)
#
# data
# TornadoPlot(Parms = paramNames, Outcomes = data, titleName = "Tornado Plot")
# Parms = paramNames
# Outcomes = data
# titleName = "Tornado Plot"
#### Mark's notes
#### Date: 01 Otober 2017
# titleName is not used in the function; removed it
# added new options: main_title for the main title
# xlab for the x-axis label
# ylab for the y-axis label
# col1 for the first bar color
# col2 for the second bar color

### ====================================================
###     Function for plotting Tornado Diagrams
### ====================================================
TornadoPlot <-function(main_title, Parms, Outcomes, outcomeName, xlab, ylab, col1, col2){
  library(ggplot2)
  library(reshape2)
  library(scales)
  library(RColorBrewer)

  # Grouped Bar Plot
  # Determine the overall optimal strategy
  paramNames2 <- Parms

  # Combine the parameter list with the data
  ymean <- Outcomes[1,1]

  yMin <- Outcomes[,2] - ymean
  yMax <- Outcomes[,3] - ymean
  ySize <- abs(yMax - yMin)  #High value - Low value


  rankY<- order(ySize)
  nParams <- length(paramNames2)

  Tor <- data.frame(
    Parameter=c(paramNames2[rankY],paramNames2[rankY]),
    Level=c(rep("Low",nParams),rep("High",nParams)),
    value=ymean+c(yMin[rankY],yMax[rankY]),
    sort=seq(1,nParams)
  )

  #re-order the levels in the order of appearance in the data.frame
  Tor$Parameter2 <- ordered(Tor$Parameter, Tor$Parameter[1:(length(Tor$Parameter)/2)])
  # Tor$Parameter2 <- factor(Tor$Parameter, as.character(Tor$Parameter))
  #Define offset as a new axis transformation. Source:
http://blog.ggplot2.org/post/25938265813/defining-a-new-transformation-for-ggplot2-scales
  offset_trans <- function(offset=0) {
    trans_new(paste0("offset-", format(offset)), function(x) x-offset, function(x) x+offset)
  }
```

```
  #Plot the Tornado diagram.
  txtsize<-12
  print(
    ggplot(Tor[Tor$Level=="Low",], aes(x=Parameter2,y=value, fill=level)) +
    geom_bar(stat="identity", fill=col1) +
      ggtitle(main_title, subtitle = outcomeName) +
      scale_fill_discrete("Parameter Level: ", l=50)+
      scale_y_continuous(name=xlab, trans=offset_trans(offset=ymean)) +
      scale_x_discrete(name=ylab) +
      geom_bar(data=Tor[Tor$Level=="High",], aes(x=Parameter2,y=value, fill=level),
stat="identity", fill=col2, alpha=1.0) +
      geom_hline(yintercept = ymean, linetype = "solid", size=0.5) +
      theme_bw(base_size = 14) + theme(panel.border = element_blank(), panel.grid.major =
element_blank(),
      panel.grid.minor = element_blank(), axis.line = element_line(colour = "black")) +
      coord_flip() +
      theme(legend.position="bottom",
            legend.title=element_text(size = txtsize,angle = 0, hjust = 1),
            legend.key = element_rect(colour = "black"),
            legend.text = element_text(size = txtsize),
            title = element_text(face="bold", size=15),
            axis.title.x = element_text(face="bold", size=txtsize),
            axis.title.y = element_text(face="bold", size=txtsize),
            axis.text.y = element_text(size=txtsize),
            axis.text.x = element_text(size=txtsize),
            axis.ticks.y = element_blank())
  )
  # ggsave(paste("results/", titleName,".png"))
}
```