Today, I am going to cover ggplot2 to visualize data.

ggplot2 is a visualization package developed by Hadley Wichham in 2005. It is based on the Grammar of Graphics created by Leland Wilkinson. ggplot2 is probably the best package for data visualization in R. The idea of ggplot2 is to use the uniform grammar to define the components of graphics.

To use ggplot2, we need to install it manually since it isn't included in the core packages of R. To install it, click Tools menu, then click Install Packages. Finally type ggplot2 in the textbox under packages and click the Install button. It takes several minutes to install it.

Histogram plots

Let's load the data set into memory:

```
library(readxl)
StudentsPerformance <- read_excel("C:/Users/yliu3/OneDrive - Maryville Univer
sity/Online DSCI502 R Programming/DataSets/StudentsPerformance.xlsx")
```

Let's make a histogram of math scores using the ggplot2 command:

```
#We need to load the ggplot2 package into memory first
library(ggplot2)

ggplot(data = StudentsPerformance, aes(x=MathScore)) + geom_histogram()
```

We set the data source based on the data frame StudentsPerformance first. Then we specify the variable to plot in the aes parameter. Finally, we add the geometric object histogram using **+** operator. We can see the graph above shows the distribution of math scores. Most scores concentrate around 65.

We can step up titles and labels using the following command and + operator.

```
ggplot(data = StudentsPerformance, aes(x=MathScore)) + geom_histogram()+ ggti
tle("Histogram of Math Test Scores") + xlab(" Math Scores") + ylab("Count") +
 xlim(0,100) + ylim(0,150)
```

- ggtitle("main_tile""): the title of the graph
- xlab("x_lab""): horizontal label
- ylab("y_lab""): vertical label
- xlim: the range of x values from the smaller number to a larger number.
- ylim: the range of y values from the smaller number to a larger number.

To generate the density plot, we can use the geom_density() function to plot it.

```
ggplot(data = StudentsPerformance, aes(x=MathScore)) + geom_density(color = "
blue", fill = "green")+ ggtitle("Density of Math Test Scores")
```

Note that we specify the line color and fill color respectively in the geom_density funcion.

If we want to plot histogram and density on the same graph, we can use the following R codes;

```
# Histogram with density plot and vertical mean line
ggplot(data = StudentsPerformance, aes(x=MathScore))+
 geom_histogram(aes(y=..density..), colour="black", fill="white")+
 geom_density(alpha=.2, fill="pink") + geom_vline(aes(xintercept=mean(MathSco
re)),
             color="green", linetype="dashed", size=1)
```

The codes above perform the following tasks:

- Produces a histogram with probability instead of frequency by setting y=..density..
- Add density plot to the histogram using the geom_density() function and set face transparency using alpha and fill in color.
- Add the vertical mean line by using the geom_vline() function and specifying the x-intercept.

For continuous variables, we may plot all the points on the dot chart to have a complete picture of the dataset.

```
ggplot(data = StudentsPerformance, aes(x=MathScore))+ geom_dotplot() + ylim
(0, 110)
```

For continuous variables, we can summarize the distribution of the data using Boxplots.
Boxplots summarize five important numbers according to Wiki Box plot and whisker of boxplot

- The bottom of the box shows the 25th percentile, denoted by Q_1 (the lower quartile)
- The middle of the box shows the 50th percentile, denoted by Q_2 (median)
- The top of the box shows the 75th percentile, denoted by Q_3 (the upper quartile)
- The upper whisker approximately shows the “maximum”
- The Lower whisker approximately shows the “minimum”"

The points below the lower whisker or above the upper whisker are the outliers

```
ggplot(data = StudentsPerformance, aes(x = "", y=MathScore))+geom_boxplot(out
lier.colour="red", outlier.shape=16)
```

Since we only plot a continuous variable, we have to set the x-axis to be empty and let y denote the variable to plot.

We have covered how to plot one continuous variable. We are also interested in plotting two continuous variables to find the relationship between them.

For example, we would like to plot math score (y-axis) against reading score (x-axis)

```
ggplot(data = StudentsPerformance, aes(x = ReadingScore, y=MathScore))+geom_p
oint()
```

We first specify the data source and two variables to plot in ggplot() function, then we add the geom_point. It is easy to see that students with higher reading scores tend to have higher math scores.

We may want to add a trend line between these two scores. We can use the following R codes:

```
ggplot(data = StudentsPerformance, aes(x = ReadingScore, y=MathScore))+geom_p
oint()+geom_smooth()
```

Note that, we only need to add the trend line by adding geom_smooth function. It is much easier than we did using basic graphics in R