Visualize factor(s)

Today, I am going to cover how to visualize factors.

We have learned to generate frequency tables for one factor. After we have the frequency table. We can plot it using barplot.

Let's generate a barplot of the race variable in the test data sets.

First, we need to convert it to a factor using as.factor function.

```r
StudentsPerformance$Race <- as.factor(StudentsPerformance$Race)
```

Since it is a factor in R memory, we can now generate a bar plot using the following commands:

```r
#Generate a freqency table of Race
tbl.race <- table(StudentsPerformance$Race)
#Bar plot of the frequency table
barplot(tbl.race, main="Race Distribution",
    xlab="Race Group")
```

The default bar plot is a vertical bar. If you want to have a horizontal bar, you can set the horiz argument to be TRUE.

```r
#Generate a freqency table of Race
tbl.race <- table(StudentsPerformance$Race)
#Bar plot of the frequency table
barplot(tbl.race, main="Race Distribution",
    ylab="Race Group", horiz=TRUE)
```

Similarly, we can generate a cross table distribution for two factors. Then we can produce a grouped bar plot.

```r
StudentsPerformance$Gender <- as.factor(StudentsPerformance$Gender)

#Generate a cross table of Race and Gender
tbl.race <- xtabs(~ Race + Gender, data = StudentsPerformance)
#Bar plot of the frequency table
barplot(tbl.race, main="Race Distribution",
    col=c("black", "blue","brown", "yellow","red"), legend = rownames(tbl.rac
e))
```

We specify the color argument for the race and the legend of it.

We can also generate a stack bar plot by setting the parameter of beside to be TRUE.

```
#Generate a cross table of Race and Gender
tbl.race <- xtabs(~ Race + Gender, data = StudentsPerformance)
#Bar plot of the frequency table
barplot(tbl.race, main="Race Distribution",
   col=c("black", "blue","brown", "yellow","red"), legend = rownames(tbl.rac
e),
   beside = TRUE)
```

We can also visualize a continuous variable against a factor. Then it is easy for us to compare the different categories on the same plot.

```
#convert the Race to factor
races <- as.factor(StudentsPerformance$Race)
#set the colors for the Race
my.cols <- c("black", "blue","brown", "green","red")
dotchart(StudentsPerformance$MathScore,
         groups = races, gcolor = my.cols,
         color = my.cols[races],
         cex = 0.6,  pch = 19, xlab = "math  score")
```

We can also summarize the five import numbers using box plot for different races.

```
boxplot(MathScore ~ Race, data=StudentsPerformance, notch=TRUE,
         col=c("green"),
   main="Math Scores by Races", xlab="Race")
```

We set the notch to be true by drawing a notch in each side of the boxes. "If the notches of two plots do not overlap this is 'strong evidence' (95% confidence) that the two medians differ" According to Graphical Methods for Data Analysis (Chambers, 1983, p.62). We can see that group E has the largest median of the math score and group A has the smallest. The two notches of Group A and E do not overlap, therefore there is a strong evidence that the two medians are different.

Sometimes, we may use the graphs in a presentation or a report. To do it, we first need to save graphs to files in R. Then insert the graphs later into a presentation or report.

To save it to a file, we typically need the following three steps:

For example, we save the previous graph to a jpeg file using the following R codes

```r
#open the file using double backslash or single forward slash
jpeg("C:\\Users\\yliu3\\Documents\\Data Science\\math_race.jpg")
#plot the graph
boxplot(MathScore ~ Race, data=StudentsPerformance, notch=TRUE,
        col=c("green"),
    main="Math Scores by Races", xlab="Race")
#close the file
dev.off()
```

- First step, specify the output file format. R supports the following file formats: pdf; png, jpeg, postscript, bmp etc.

- Second step, create the plot
- Third step, close the current file using dev.off()

Note that no graph shows in the R windows. The file is saved in the specified directory. We can open this file at the specified directory.