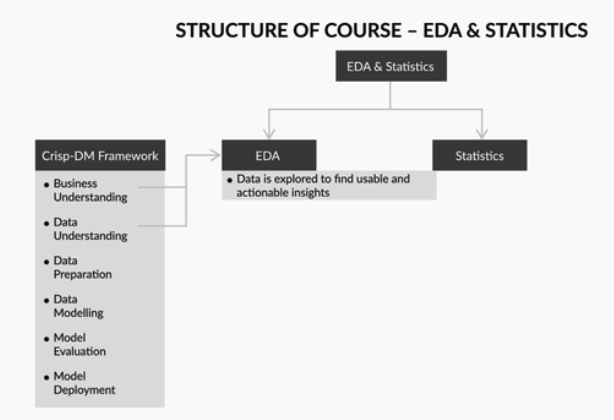
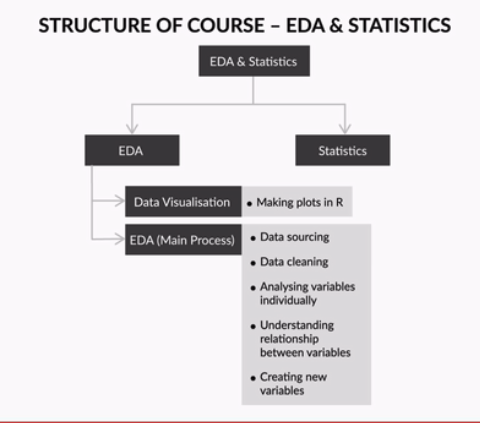
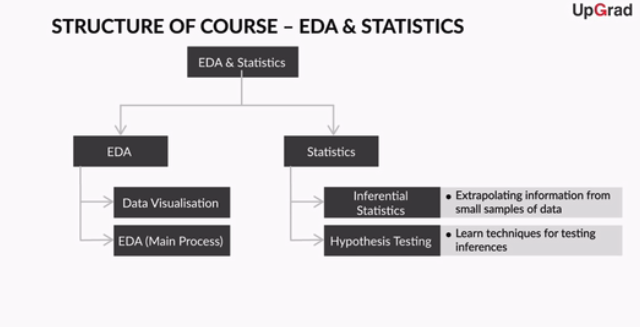
Data Visualization







Welcome to the module on 'Data Visualisation - How to  Make Data Presentable!'

**In this module**

In the previous sessions, you learnt the multiple steps of the **CRISP-DM framework.** One of the steps in the CRISP-DM framework was **data understanding**. To a large extent, the understanding of data is dependent on how the data is presented. It is here that the science of data visualisation or the visual representation of data becomes important.

Also, data visualisation is heavily used in exploring data and finding insights from it. Thus, this module is also a prerequisite for the next one - exploratory data analysis. Since most of your future work will be in R, this module will be heavily skewed towards how to plot charts in R.

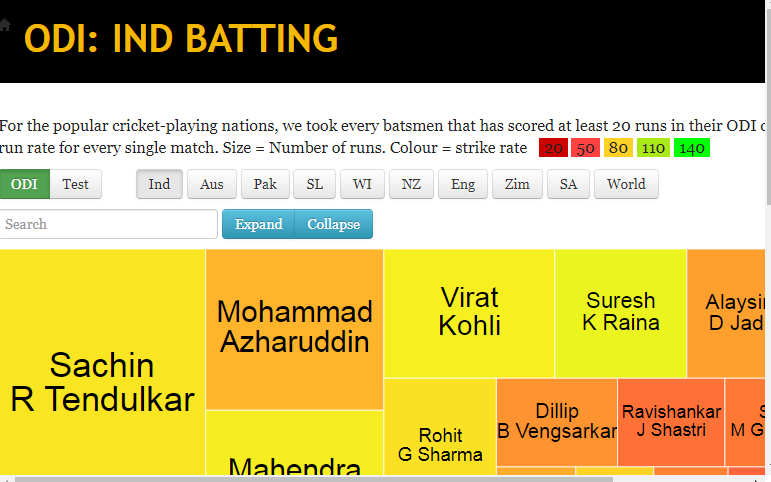
In this module, you will learn about the various ways to represent data and learn how to construct and modify various charts in R.

Let's begin this session by understanding how data visualisation can add value to the information you want to convey. Graphics and visuals, if used intelligently and innovatively, can convey a lot more than what raw data alone can convey. You will learn about the pre-attentive and attentive attributes, which are the differentiating factors in making your visuals interesting for the reader.

Next, we will delve deeper into the different types of charts such as:

1. Line chart
2. Bar chart
3. Histogram
4. Pie chart
5. Stacked bar chart
6. Scatter plot
7. Box plot
8. Grouped bar chart

You saw how data visualisation can improve data density and improve the amount of information being conveyed. Just imagine how difficult it would be to interpret a spreadsheet with the score of each inning of each batsman being recorded along with his strike rate. In this case, data visualisation can help you figure out many insights just by looking at the plot. You can see the visualisation here:



You saw how a tree map diagram showed how multiple companies and sectors react to the budget. You can find the interactive graph [here](https://gramener.com/budget/?Year=2007).

An important point to remember is that visualisations should be accompanied by a voice or text narrative if possible - it improves the experience of the user drastically.

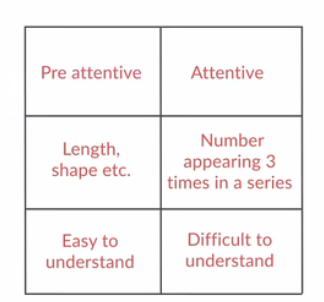
Next, let's see how visualisation can help in visual exploratory analytics. Here, you will see how data visualisation helped in understanding the connections between different software and clustering them together based on common features. You can find the visual[here](https://gramener.com/software/).

# Visualisations - The World of Imagery

One picture is worth a thousand words. That’s the power of visual imagery. A message which cannot be conveyed through a large set of texts and tabular data can easily be presented through visuals. We often use graphics to make sense of large and complex sets of information. This makes data visualisation a very important step for data understanding.

However, to make sense through visuals, it's important to first understand the different types of visuals and their common usage. So let’s dive into the fascinating world of visual imagery.

# Understanding Basic Chart Types I



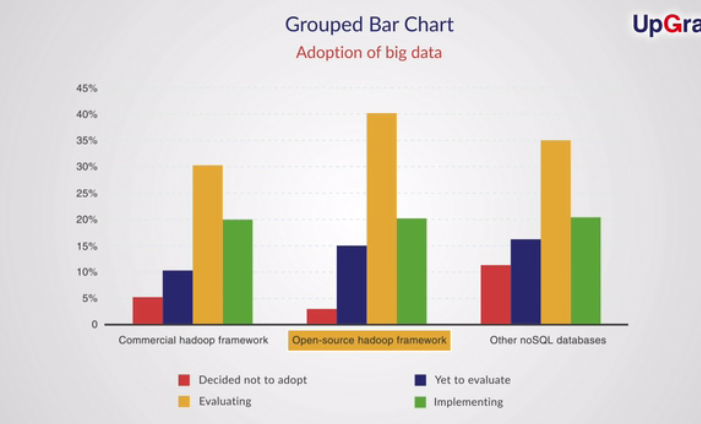
In which of the following cases would a pie chart be ineffective in displaying percentages? Only one option is correct.

**Feedback :**If the number of categories is too high, then it would be difficult to represent all of them on a pie chart, since the area of representation remains the same, while the number of slices of the pie chart increases.

You saw that pre-attentive attributes make graphics easy to understand whereas attentive attributes are relatively difficult to grasp.

You also learnt about line charts, bar charts, histograms, pie charts, stack bar charts, and also understood when to use them and what their limitations are. While a line chart can be used to present a time-dependent trend, bar graphs and histograms are best used for categorical and continuous data respectively. A pie chart best summarises the share of different components in an aggregate whole, while a stacked bar chart is used to compare the share and contribution of categories across different sectors.

So now, you have some basic understanding of when to use this type of graph. You can refer to some of the pointers regarding pie charts in the Additional References section below.



Which of the following plots can be used to show the relation between two quantitative variables?

1. Box plot

2. Line plot

3. Scatter plot

4. Histogram



**2 and 3**

**Feedback :***Line plot and Scatter plot essentially plot the x-y relationship between the two quantitative variables. They show how one variable changes with the other quantitative variable.*

# Summary: Data Visualisation

This session was an introduction to the world of data visualisations. Let's summarise all that you learnt in this session.

You started with understanding the importance of visualisation for data understanding. Then, we covered the differences between the attentive and pre-attentive attributes, to understand the factors that make a visual stand out among the rest.

You glanced through the different types of graphs and charts:

1. Line chart
2. Bar chart
3. Histogram
4. Pie chart
5. Stacked bar chart
6. Scatter plot
7. Box plot
8. Grouped bar chart

You covered the main areas of application and the limitations of all these chart types. You also saw an industry demonstration of how data visualisation can help you easily understand complex and large data.

# Introduction: Visualisation in R - Using the Base Package

Welcome to the session on 'Visualisation in R - Using the Base Package'. In the previous session, you learnt about the importance of visualisation for data understanding and data presentation.

## In this session

The previous session helped you understand the features, utilities, and limitations of the different types of graphs and charts. Now, it's time to learn how to create some of these interesting graphics using R functions.

We will start by looking at some of the inbuilt functions in R for plotting, such as plot(), hist(), boxplot(), etc. We will also be using the inbuilt data set in R, called the "iris" data set and the "mtcars" data set.

However, as you already know, there are many limited functionalities offered by these inbuilt functions in R. Thus, we will move on to the ggplot2 library, which is used to create high-quality graphics in R. You have already learnt how to install library packages in R. You can use the command **install.packages("ggplot2")**to install this high-utility library before proceeding.

# Basic Plotting in R

In the previous session, you learnt about the various types of visualisations and how important they are in the field of data analytics. Now, it is time to get your hands dirty and build some really cool plots.

For starters, let's use the inbuilt R function called plot(). For this, you will be using one of the inbuilt data sets in R – the "iris" data set.

**Scatter Plot**

In the Sepal Length versus Sepal Width plot, the plot() function mapped three colours automatically to represent the three species. What could have caused the plot command to decide that?



**Species is a factor type column having three discrete values**

**Feedback :***The plot command automatically plots all the different factor type variables using different ways of representation based on the command used. The point is that it should be a factor a type, else R wouldn't know how many colours to use etc.*

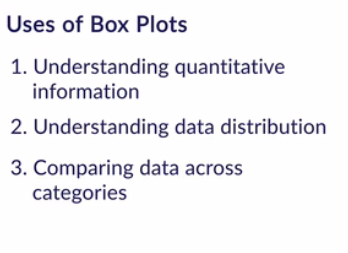
Those were some great warm-up exercises, making you ready for the world of visualisations. You created scatter plots for sepal width vs sepal length using the plot() command. You also learnt how to use the points() command to add more information in the graph. You finally saw that you can add a third dimension of information on a 2-D plot by passing some additional attributes such as "species" and representing them using colour.

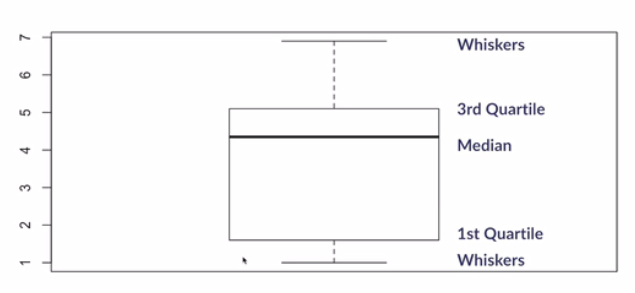
Based on the additional reading above, answer the following: What does this code do?

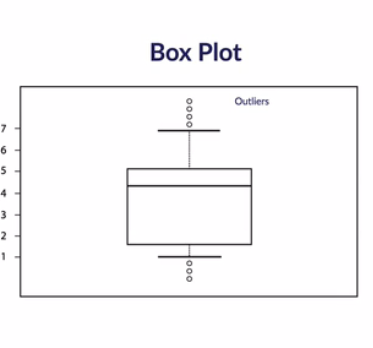
par(mfrow = c(1,2))

plot(iris$Petal.Length, iris$Petal.Width, col = iris$Species, main = "iris", xlab = "Petal Length", ylab = "Petal Width")

plot(iris$Sepal.Length, iris$Sepal.Width, col = iris$Species, main = "iris", xlab = "Sepal Length", ylab = "Sepal Width")







You saw how histograms can be plotted using the hist() command. Then, you moved on to box plots. Box plots are extremely useful to understand the **data distribution** or the **spread** of data.

You also became familiar with the different limitations of the inbuilt graphical functions in R such as:

* Plots are not updated automatically with the data
* Plots are images, like jpeg, not objects; so it is hard to manipulate them after plotting

In the next segment, you will learn about the library package **ggplot2**, which will provide you with many new options and functionalities.

Load the "InsectSprays" data set in R. This data set shows the number of insects left in a field after it was treated with various insecticides.

Plot a box plot of the data. Which of the following insecticides has the maximum median count of insects left after spraying?



**B**

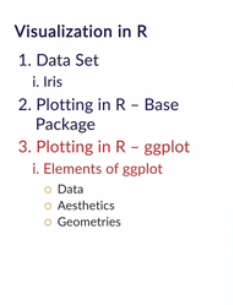
**Feedback :***Use the command: boxplot(InsectSprays$count~InsectSprays$spray). This will plot the graph according to the spray type used. The box plot can tell you at a glance the performance of different sprays.*

For the iris data set, choose the right answer:

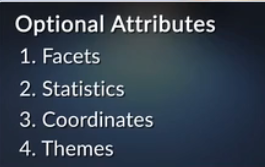


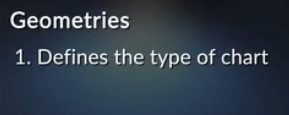
**None of the above are correct**

**Feedback :***Histogram can be plotted with just one continuous variable which is plotted on the x-axis, while its frequency is plotted on the y-axis*



Additional attributes of ggplot 2





That was an introduction to the basic structure of ggplots. The **essential** elements of ggplot are **data, aesthetics and geometries:**

* **Data** provides the values or the information to be visualised
* **Aesthetics** are **properties of graphs** that can be changed as per your requirements
* **Geometries** define the **type** **of chart** that may be used

You learnt the basic structure to create point or scatter plots in ggplot:

**ggplot(dataset, aes(x = \_\_\_, y = \_\_\_)) + geom\_point()**

First, start by identifying the **data** set, which in this case is "mtcars". Then, mention the **aesthetic**element, which essentially defines what should come on the x and y axes. Finally, using the **geometry** layer, to specify what the graph should look like, i.e. a point / scatter plot in this case.

You also have the option of making tweaks to this format to represent more information than normally possible on a 2-D plane. In our case, we defined “cyl” as a factor variable because of its categorical nature and used it to add an extra level of information to the graphics.



**It displays two side-by-side graphs of sepal length vs sepal width and petal length vs petal width**

**Feedback :***The par(mfrow) command helps you plot multiple plots at once. You can decide how the graphs should be arranged.*

**ggplot**

In the mtcars data set, the variable am stands for automatic or manual transmission. The command for plotting am (x) versus mpg (y) is ggplot(mtcars, aes(x = \_\_\_,  y = \_\_\_\_  )) + geom\_point()

What are the most appropriate values of x and y respectively?



**factor(am), mpg**

**Feedback :***Since am is a categorical variable, it should be used as factor(am).*

Consider the following ggplot2 command:

ggplot(mtcars, aes(x = wt,y = mpg, col = factor(cyl))) + geom\_point()

In the above command, what would be the effect of replacing factor(cyl) by just cyl?



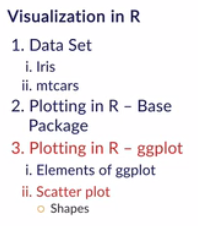
**The values of cyl would be shown using varying shades of a single colour**

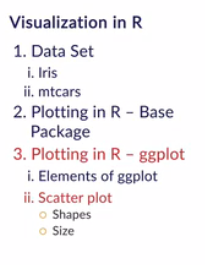
**Feedback :***cyl is interpreted as a numeric variable. Different shades of the same colour are used to represent the numeric value of the cyl variable.*

# Scatter Plots in ggplot

Scatter plots are commonly used to show the relationship between **quantitative or numeric variables.**

Though scatter plots usually display two numeric variables (on the x and y-axes), you can also represent more than two variables in it - even categorical ones. Let's learn how to do that.





**ggplot**

Suppose you want to plot quantitative variables mpg, wt, and hp on a scatter plot with wt on the x axis and mpg on the y axis. What options of aesthetics in ggplot2 are available to plot hp in the same plot?



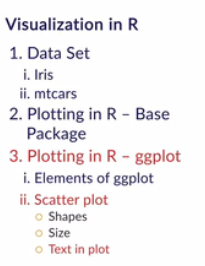
**Both size and colour**

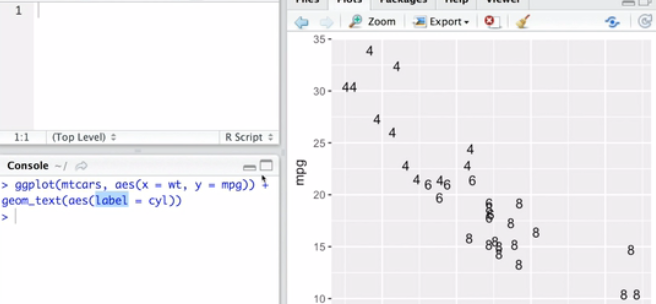
**Feedback :***Both colour and size can be used to represent hp. They might also be used together if needed.*

To summarise, you now know the following things about a ggplot command:

* Aesthetics make it possible to display more than two variables in a plot
* You can display **more than two variables** in a scatter plot using **aesthetics such as size, col** etc.

Apart from aesthetics such as size, col etc., there are many more such as **text, transparency etc.** Now, let's see how text can be used for visualising data.





So, using the geom\_text() command, you saw how to use text instead of graphical shapes to convey some additional information. This will come in useful when want the plot to display the actual values of the data (in any type of plot such as bar, scatter, histogram etc.)

You also learnt that it is often combinations of geometries and aesthetics that produce the best result.  Moreover, creating great visuals is as much an art as it is science. Thus, there is a lot of imagination and mental visualisation involved before the coding actually begins.

Consider the plot of mpg vs wt on the y and x axes respectively, the size mapped to disp (engine displacement) and the colour mapped to hp (horsepower). For questions 1, 2, and 3, mark True or False.

# Plotting Larger Data Sets

In the previous session, you learnt how to visualise small data sets such as "iris" and "mtcars". Now, let's learn some techniques to help you visualise large data sets effectively. Let's use the "diamonds" data set to learn some new concepts.

You will first plot a basic scatter plot and then we'll discuss various ways for manipulating the plot such as adding a line to a scatter plot or changing the colour, position, transparency,  etc.

Heavier cars have lower mileage.

Top of Form



**True**

**Feedback :***You can see from the graph that the points are in a downward trend — as wt increases, mpg decreases.*

Bottom of Form

Consider the plot of mpg vs wt on the y and x axes respectively, the size mapped to disp (engine displacement) and the colour mapped to hp (horsepower). For questions 1, 2, and 3, mark True or False.

Heavier cars have higher horsepower.

Top of Form



**True**

**Feedback :***You can see that as the wt increases, the colour changes from that associated with smaller horsepower to one representing higher horsepower.*

Bottom of Form

Consider the plot of mpg vs wt on the y and x axes respectively, the size mapped to disp (engine displacement) and the colour mapped to hp (horsepower). For questions 1, 2, and 3, mark True or False.

Low horsepower corresponds to low values of engine displacement.

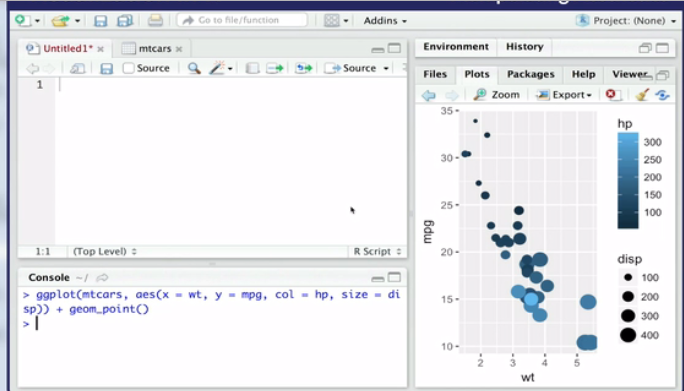
Top of Form

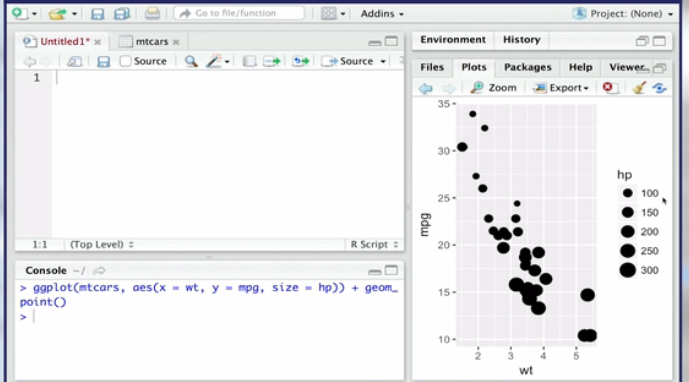


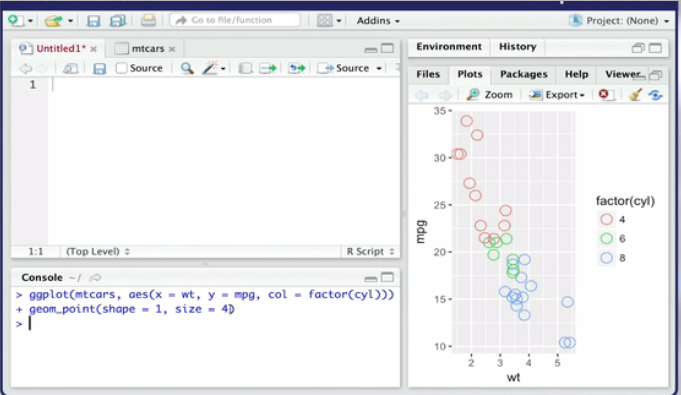
**True**

**Feedback :***You can see from the plot that larger circles are associated with higher hp colours while smaller circles are associated with lower hp colours.*

Bottom of Form





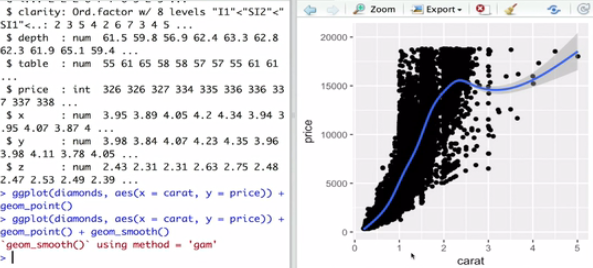


# Plotting Larger Data Sets

In the previous session, you learnt how to visualise small data sets such as "iris" and "mtcars". Now, let's learn some techniques to help you visualise large data sets effectively. Let's use the "diamonds" data set to learn some new concepts.

You will first plot a basic scatter plot and then we'll discuss various ways for manipulating the plot such as adding a line to a scatter plot or changing the colour, position, transparency,  etc.

**Download the R practice file attached below and code along as you watch the lecture.**



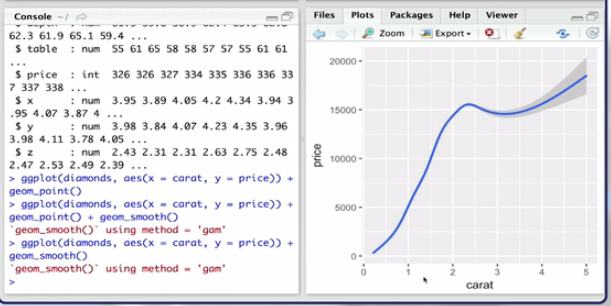
**Large Data Sets**

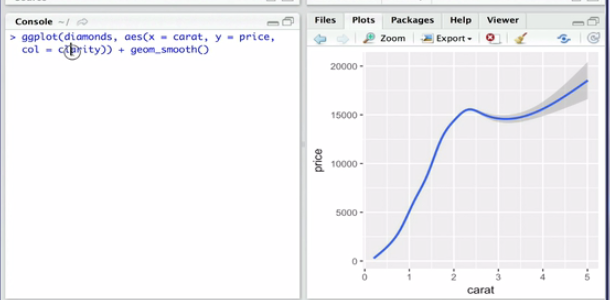
Between carat values 1 and 2, there is a completely black region. Why are some parts of the plot totally black?

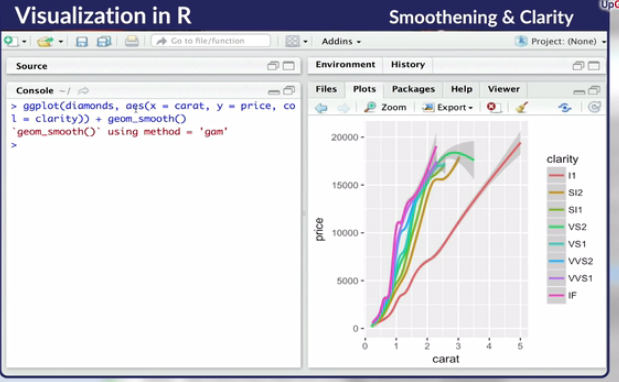


**The number of observations are extremely high in that region, resulting in the overlapping of points**

**Feedback :***Since there are a large number of points in that region, they are very close together and overlapping other points*







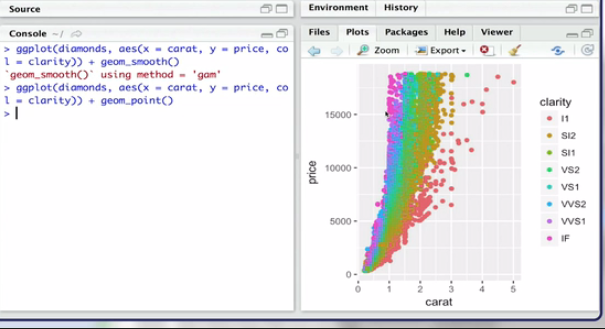
**Large Data Sets**

We removed the geom\_point() layer and kept only the geom\_smooth() layer. Which piece of information is NOT lost when you remove the points and keep only the smooth line?



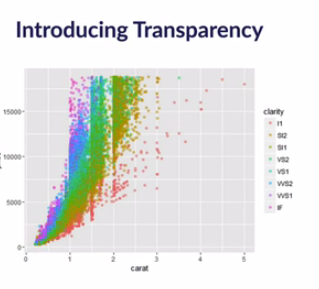
**The trend between price and carat value of diamonds**

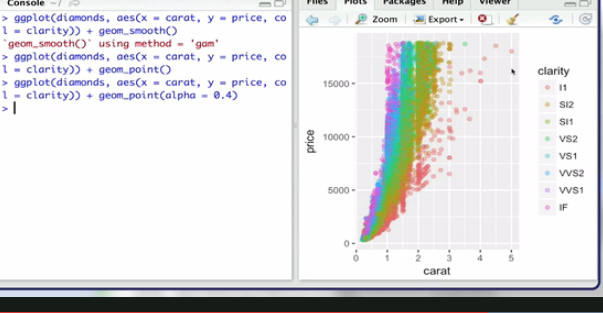
**Feedback :***The lines show the trend. You cannot infer the other information from just the lines.*



When points in a scatter plots are cluttered, a smooth line can be used to visualise the underlying trend or relationship between variables on the x and y axes.  It is important to keep in mind that removing the number of points and replacing them by just a line is not always a good idea since you lose a lot of information.

You looked at how colour can be used to represent another dimension of the data. But, if you are dealing with a large number of data points, the points overlap and hence not all the points can be shown. Let's see a way to counter this.

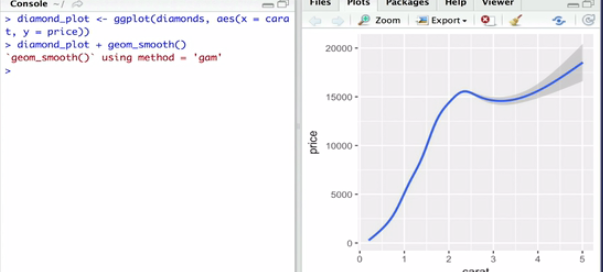




Colour can be used to visualise data spread across various categories effectively, and transparency can be used to visualise data points shadowed by other data points.

**Visual as Objects**

You saw how you can overcome limitations of the base package using the ggplot package, such as being able to plot only a limited number of points. The ggplot command is a more powerful and improved command compared to the basic plot command in R. Let's understand one such critical limitation of the base package — once a plot is made, it cannot be modified. Let's see how ggplot helps you overcome it.



You learnt that the plots created by the ggplot command can be stored as variables or objects and can be manipulated further, unlike the plots created by the base plotting package. Now let’s move ahead and learn to make some more plots.

s the value of the  alpha parameter**decreases**from 1 to 0, the level of transparency:

(Note: High transparency means that the points are more transparent while lower transparency means that the points are more opaque with less visibility of the points behind.)



**Increases**

**Feedback :***The transparency value increases as alpha decreases.*

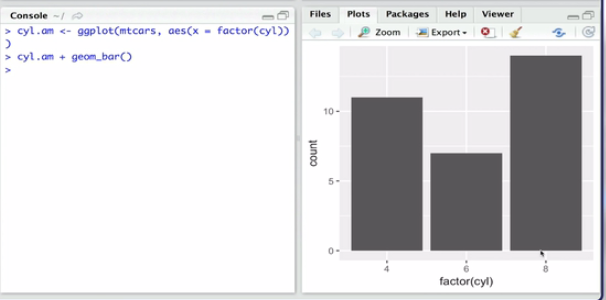
**Bar Charts**

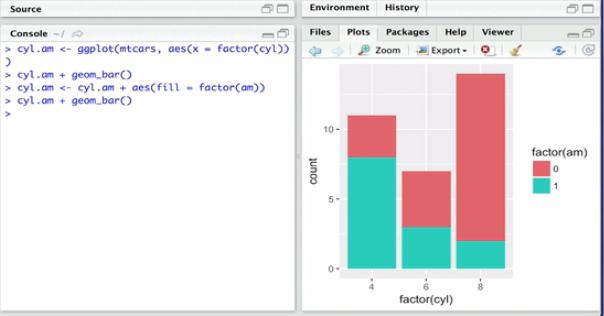
In the bar plot, if you write col = "red" inside geom\_bar, what is the result? Only one option is correct.



**The boundary of all three bars becomes red**

**Feedback :***Col changes the boundary colour, while fill changes the colour inside the bar plots.*





**Bar chart**

We plotted a bar chart with x=factor(cyl) and col representing the am (blue = automatic, pink = manual). What does the length of blue and pink parts of each bar represent?



**The number of cars with am=0 and 1 respectively**

**Feedback :***Col will segregate the number of cars based on the transmission type*

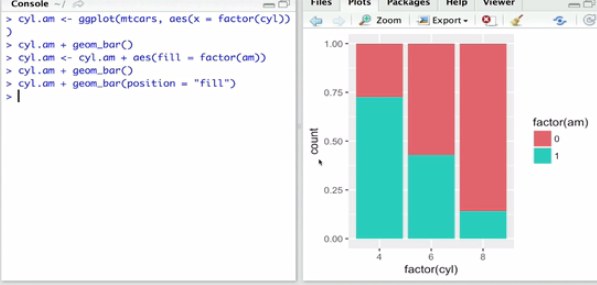
**Bar Chart**

We plotted a bar chart with x=factor(cyl). What is on the y axis?



**Count associated with cylinder value**

**Feedback :***Bar chart shows the count associated with the categorical variable*



You learnt how to plot a basic bar chart using the ggplot package and make stacked and absolute bar charts. It was also interesting to see that the position of the attribute **fill**decides if the chart is absolute or stacked. In the next lecture, you will see how data can be effectively visualised by leveraging aesthetics optimally.

**ggplot**

The default position of geom\_bar() is:

Top of Form



**Stack**

**Feedback :***You can also read the documentation of geom\_bar() to check this.*

Bottom of Form

**ggplot**

The default position of geom\_bar() is:

Top of Form



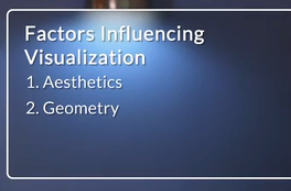
**Stack**

**Feedback :***You can also read the documentation of geom\_bar() to check this.*

Bottom of Form

# Factors Affecting Visualisation

Till now, you have created plots using the ggplot command and the base package. We discussed scatter plots, bar charts, histograms, etc. You learnt how to map aesthetics such as colour and shape to variables, but all aesthetics may not go well with all types of variables. Let's see how appropriate aesthetics can be chosen to map various variable types.

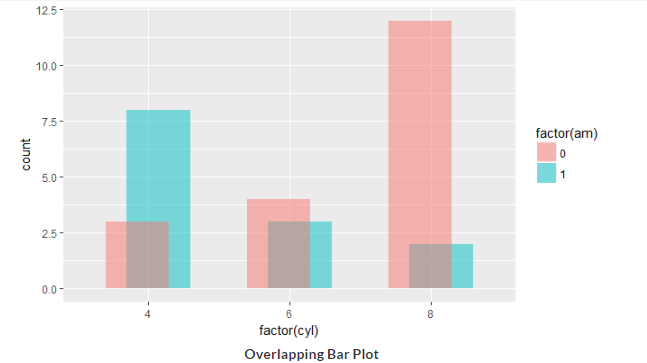


This segment helped you understand how you can leverage various attributes associated with geometry and aesthetics such as shape, colour, etc to visualise data effectively. In the next video, we will discuss a very important aspect of visualisation called **jitter**.

You want to plot the bar chart you drew in the previous segment in a different way. The values of both the factors should appear side by side for all the cylinder values. Write down the correct code.

You can also search some ways to make it look even better by introducing overlap. Here are some hints for you. Write the code for doing this as well.

ggplot(\_\_\_\_, aes(x = \_\_\_\_, fill =\_\_\_\_)) + geom\_bar(alpha = \_\_\_\_, position = position\_dodge(\_\_\_\_))

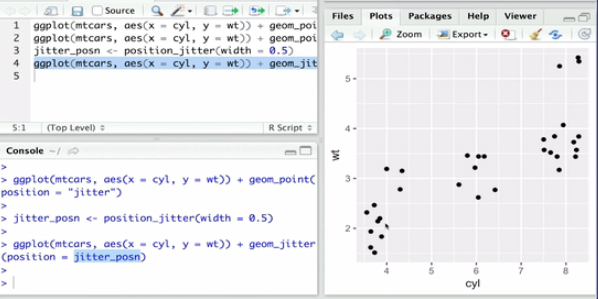


# Jitter

Let's discuss a very interesting concept — jitter. It comes handy when you face the problem of overlapping points, i.e. a point shadowing another point by lying exactly on top of it. Jitter helps you shake the points a little bit from their original location for better visualisation. Let’s learn from Prof. RC what that means.

You learnt that jitter can be added as a **position** inside the **geometry**layer. It is used for better visualisation of **categorical** data and often helps solve the problem of overplotting.

Also, jitter can be plotted using geom\_jitter(), which is equivalent to geom\_point(position = "jitter").



Consider the diamonds data frame. You want to display the points in the data set and the carat value for each clarity (categorical) level. Further, you want to display the price using shades of a colour. You decide to plot clarity on the x axis and carat on the y axis.

Given this much information, the ggplot object can be created using:

Top of Form



**dia.object <- ggplot(diamonds, aes(x = clarity, y = carat, col = price))**

**Feedback :***The first command will create a ggplot object with the required features. The other two will give errors. If you want to plot a graph using the ggplot object then use dia.object + some geometry type.*

Bottom of Form

**Diamonds data**

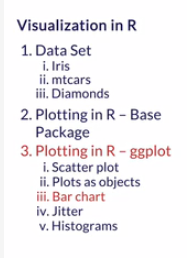
Consider the diamonds data frame. You want to display the points in the data set and the carat value for each clarity (categorical) level. Further, you want to display the price using shades of a colour. You decide to plot clarity on the x axis and carat on the y axis.

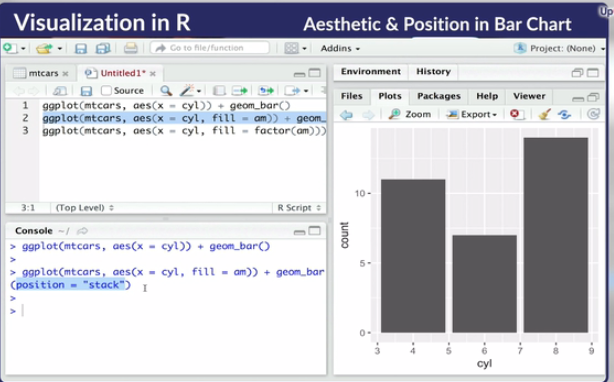
Using the dia.object created in the previous question, what should be the command for plotting all the points for each clarity level?

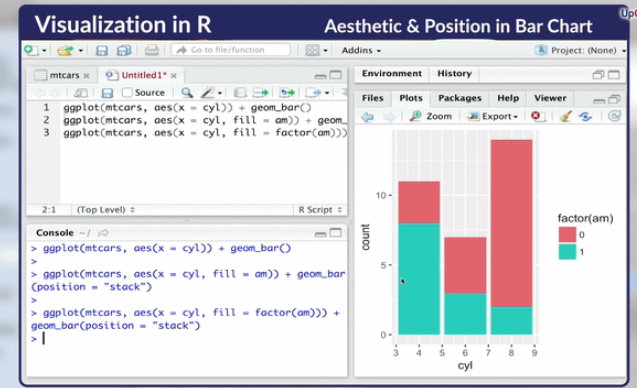


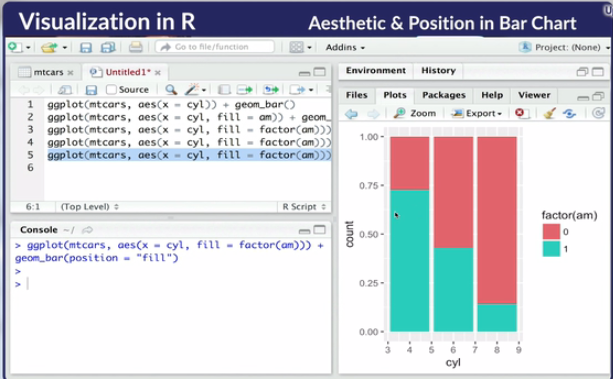
**dia.object + geom\_point()**

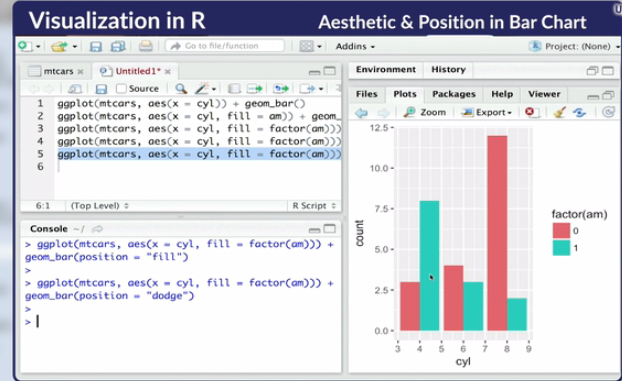
**Feedback :***A scatter plot will show the values of carat for each clarity value. A bar chart shows the aggregate value such as count, sum, etc and not the individual values. Scatter plot is made using geom\_point()*

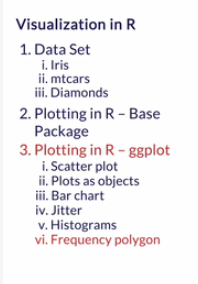


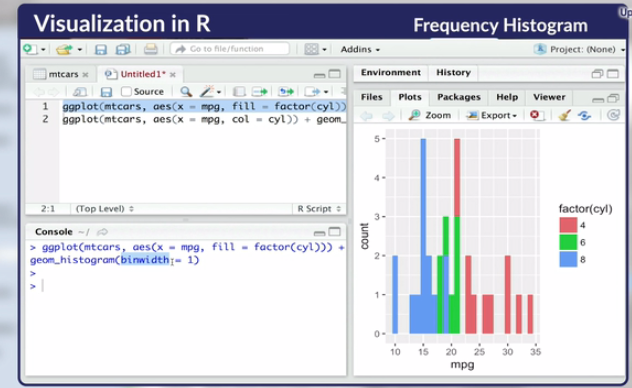


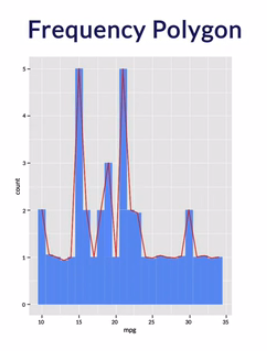


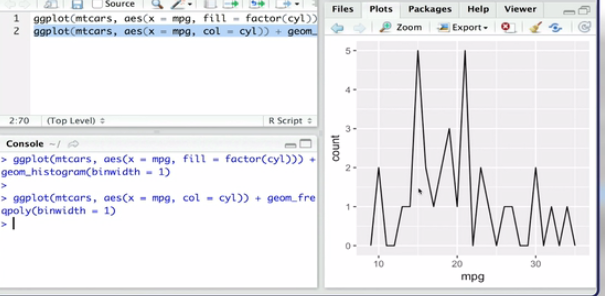






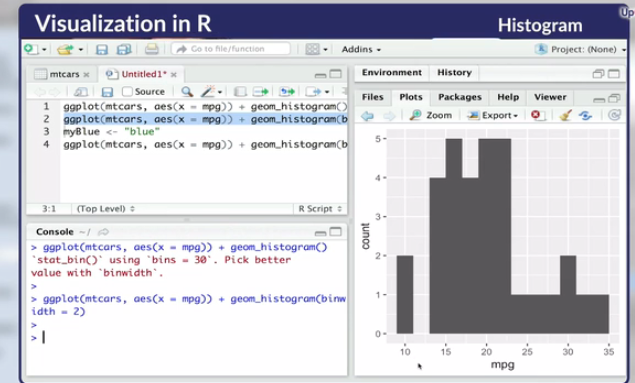


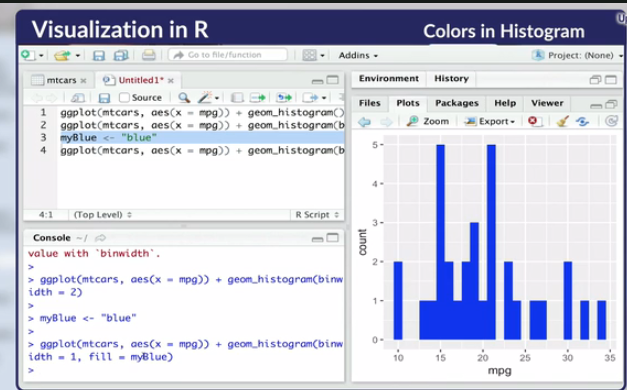




You learnt that the default bin width for a histogram using the ggplot command is one and that you can manipulate the bin width as per your requirement. Also, introducing the**fill** attribute helps you introduce colour to the plot.

Now, let's quickly revisit bar charts and learn a few more manipulations that can be done on bar charts for better visualisation and understanding of data. Let's also revisit the frequency histogram, but the position of the fill parameter will not be the same this time.







**The y-axis of a histogram usually represents the count (or frequency)**

**Feedback :**A frequency histogram shows the frequency or the number of times the data is in a given range.



**Only one variable (column) is needed to plot a histogram**

**Feedback :**Only one variable is needed. Histogram plots the frequency of the values of the variable in various numeric bins.

The minimum number of variables required to plot a frequency polygon is:

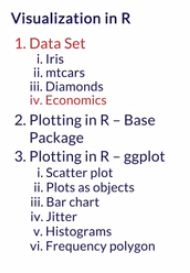
Top of Form

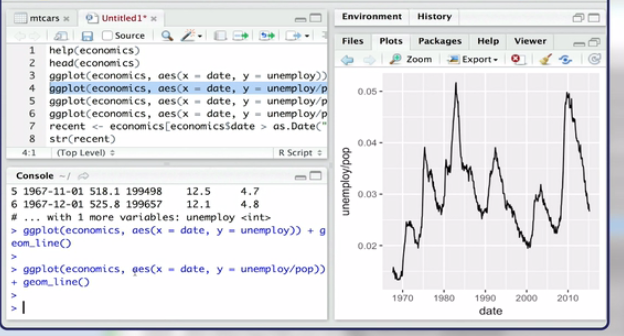


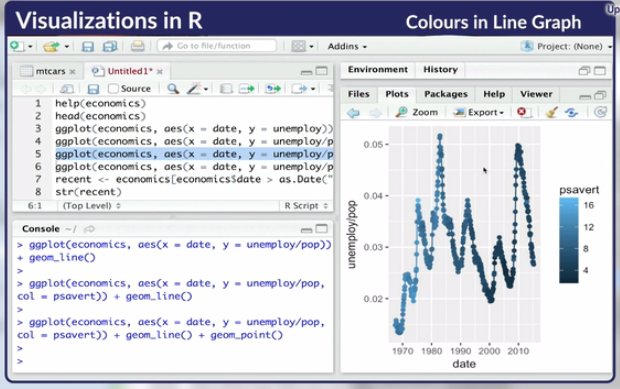
**1**

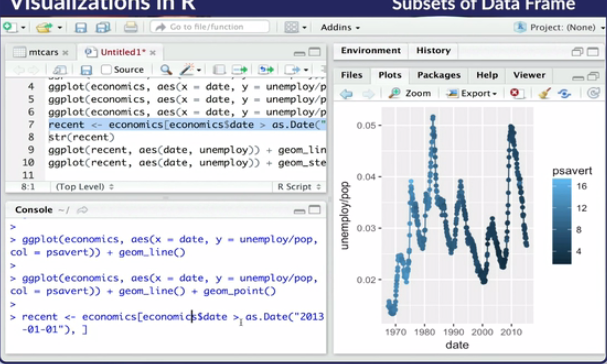
**Feedback :***The x-axis needs one continuous variable, while the y-axis represents the frequency.*

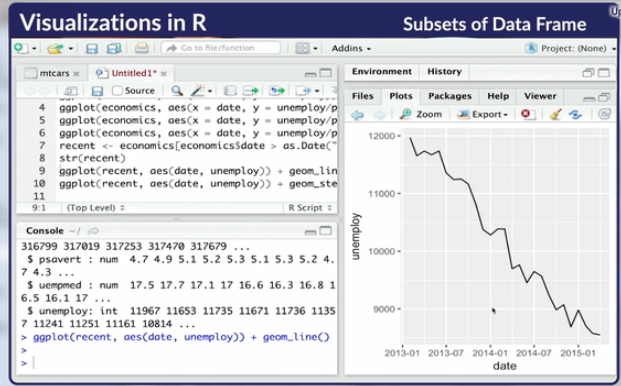
Bottom of Form

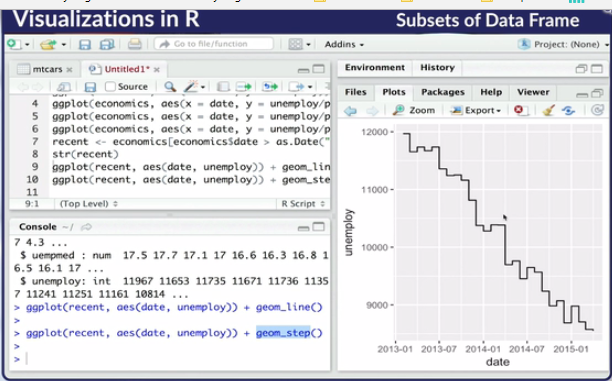












You learnt that a line graph can be used to visualise time series data effectively. You can also manipulate the variables to be plotted on a line graph; for example, you can divide or multiply two variables and plot that using the line graph. You then learnt how to add additional attributes such as colour and additional points on the line graph.

Furthermore, step plots can be used to analyse the spikes in a line plot.

**Part II: Time Series Data**

Now you will work on a new data set called **Seatbelts**. It is an inbuilt data set in R. Use the following commands to convert this data set into a data frame which you can use for plotting ggplot graphs. The commands are given below:

install.packages("zoo")

library(zoo)

ukacc <- data.frame(Seatbelts, date = as.Date(as.yearmon((time(Seatbelts)))))

This data is about deaths due to road accidents in the UK. The various parameters are:

* DriversKilled: Number of car drivers killed
* drivers: Number of car drivers killed or seriously injured
* front: Number of front-seat passengers killed or seriously injured
* rear: Number of back-seat passengers killed or seriously injured
* kms: Distance travelled
* PetrolPrice: the price of petrol
* VanKilled: Number of van drivers killed
* law: Factor showing whether the law making seat belts compulsory was in effect or not
* **Seat Belts - Data Exploration**
* What are the factors that can affect the number of driver deaths?
* 
* **All three**
* **Feedback :***kms shows the number of kms travelled - more kms means more chances of getting an accident. Petrol price might affect the number of kilometers travelled while the seat belt law will decrease the number of deaths resulting from the accidents*

Plot a graph of drivers on the y-axis and date on the x-axis. What is your observation? Choose the best answer among the following options. 

**There is little trend variation in graph, however there is a prominent periodic variation**

**Feedback :***Use the following command: ggplot(ukacc,aes(x = ukacc$date, y = ukacc$drivers)) + geom\_line(). You can appreciate how important it is to use the correct visualisation for the data type. Plot the same data using geom\_point and geom\_smooth. Also use these commands: ggplot(ukacc,aes(x = ukacc$date, y = ukacc$drivers)) + geom\_line() + geom\_smooth(span = 0.1) and ggplot(ukacc,aes(x = ukacc$date, y = ukacc$drivers)) + geom\_line() + geom\_smooth(span = 0.5)*

You want to compare how other factors such as front and rear passenger deaths, law, petrol price, etc. change with time and how they compare to driver deaths and injuries. Remember one thing though — just because they are changing similarly, doesn't mean that they are the cause. Correlation does not mean causation.

A good way to study time series is to plot them on the same plot or against each other and look for similarities in the time series. Plot drivers, front, and rear on the same plot. See if the scales of each series are the same (this would mean that the numbers you plot are similar, rather than one series having numbers ranging from 0 - 10 and another one having numbers from 1000-2000). What conclusion can you draw from it?

Top of Form



**All 3 have similar pattern in time**

**Feedback :***Use the command: ggplot(ukacc,aes(x = ukacc$date, y = ukacc$drivers), col = "black", size = 1) + geom\_line() + geom\_line(data = ukacc,aes(x = ukacc$date, y = ukacc$front), col = "red", size = 1) + geom\_line(data = ukacc,aes(x = ukacc$date, y = ukacc$rear), col = "blue", size = 1). You will see that all of them are having similar periodic oscillations. It could be because the number of accidents is seasonal in nature and that number of serious accidents affects all 3 of them*

Bottom of Form

**Seat Belts - Deriving Inferences**

One way to show whether the law was in effect or not is to insert a vertical line. Which do you observe once you do it?

Complete and use the following command:

geom\_vline(xintercept = as.numeric((\_\_\_\_\_\_\_)), linetype=\_\_\_\_\_\_, size = \_\_\_\_\_, alpha = \_\_\_\_\_)

You can also use [this](http://stackoverflow.com/questions/5388832/how-to-get-a-vertical-geom-vline-to-an-x-axis-of-class-date) link for reference.

What is your observation?

Top of Form



**There is a sharp decline in the number of serious injuries and deaths to drivers after this law was enacted**

**Feedback :**Looking at the data, there is a sharp decline in the number of serious driver injuries and deaths, anf this is also true for front row passengers. It makes sense too, since looking at the history of legislation, we see that this law only mandated front seat belts, whereas the law concerning back seat belts was implemented later. Hence, the rear value doesn't change much.

Bottom of Form

Extract new columns 'month' and 'year' from date. Aggregate on the months and identify the month when the average number of driver deaths and serious injuries the highest. (Hint: Separate the date into year, month and day using tidyr)



**December**

**Feedback :***First use tidyr to separate the date. Use this command ukacc <- separate(ukacc, date, c("year","month","date"),sep = "-",remove=TRUE). Then use the following commands t group the driver deaths by month: drivers\_month = ukacc %>% group\_by(month) %>% summarise(driverdeaths=mean(drivers)). You will have to load the dplyr package for using this command and ggplot(drivers\_month, aes(x=month, y=driverdeaths)) + geom\_bar(stat = "identity") to plot the graphs.*



**The highest average is for 1973**

**Feedback :***drivers\_year = ukacc %>% group\_by(year) %>% summarise(driverdeaths=mean(drivers)) and then plot a ggplot using ggplot(drivers\_year, aes(x=year, y=driverdeaths)) + geom\_bar(stat = "identity").*

Now, plot a box plot using yearly data. What do you observe?



**The median number of deaths and serious accidents for drivers are minimum for 1983**

**Feedback :***Use the following command to plot the box plot: ggplot(ukacc, aes(group = year,x = date, y=ukacc$drivers)) + geom\_boxplot()*

Can you conclude that making seat belts compulsory helped in reducing the number of deaths and serious injuries?



**Not clear with just the visualisations.**

**Feedback :***The definitely yes and definitely no questions don't make sense. In solving things, real life situations may not be so clear. Rather than saying yes or no, we can say that I think/ probably/ am confident that there is an effect. In the later sessions, you will learn how to find out exactly how much confident you are*

# Summary: Visualisation in R - Using ggplot2

The most important takeaways from this session are as follows:

* There are three essential layers of ggplot - data, aesthetics and geometries
* Aesthetics are used to map variables (x, y , cyl, am, etc.) to attributes (colour, size, etc.) whereas the geometry layer is usually used to assign constant attributes to plots (col = "red", shape = 4, etc.)
* Large data sets often face overplotting problems which can be solved using jitter plots
* Different aesthetics may be better for categorical and continuous variables; this depends heavily on the data set being plotted
* Three common bar positions are 'stack', 'fill' and 'dodge'
* Line plots are used to display time series data and can be combined with geom\_points() for better readability

Plot a bar chart to find out the most common number of wickets Ravindra Jadeja has taken in an innings. What is the number?



**2**

**Feedback :***Use the following command: ggplot(sir, aes(x = Wkts)) + geom\_bar() . Here the bowling figures data has been loaded into a data frame called sir.*

In which innings is Jadeja's median number of wickets taken the highest? (Innings means the innings of a test match — first, second, third or fourth.)



**3**

**Feedback :***Use the following command: ggplot(sir, aes(x = as.factor(Inns), y = Wkts)) + geom\_boxplot()*

Plot a scatter plot between the number of maiden overs bowled and the economy rate and answer the following:



**The economy rate decreases as the number of maiden overs bowled increases.**

**Feedback :***Use the following command: ggplot(sir, aes(x = as.numeric(Mdns), y = as.numeric(Econ))) + geom\_point()*