Today, I am going to cover plot factors using ggplot2.

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

Let's load the data set into R memory.

library(readxl)

StudentsPerformance <- read_excel("C:/Users/yliu3/OneDrive - Maryville University/Online DSCI502 R Programming/DataSets/StudentsPerformance.xlsx")

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

First, we need to convert race to a factor

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

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

```
ggplot(data = StudentsPerformance, aes(x = Race, y=(..count..)))+geom_bar()
```

The default bar plot is a vertical bar. If you want to have a horizontal bar, you can set the horiz argument by calling the coord_flip() function.

```
ggplot(data = StudentsPerformance, aes(x = Race, y=(..count..)))+geom_bar() +
 coord_flip()
```

Similarly, we can produce a grouped bar plot for two factors.

```
ggplot(data = StudentsPerformance, aes(x=Race, y= ..count..)) + geom_bar(aes
(fill = Gender))
```

Note that we specify the variables to plot: x-axis is the factor Race and y-axis shows the count denoted by ..count.., then we fill the bar by the second factor Gender. Using this way, ggplot2 produces a bar plot of two factors.

A stacked barplot is created by default. You can use the function position_dodge() to change this.

We can also generate a dodge bar plot by setting the parameter of position to be "dodge".

```
ggplot(data = StudentsPerformance, aes(x=Race, y= ..count..)) + geom_bar(aes
(fill = Gender), position = "dodge")
```

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.

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

ggplot(data = StudentsPerformance,aes(x = Race, y = MathScore)) + geom_dotplo
t( aes(fill = Race),
                 binaxis = "y",
                 binwidth = 2,
                 stackdir = "center" )
```

The dotplot specifies the variables to plot, the continuous variable, MathScore on the y-axis and the discrete variable on the x-axis. Since the continuous variable is on the y-axis, we need to set the binaxis to be “y” since the default is “x”. The binwidth specifies the bin width.

We can also add an average line by using the stat_summary function and specifying the three functions; fun.y, fun.ymin and fun.ymax to be the mean using crossbar.

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

ggplot(data = StudentsPerformance,aes(x = Race, y = MathScore)) + geom_dotplo
t( aes(fill = Race),
                 binaxis = "y",
                 binwidth = 2,
                 stackdir = "center" ) + stat_summary(fun.y = mean, fun.ymin
= mean, fun.ymax = mean,
                 geom = "crossbar")
```

Let’s generate a box plot of Math scores against Race by using the geom_boxplot function.

```
ggplot(data=StudentsPerformance, aes(x=Race, y=MathScore)) +
  geom_boxplot(aes(col= Race ), notch = TRUE)
```

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 the 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 need to save graphs to files in ggplot2. To save it to a file, we typically need two steps.

- First step, create the plot using ggplot2
- Second step, save the last plot using the ggsave() function. This function has several arguments: filename, width, height, and units.

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

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

```
#save the file by specifying the file size
ggsave("C:\\Users\\yliu3\\Documents\\Data Science\\math_race.jpg", width = 2
0, height = 15, units = "cm")
```

We can open the file at the specified directory