Problem Set 4

Prateek Kumar

09/28/2018

1. Custom population pyramids

I have selected the database of `India`, for years `1991`,`2001` and `2011`. The databases have been choosen under the category “Population by five-year age groups”, so for each of the three years we get a table with the population pyramid. Later we have their pyramid plots and the comparison of the three years.

```
India_1991 <- read_excel("India_1991.xlsx",skip = 1) # Reading the data from the excel
India_2001 <- read_excel("India_2001.xlsx",skip = 1)
India_2011 <- read_excel("India_2011.xlsx",skip = 1)
```

Setting the age labels

```
agelabels <- India_1991$Age
```

For the coloring here we are using the `colorspace` package and for plotting the pyramids we are using plotrix

```
library("colorspace", lib.loc="~/R/win-library/3.5")
library("plotrix", lib.loc="~/R/win-library/3.5")
```

Setting up the colors for the male and female data for all the three databases. Here I am breaking the data into quantiles and assigning color to each quantile for both the male and female percentage, so that we get an insight of population percentage distribution over the age groups.

```
# setting up the color for the year 1991
c_m_91 <- cut(India_1991$`Percent Male`, breaks=quantile(India_1991$`Percent Male`,
                                                         c(0, 0.25, 0.5, 0.75, 1)))
c_f_91 <- cut(India_1991$`Percent Female`, breaks=quantile(India_1991$`Percent Female`,
                                                           c(0, 0.25, 0.5, 0.75, 1)))

# setting up the color for the year 2001
c_m_01 <- cut(India_2001$`Percent Male`, breaks=quantile(India_2001$`Percent Male`,
                                                         c(0, 0.25, 0.5, 0.75, 1)))
c_f_01 <- cut(India_2001$`Percent Female`, breaks=quantile(India_2001$`Percent Female`,
                                                           c(0, 0.25, 0.5, 0.75, 1)))
```

```
# setting up the color for the year 2011
c_m_11 <- cut(India_2011$`Percent Male`, breaks=quantile(India_2011$`Percent Male`,
                                                         c(0, 0.25, 0.5, 0.75, 1)))
c_f_11 <- cut(India_2011$`Percent Female`, breaks=quantile(India_2011$`Percent Female`,
                                                           c(0, 0.25, 0.5, 0.75, 1)))
```

The pyramid plots for male and female percentage over the age group for the three years.

```
pyramid.plot(lx = India_1991$`Percent Male`,rx = India_1991$`Percent Female`,
             labels = agelabels, gap = 1, main = "Indian Population Pyramid 1991",
             show.values = T, lxcol=c_m_91,rxcol=c_f_91, space = 0.4,
             top.labels=c("Male","Age","Female"), unit="Percentage", ppmar=c(4,3,4,3), labelcx
= 0.8,
             xlim = c(15,15), laxlab=c(0,2,4,6,8,10,12,14), raxlab=c(0,2,4,6,8,10,12,14),
             add = F, ndig = 1, do.first = NULL)
```

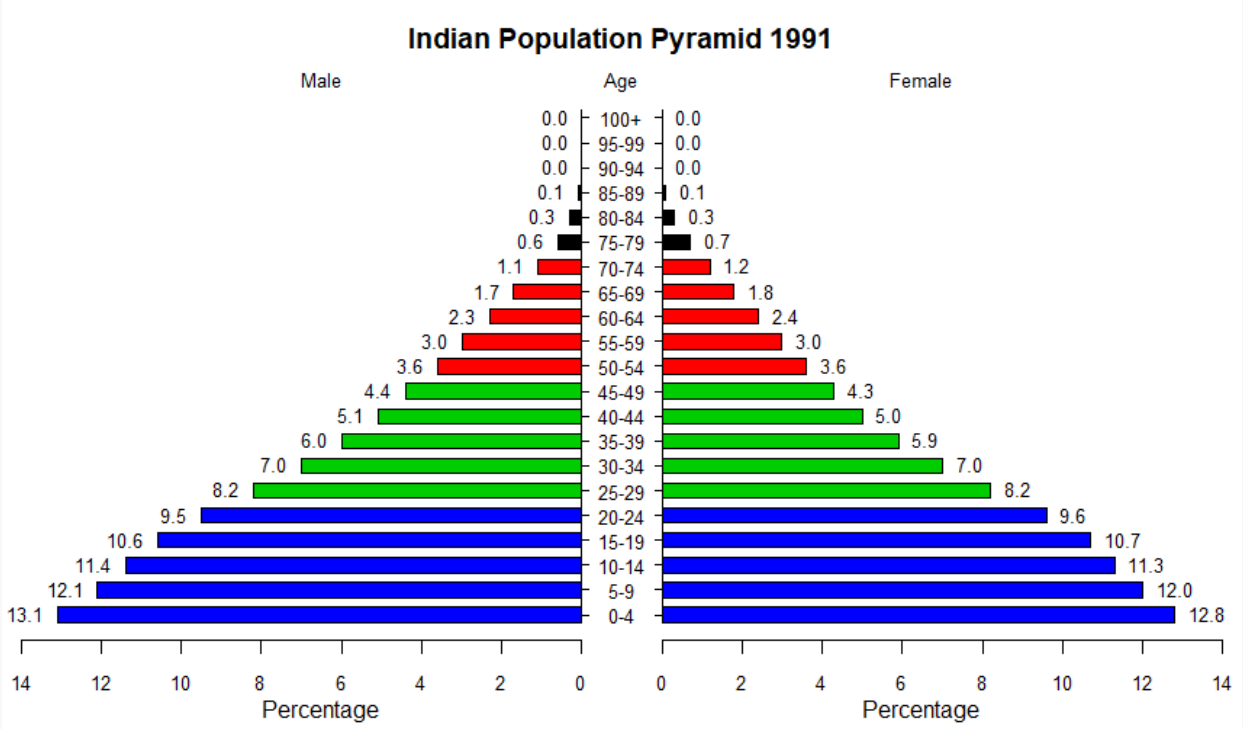

Figure 1: The pyramid plot for male and female populations in India in year 1991.

```
pyramid.plot(lx = India_2001$`Percent Male`,rx = India_2001$`Percent Female`,
             labels = agelabels, gap = 1, main = "Indian Population Pyramid 2001",
             show.values = T, lxcol=c_m_01,rxcol=c_f_01, space = 0.4,
             top.labels=c("Male","Age","Female"), unit="Percentage", ppmar=c(4,3,4,3), labelce
x = 0.8,
             xlim = c(15,15), laxlab=c(0,2,4,6,8,10,12,14), raxlab=c(0,2,4,6,8,10,12,14),
             add = F, ndig = 1, do.first = NULL)
```

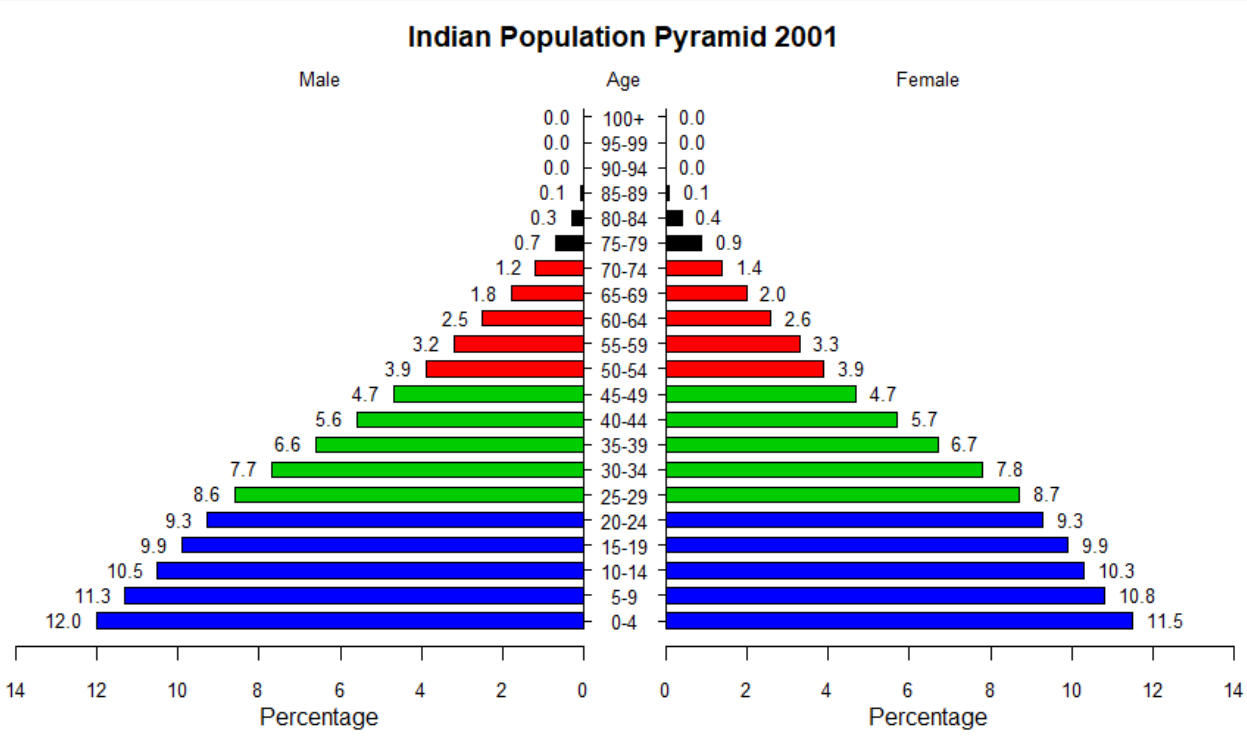

Figure 2: The pyramid plot for male and female populations in India in year 2001.

```
pyramid.plot(lx = India_2011$`Percent Male`,rx = India_2011$`Percent Female`,
             labels = agelabels, gap = 1, main = "Indian Population Pyramid 2011",
             show.values = T, lxcol=c_m_11,rxcol=c_f_11, space = 0.4,
             top.labels=c("Male","Age","Female"), unit="Percentage", ppmar=c(4,3,4,3), labelce
x = 0.8,
             xlim = c(15,15), laxlab=c(0,2,4,6,8,10,12,14), raxlab=c(0,2,4,6,8,10,12,14),
             add = F, ndig = 1, do.first = NULL)
```

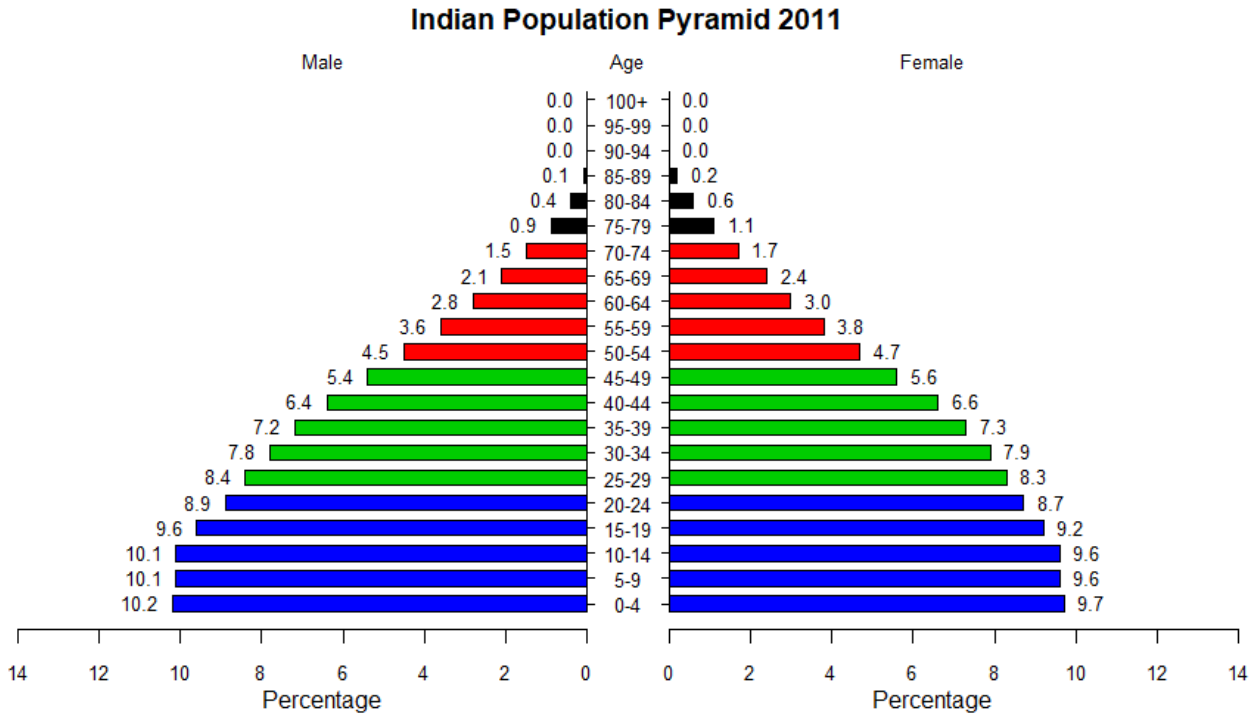

Figure 3: The pyramid plot for male and female populations in India in year 2011.

- So, I have taken the three pyramid data of India for years 1991, 2001 and 2011. I chose the three sets to see the percent change in male and female population over every 10-year gap for all the different age groups.

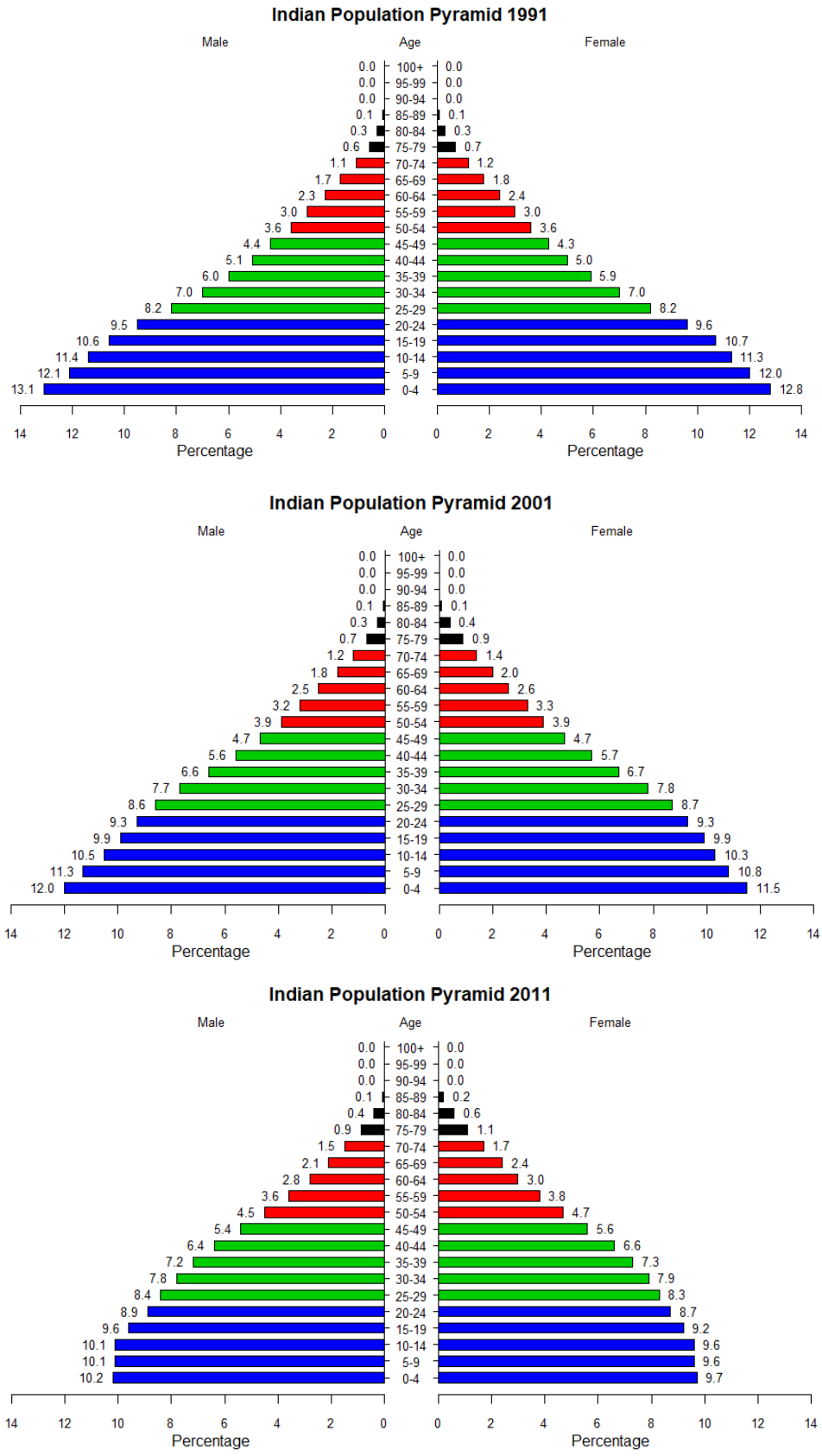

Figure 4: Comparing the three pyramids

- When we look at the three population pyramids we do not see the actual population in numbers rather the pyramids display percentages and shows what portion of people fall into each age group.
- On observing the three pyramids we can see how was the population trend in the past and we can examine the current resident profile and can eventually project how the population will increase/decrease in the future.
- In the pyramids of 1991 and 2001 we see a more uniform population size across age groups. Old generations being replaced by new generations of approximately same size. Both the graphs look like a steep pyramid so the population growth is faster, old generations are producing larger new generations.
- Whereas in pyramid of 2011 we see the bottom to be much more rectangular than the years of 1991 and 2001. This means that in the age group from 0-14 the population growth is slower so we see a uniform population size across the 3 age groups. This seems to be a good sign as India being the most populated country is the world is aiming to control the population growth to tackle the issues of overpopulation.
- In the three pyramids we see that the pyramid shape has a larger bottom than the top so, a larger population are in their reproductive years or have not even reached reproductive age. As a result, there is much potential for growth.

2. Demonstration of a new Graphics Function or Package (Diamond Plot)

- A diamond plot is a dot plot to which a diamond shape has been added to show the confidence interval for the population mean.
- The underlying idea is that point estimates are uninformative, and it would be better to focus on confidence intervals. The problem of the points with error bars that are commonly employed is that the focus the audience's attention on the upper and lower bounds, even though those are the least relevant values. Using diamonds remedies this.

```
fifa <- read_excel("FIFA_18.xlsx")
la_liga <- read_excel("FIFA_LA_LIGA.xlsx")
library("userfriendlyscience", lib.loc="~/R/win-library/3.5")
```

- Here I have taken FIFA 18 players data of 9 teams. I have imported the excel to the variable ‘fifa’ and the La Liga data to the variable la_liga.
- The diamond plot uses the library *"userfriendlyscience"* so we have to include the library in our code.

- The first way to plot diamond plot is *"meansDiamondPlot"*. Here the diamonds center on the means of the columns and its ends extend to the confidence intervals. Here I have plotted for 95% confidence.

```
meansDiamondPlot(fifa,
                 items = c('Acceleration','Aggression', 'Agility','Ball control','Finishing'),
                 conf.level = .95, jitterWidth = .5, jitterHeight = .5,
                 dataAlpha = 0.5, dataSize = 2,showData = T,
                 xlab = 'Value and Means', theme = theme_light(), xbreaks = 'auto',
                 outputFile='figure 1.png',outputWidth=15, outputHeight=15)
```

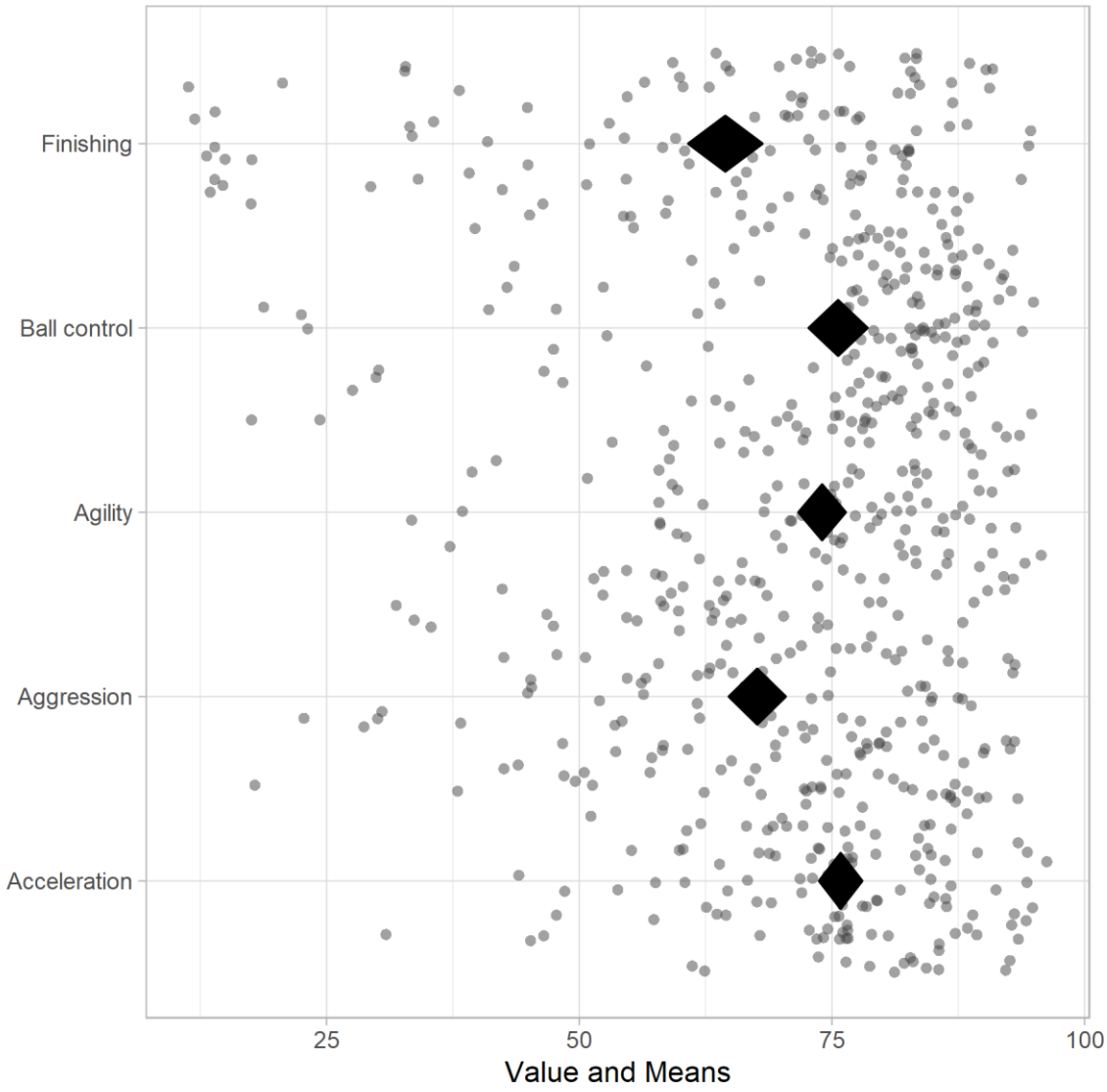

Figure 5: meansDiamondPlot on 95% confidence

- For 50% confidence

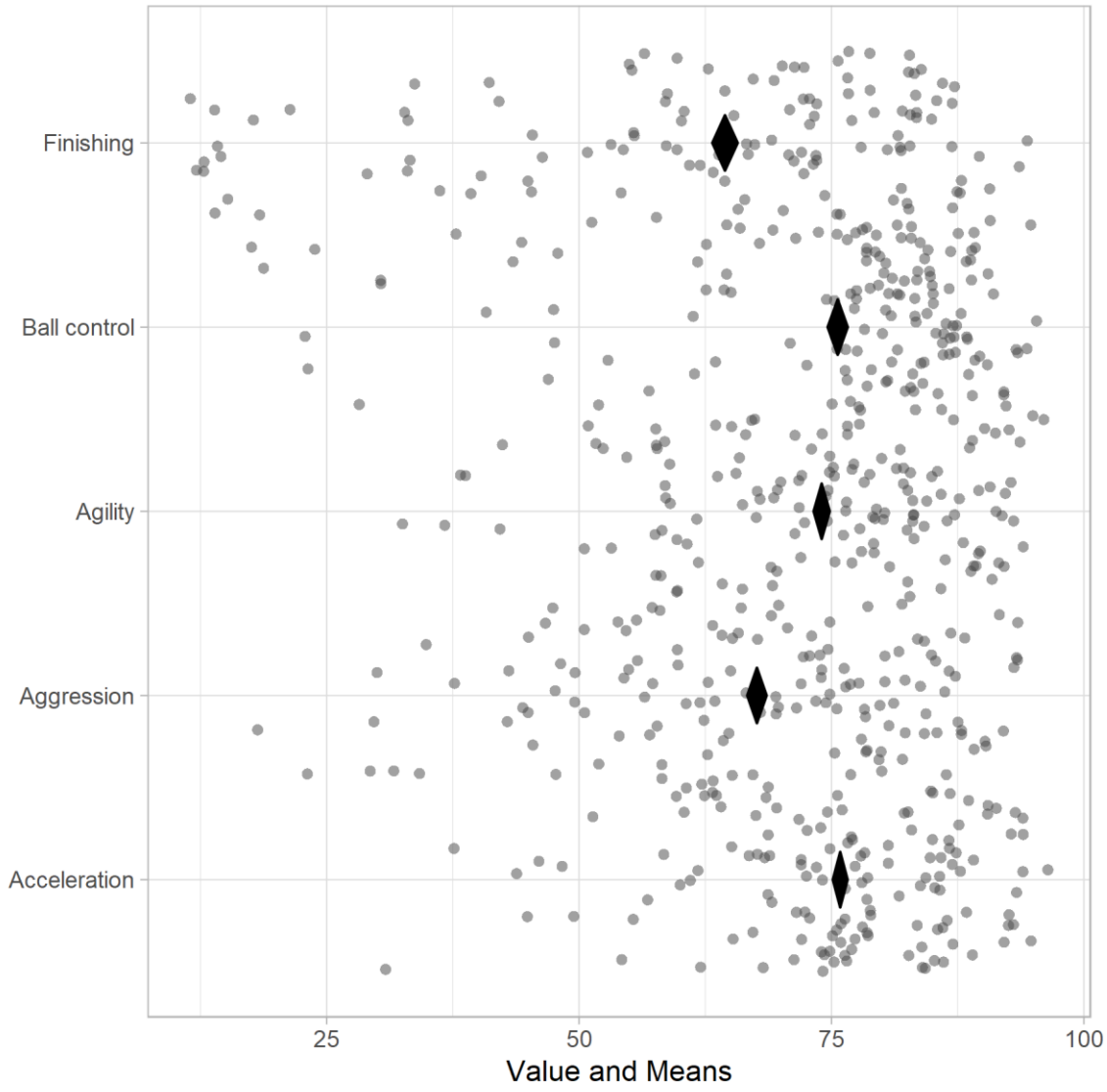

Figure 6: meansDiamondPlot on 50% confidence

- We can clearly see how diamonds get skewed due to less confidence interval.
- Here I have used the color gradient to view the diamonds better.

```
meansDiamondPlot(fifa,
                 items = c('Acceleration','Aggression','Agility','Ball control','Finishi
ng'),
                 conf.level = .95, jitterWidth = .5, jitterHeight = .5,
                 dataAlpha = 0.5, dataSize = 2,
                 xlab = 'Value and Means', theme = theme_light(), xbreaks = 'auto',
                 generateColors = c('navy','gold'),
                 outputFile='figure 2.png',outputWidth=15, outputHeight=15)
```

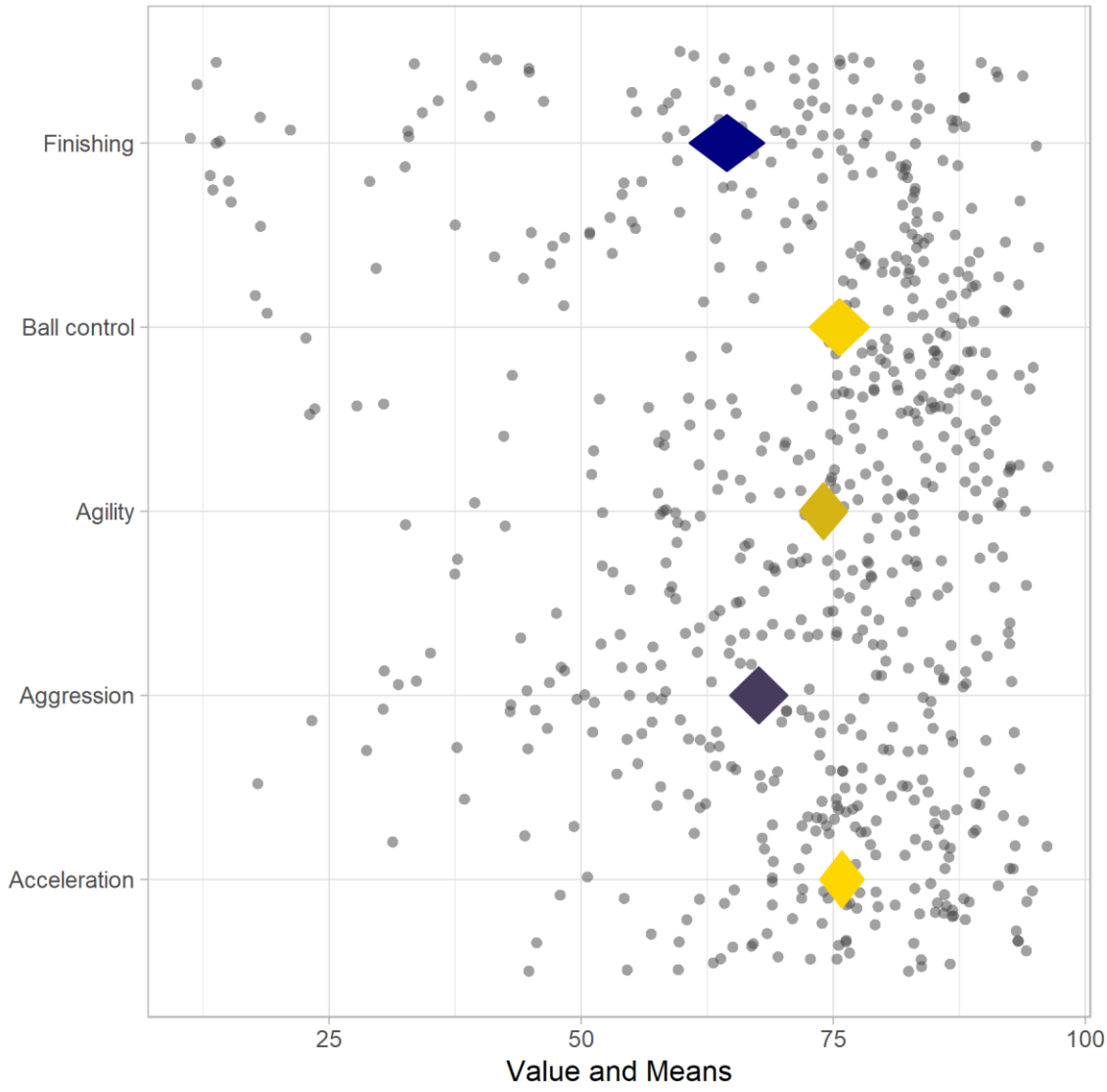

Figure 7: meansDiamondPlot with colored diamonds

- The next function is *"biAxisDiamondPlot"*. This function is basically a *meansDiamondPlot*, but extended to allow specifying subquestions and anchors at the left and right side. This is convenient for psychological questionnaires when the anchors or dimensions were different from item to item.

```
biAxisDiamondPlot(dat=fifa,
                  items=c('Acceleration', 'Aggression', 'Agility', 'Ball control','Finis
hing'),subQuestions=c('How much acceleration?', 'How much aggression?', 'How is the agil
ity?','How is the ball control?','How is the finishing?'),
                  leftAnchors=c('less', 'less','poor','poor','worse'),
                  rightAnchors=c('more', 'more','excellent','great','outstanding'),
                  decreasing = F, conf.level = 0.95, showData = T, dataAlpha = 1,
                  dataColor = 'black',
                  diamondColors = c('red','blue','green','gold','cyan'),
                  jitterWidth = 0.5, jitterHeight = 0.3,
```

```
                xbreaks='auto', xLabels = NA,
                xAxisLab = paste0("Value and ", round(100 * 0.95, 2),"% CIs"),
                drawPlot = T, baseSize = 1, dotSize = 1.5, baseFontSize = 10,
                theme = theme_light(),
                outputFile='figure 3.png',outputWidth=15, outputHeight=15);
```

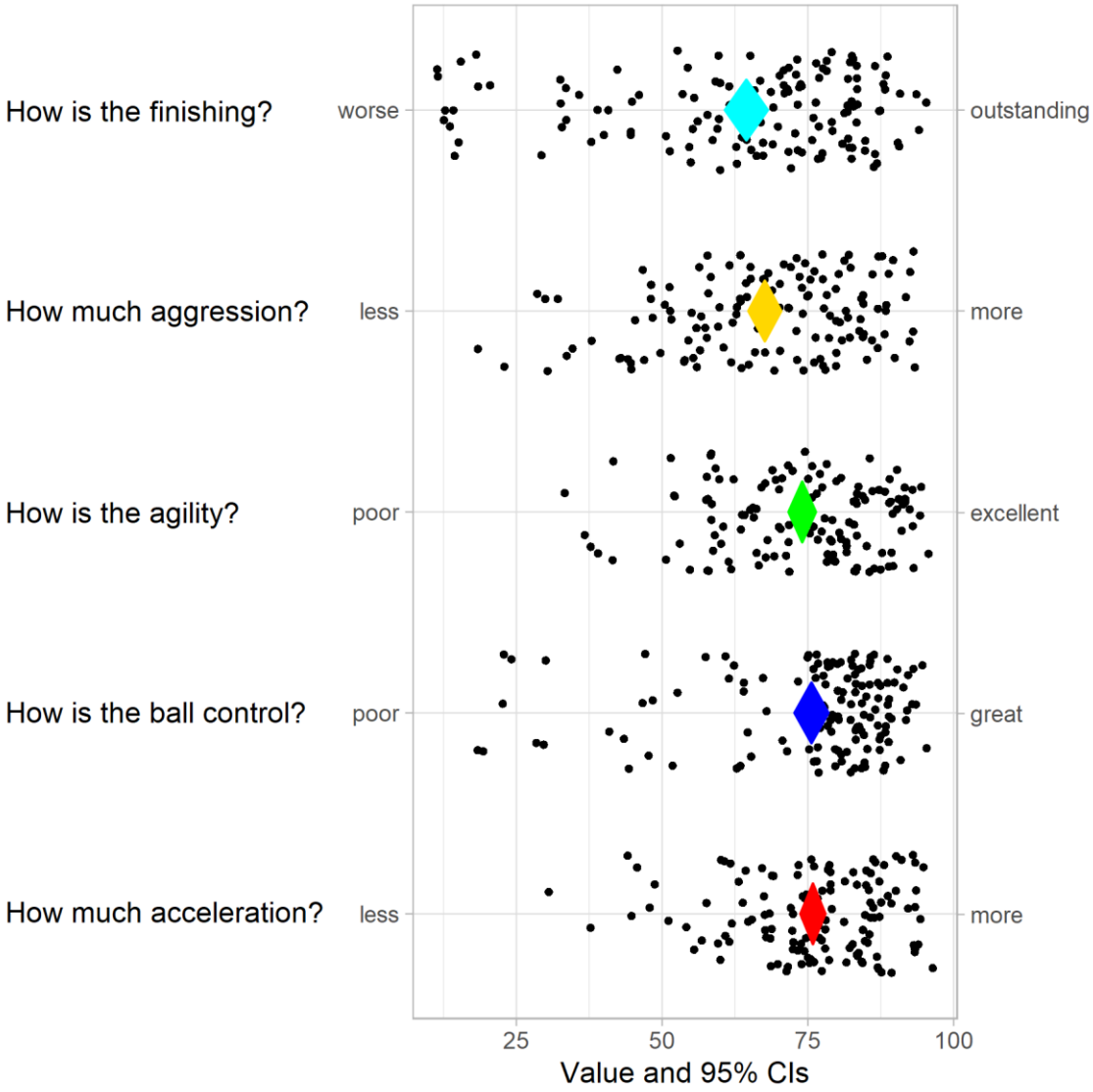

Figure 8: biAxisDiamondPlot mainly used for psychological questionnaires

- The next function is *"meansComparisonDiamondPlot"*. This function draws multiple diamond plots to conveniently to compare subgroups or different samples.
- They are both based on a univariate diamond plot where colors are used to distinguish the data points and diamonds of each subgroup or sample.

```
meansComparisonDiamondPlot(la_liga,
                           items=c('Acceleration', 'Aggression', 'Agility', 'Ball contro
l','Finishing'),
                           compareBy='Club', labels = NULL, conf.level = 0.95,
                           showData = T, dataAlpha = 0.3, dataSize = 2,
                           alpha = 0.33, jitterWidth = 0.5, jitterHeight = 0.4,
                           xlab = 'Scores and means', ylab = NULL,
                           theme = theme_light(), showLegend = T,
                           lineSize = 1,xbreaks=c(0,20,40,60,80,100),
                           outputFile='figure 4.png',outputWidth=15, outputHeight=15)
```

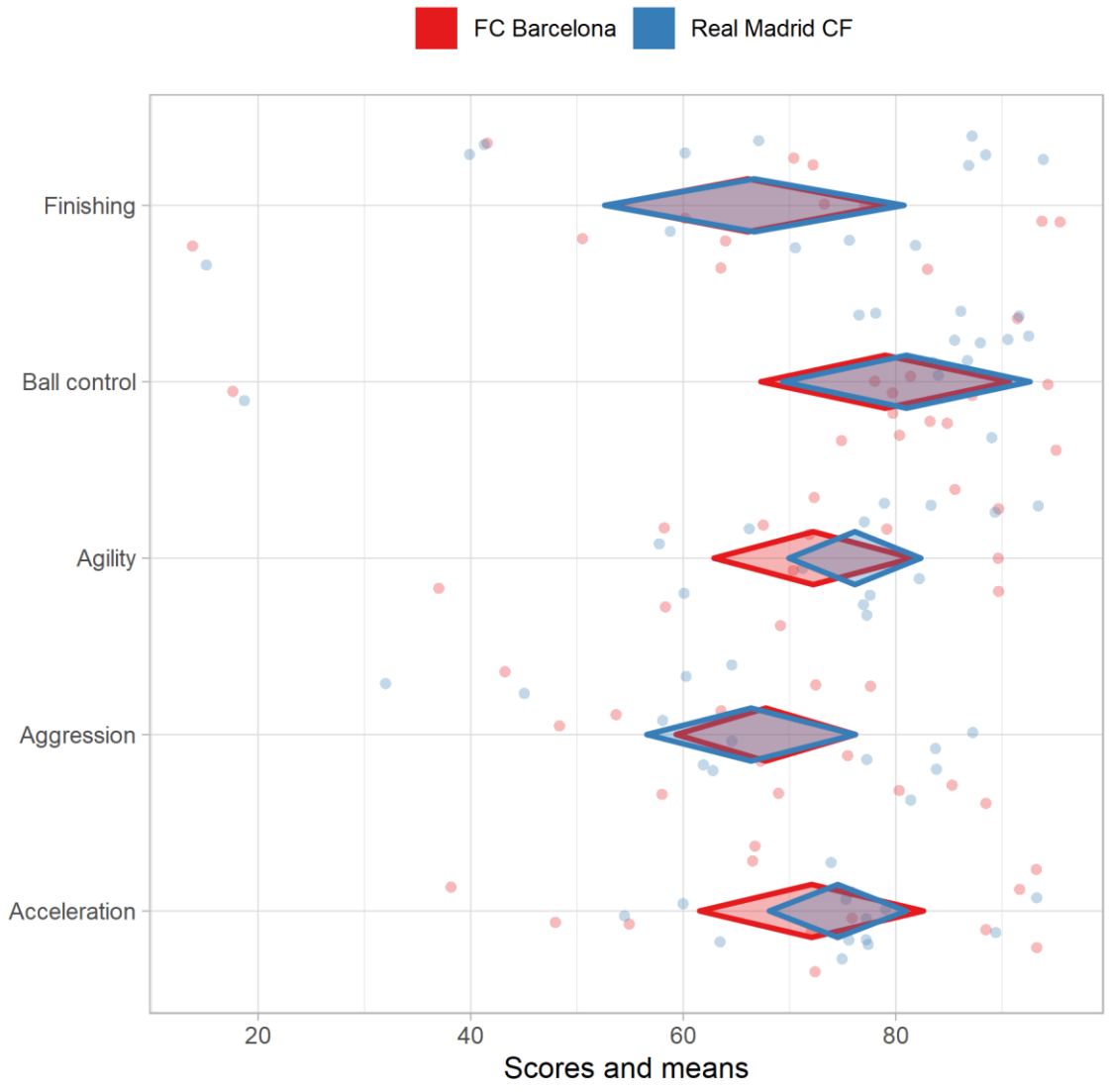

Figure 9: meansComparisonDiamondPlot for comparing the subgroups