

# Hands-on Exercise 1: A Layered Grammar of Graphics: ggplot2 methods

PUBLISHED

January 1, 2024

MODIFIED

January 12, 2024

## Getting Started

Loading R package

Importing Data

Summary Data

The code chunk below uses `p_load()` of pacman package to check if tidyverse packages are installed in the computer. If they are, then they will be launched into R.

[► Code](#)

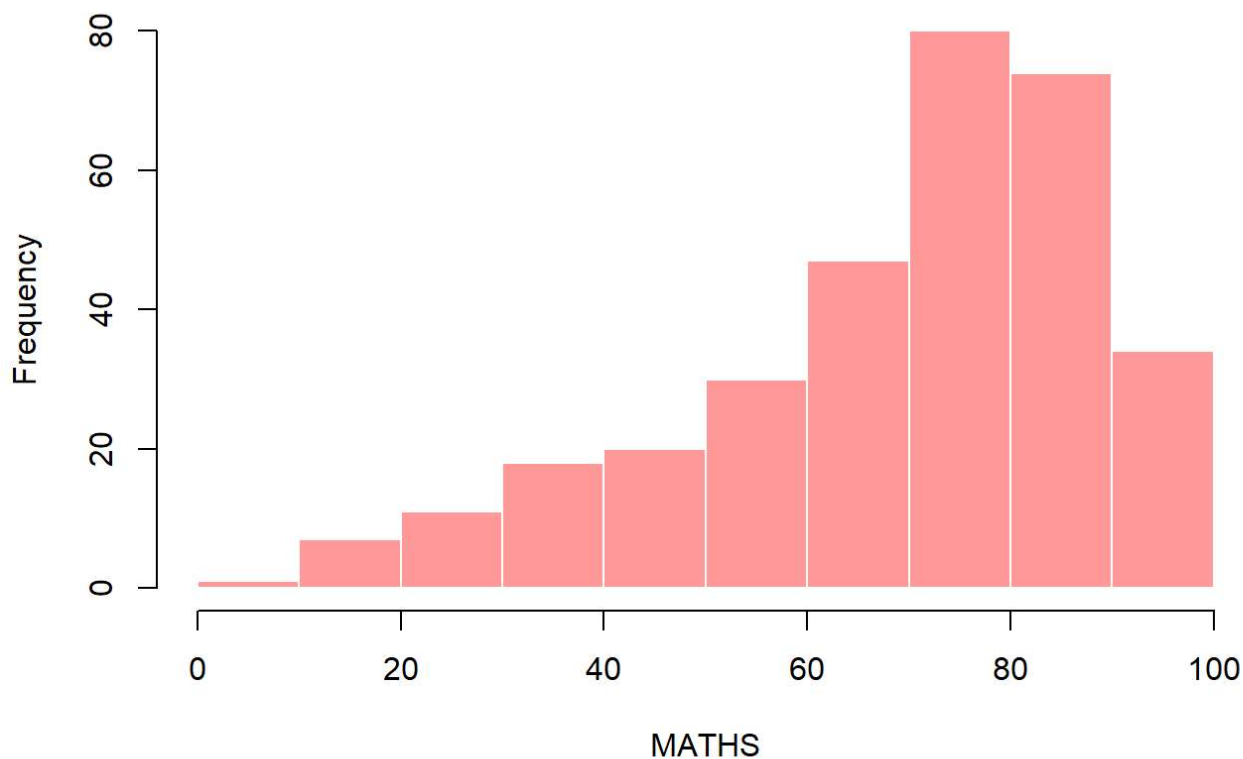
## R Graphics VS ggplot

R Graphics

ggplot2

[► Code](#)

**Distribution of Maths Scores**

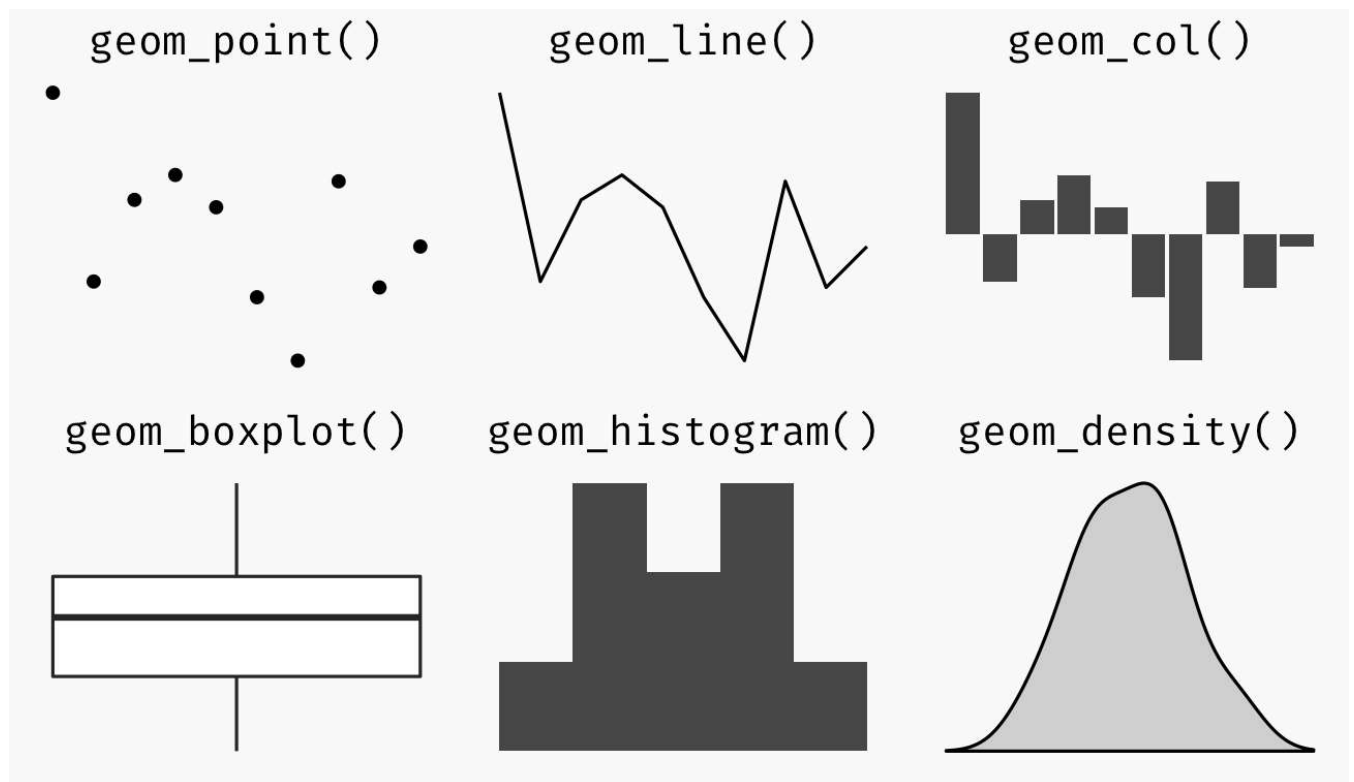


According to [Hadley Wickham](#), the transferable skills from ggplot2 are not the idiosyncrasies of plotting syntax, but a powerful way of thinking about visualisation, as a way of mapping between variables and the visual properties of geometric objects that you can perceive.

## Essential Grammatical Elements in ggplot2: geom

Geometric objects are the actual marks we put on a plot. Examples include:

- `geom_point()` for drawing individual points (e.g., a scatter plot)
- `geom_line()` for drawing lines (e.g., for a line charts)
- `geom_smooth()` for drawing smoothed lines (e.g., for simple trends or approximations)
- `geom_bar()` for drawing bars (e.g., for bar charts)
- `geom_histogram()` for drawing binned values (e.g. a histogram)
- `geom_polygon()` for drawing arbitrary shapes
- `geom_map()` for drawing polygons in the shape of a map! (You can access the data to use for these maps by using the `map_data()` function).

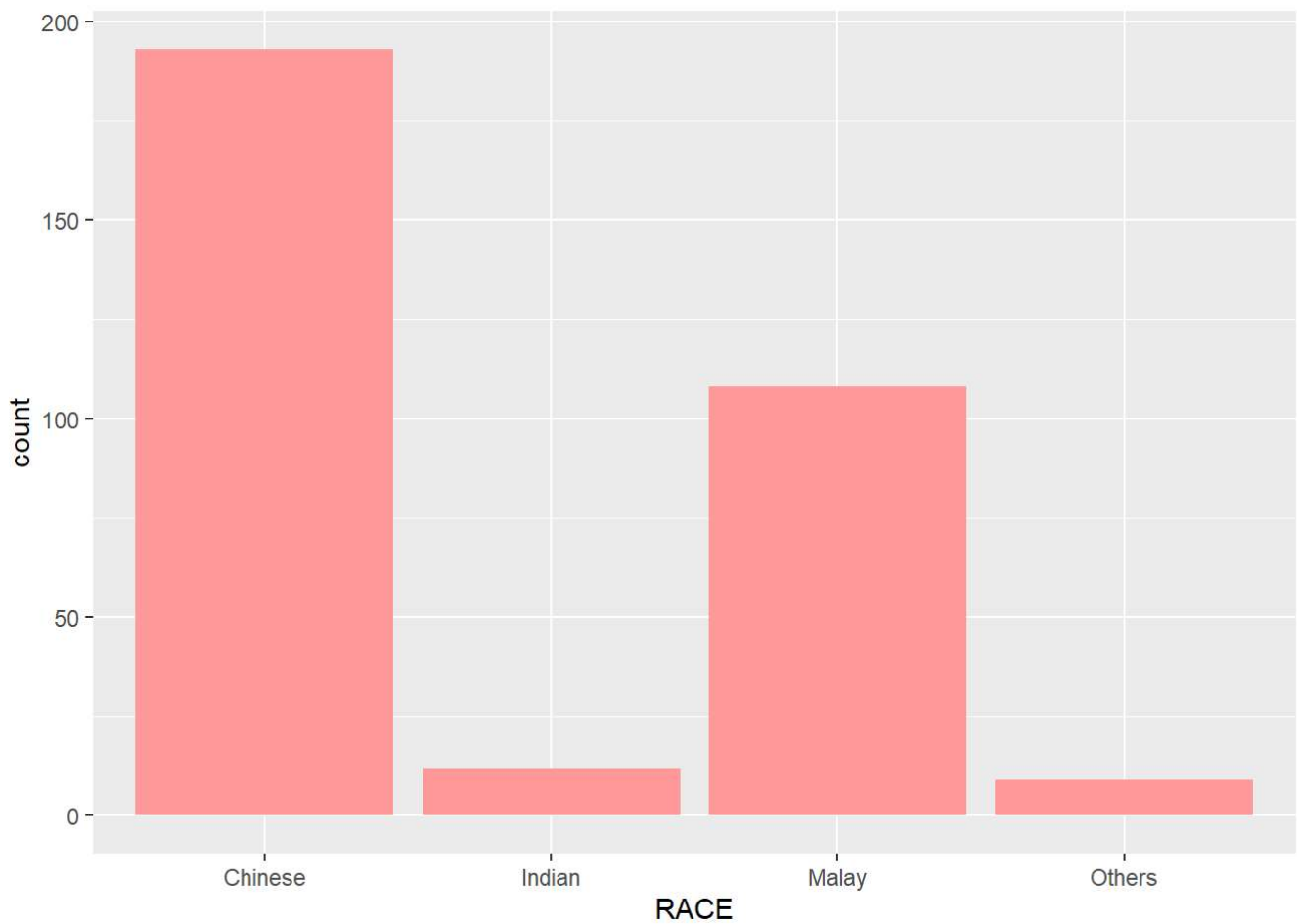


- A plot must have at least one geom; there is no upper limit. You can add a geom to a plot using the + operator.
- For complete list, please refer to [here](#)

## geom\_bar

The code chunk below plots a bar chart by using [geom\\_bar\(.,\)](#).

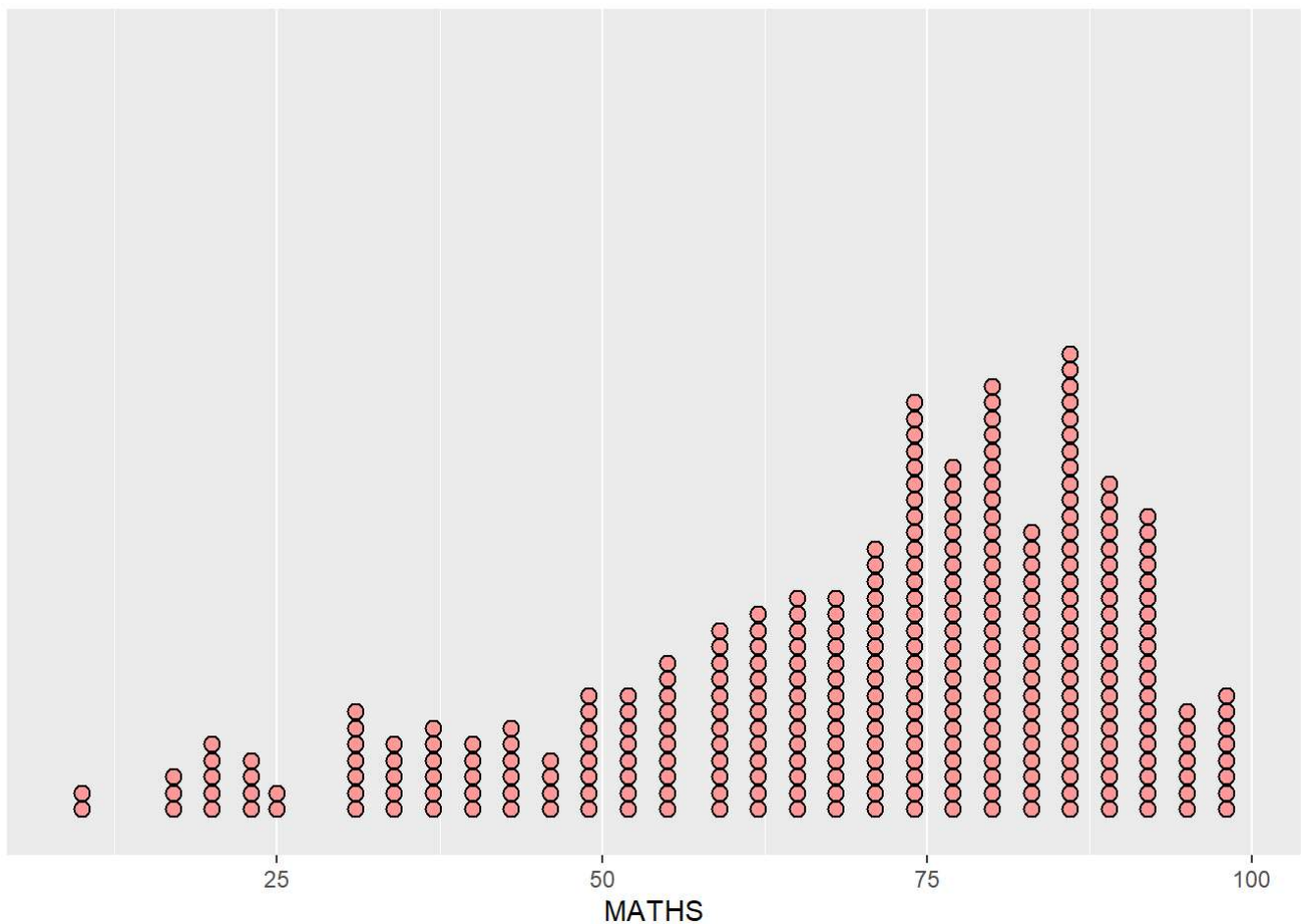
► Code



## geom\_dotplot

In the code chunk below, [geom\\_dotplot\(\)](#) of ggplot2 is used to plot a dot plot.

► Code



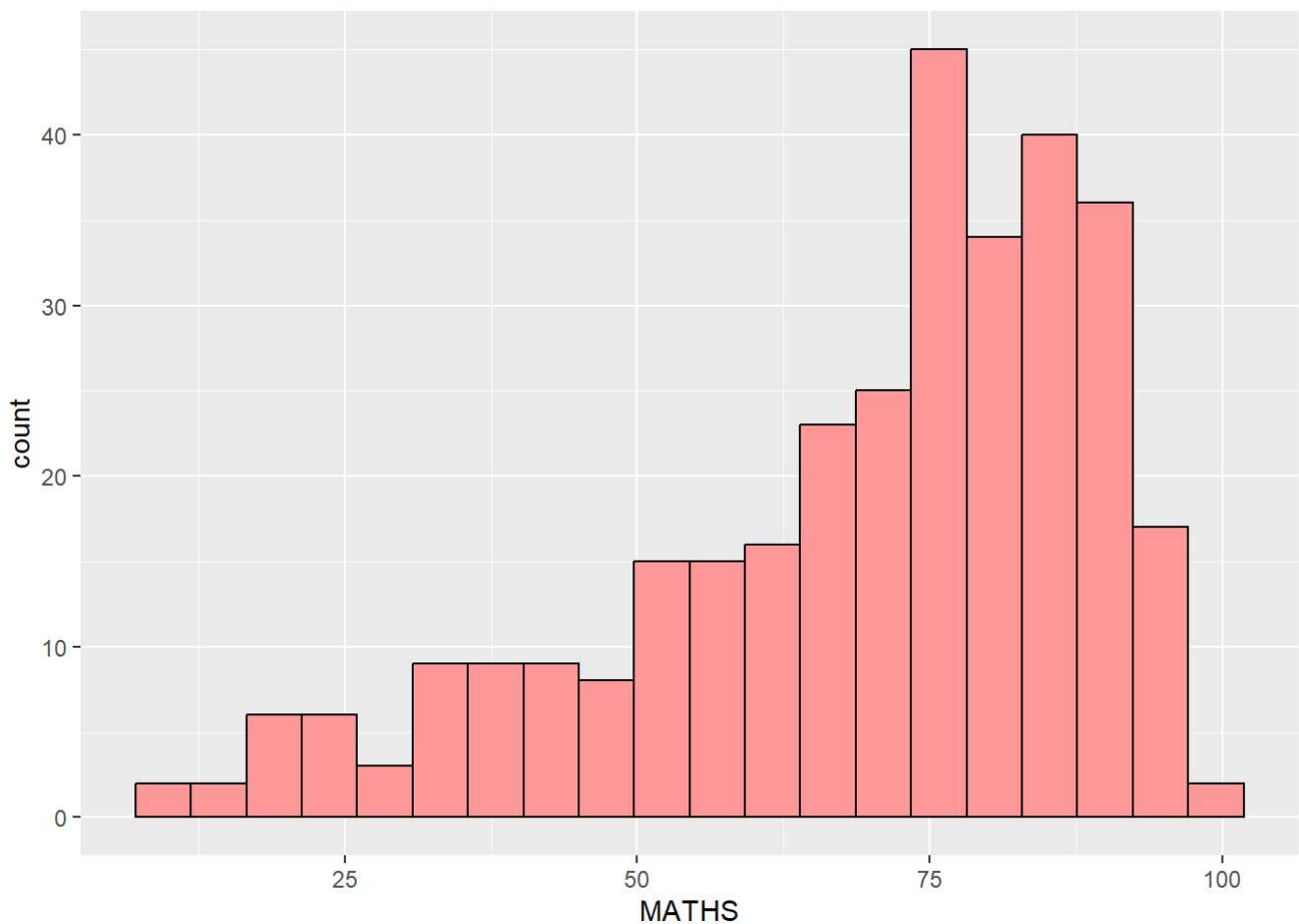
For above `geom_dotplot()`, we use two arguments.

- *binwidth argument* is used to change the binwidth to 2.5., and
- `scale_y_continuous()` is used to turn off the y-axis by setting to NULL.

## geom\_histogram

In the code chunk below, [geom\\_histogram\(\)](#) is used to create a simple histogram by using values in MATHS field of exam\_data.

► Code



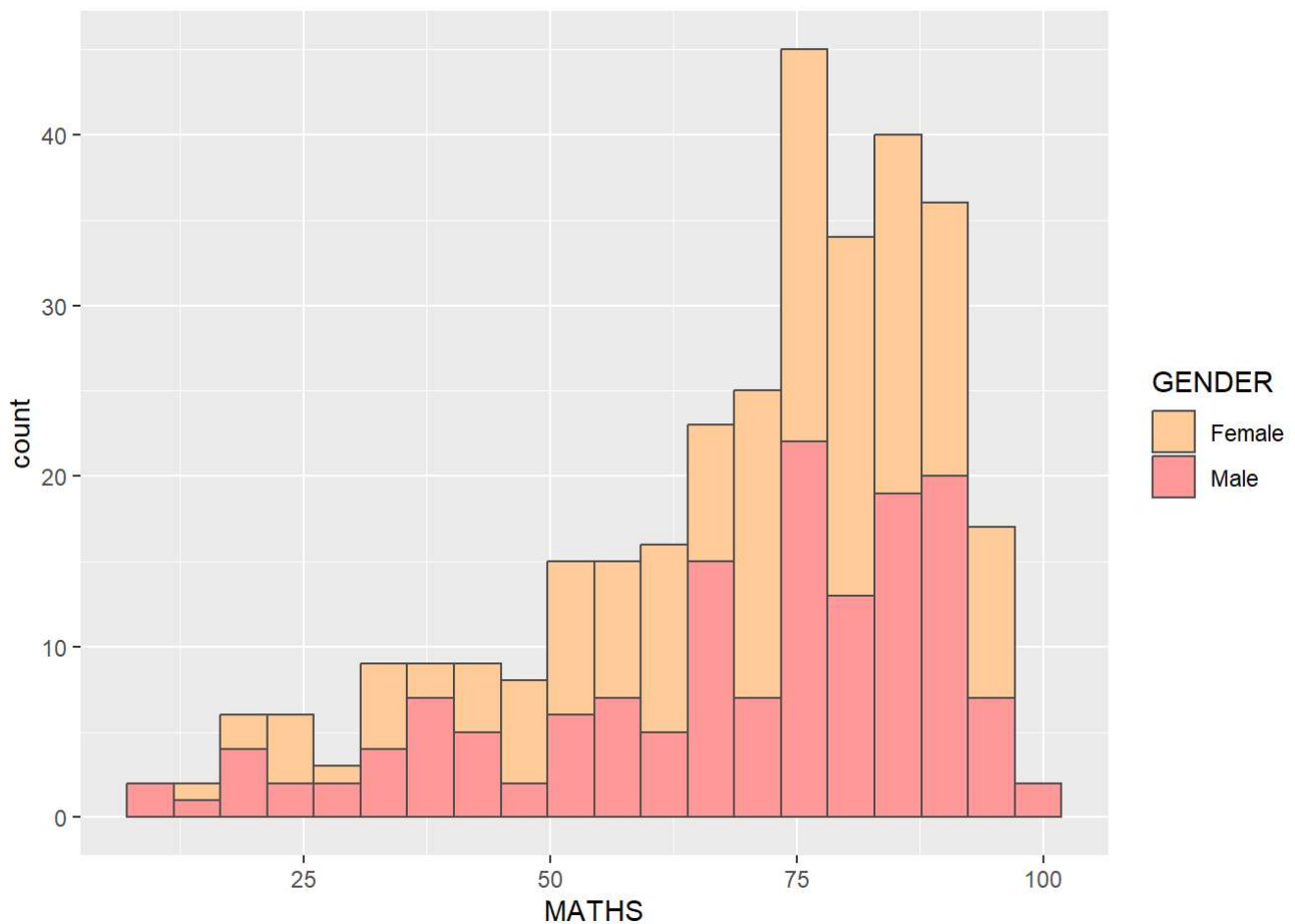
For above `geom_histogram()`, we use three arguments.

- *bins argument* is used to change the number of bins to 20,
- *fill argument* is used to shade the histogram with light pink color, and
- *color argument* is used to change the outline colour of the bars in black.

## Modifying a geometric object by changing aes()

The code chunk below changes the interior colour of the histogram (i.e. fill) by using sub-group of `aesthetic()`.

► Code

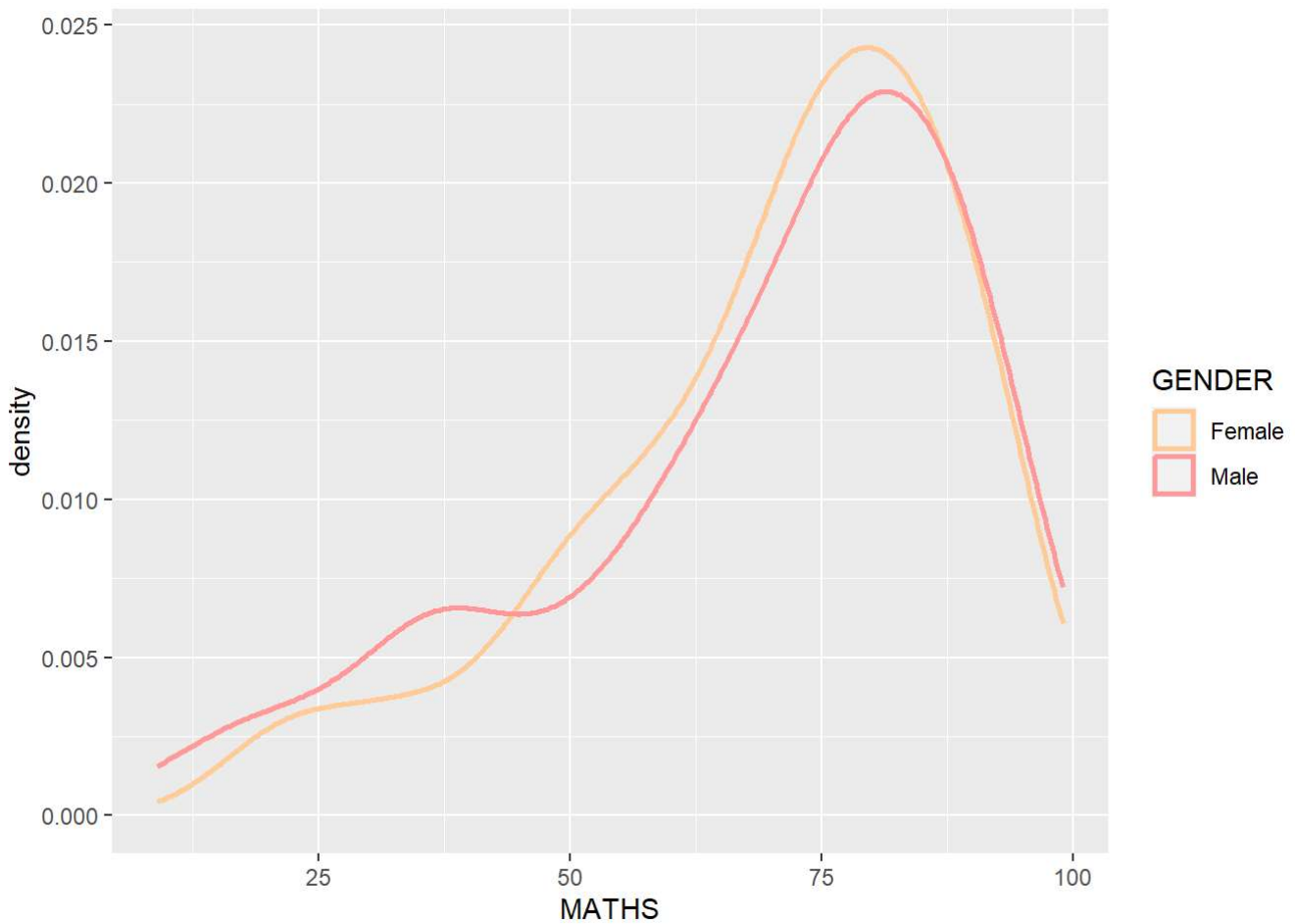


## geom-density

`geom-density()` computes and plots [kernel density estimate](#), which is a smoothed version of the histogram. It is a useful alternative to the histogram for continuous data that comes from an underlying smooth distribution.

The code chunk below plots two kernel density lines by using colour or fill arguments of `aes()`

► Code



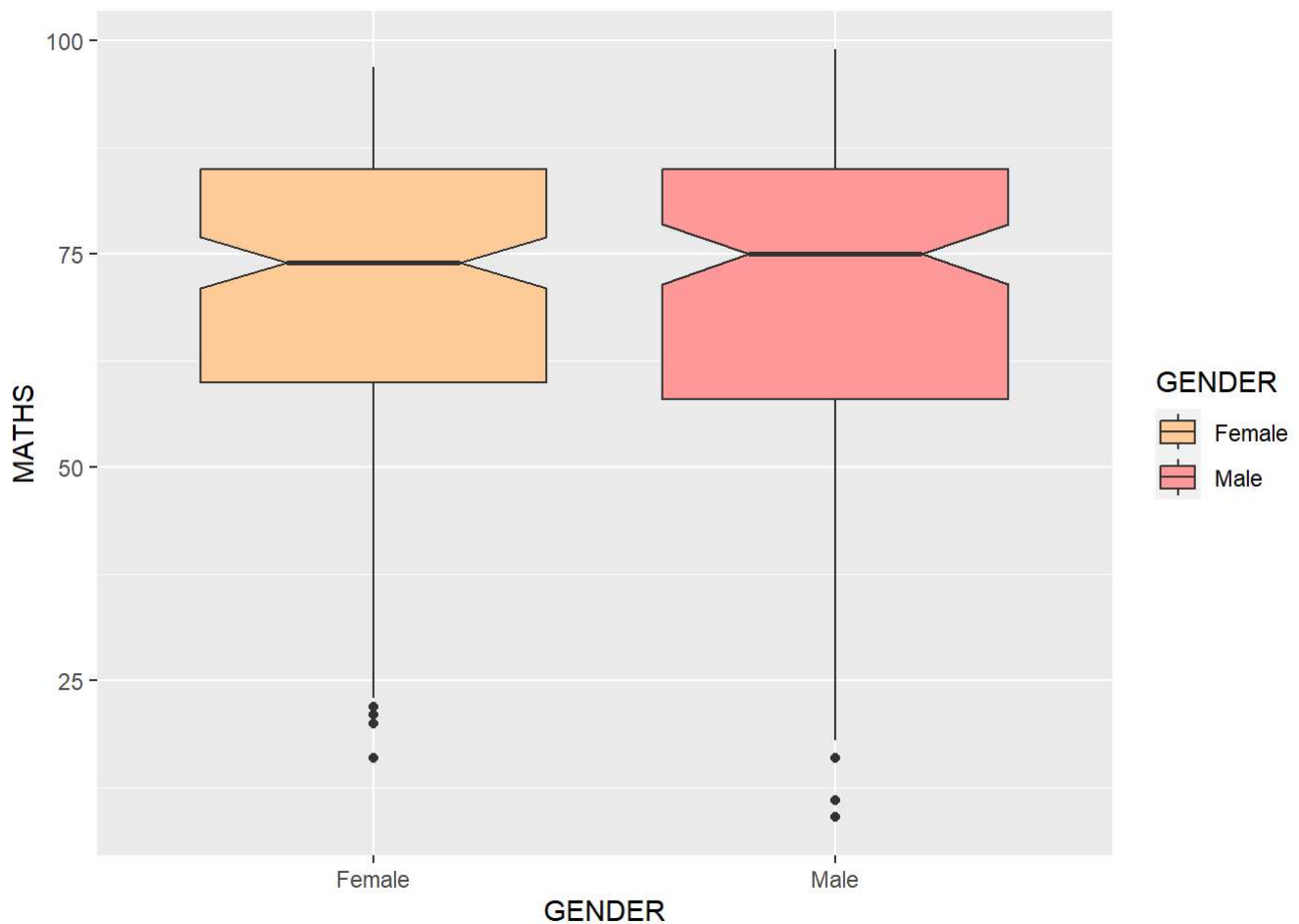
## geom-boxplot

`geom_boxplot()` displays continuous value list. It visualises five summary statistics (the median, two hinges and two whiskers), and all "outlying" points individually.

The code chunk below plots boxplots by using `geom_boxplot()`.

► Code





## geom\_violin

[geom\\_violin](#) is designed for creating violin plot. Violin plots are a way of comparing multiple data distributions. With ordinary density curves, it is difficult to compare more than just a few distributions because the lines visually interfere with each other. With a violin plot, it's easier to compare several distributions since they're placed side by side.

The code below plot the distribution of Maths score by gender in violin plot.

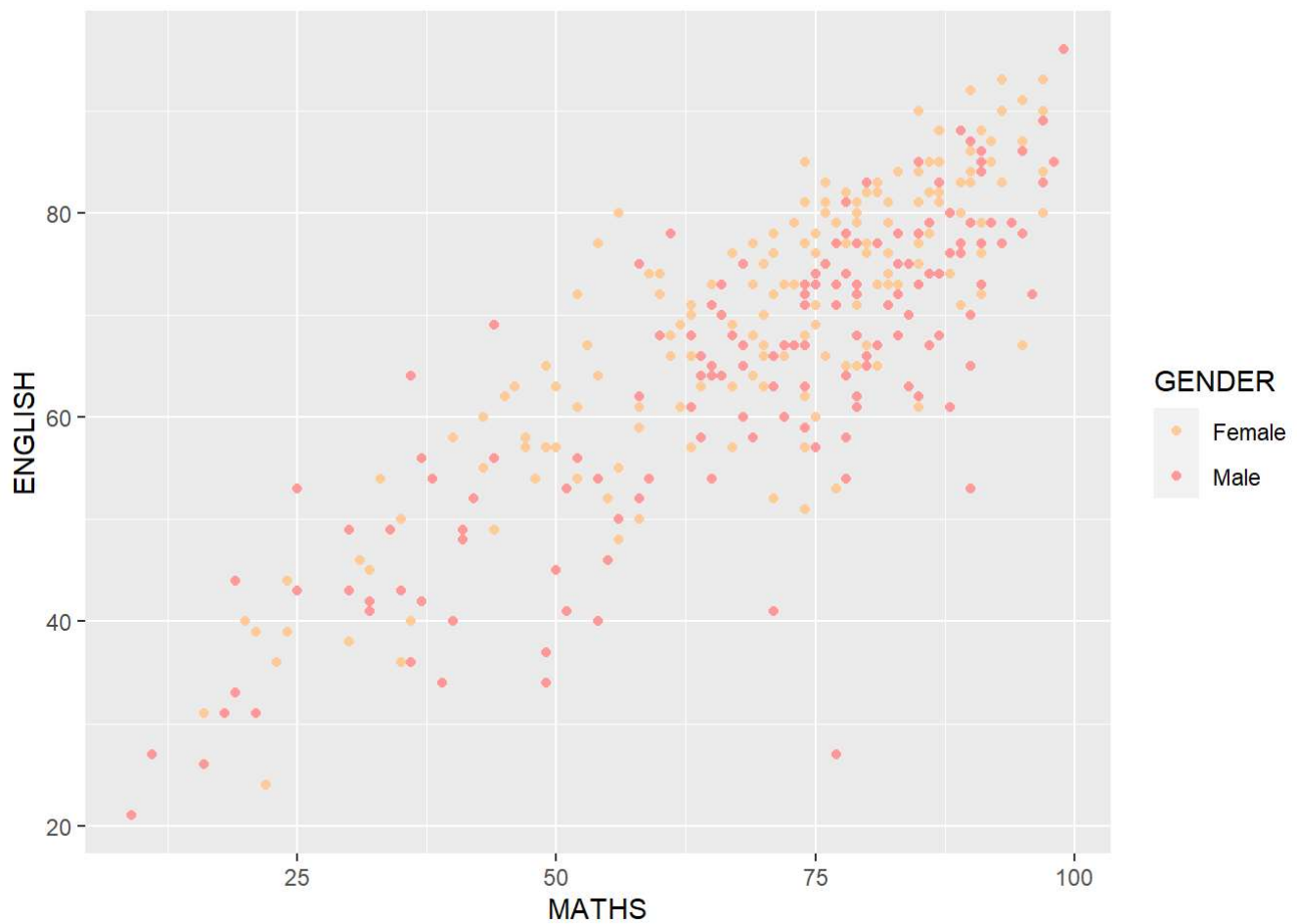
► Code



## geom\_point()

[geom\\_point\(\)](#) is especially useful for creating scatterplot. The code chunk below plots a scatterplot showing the Maths and English grades of pupils by using `geom_point()`.

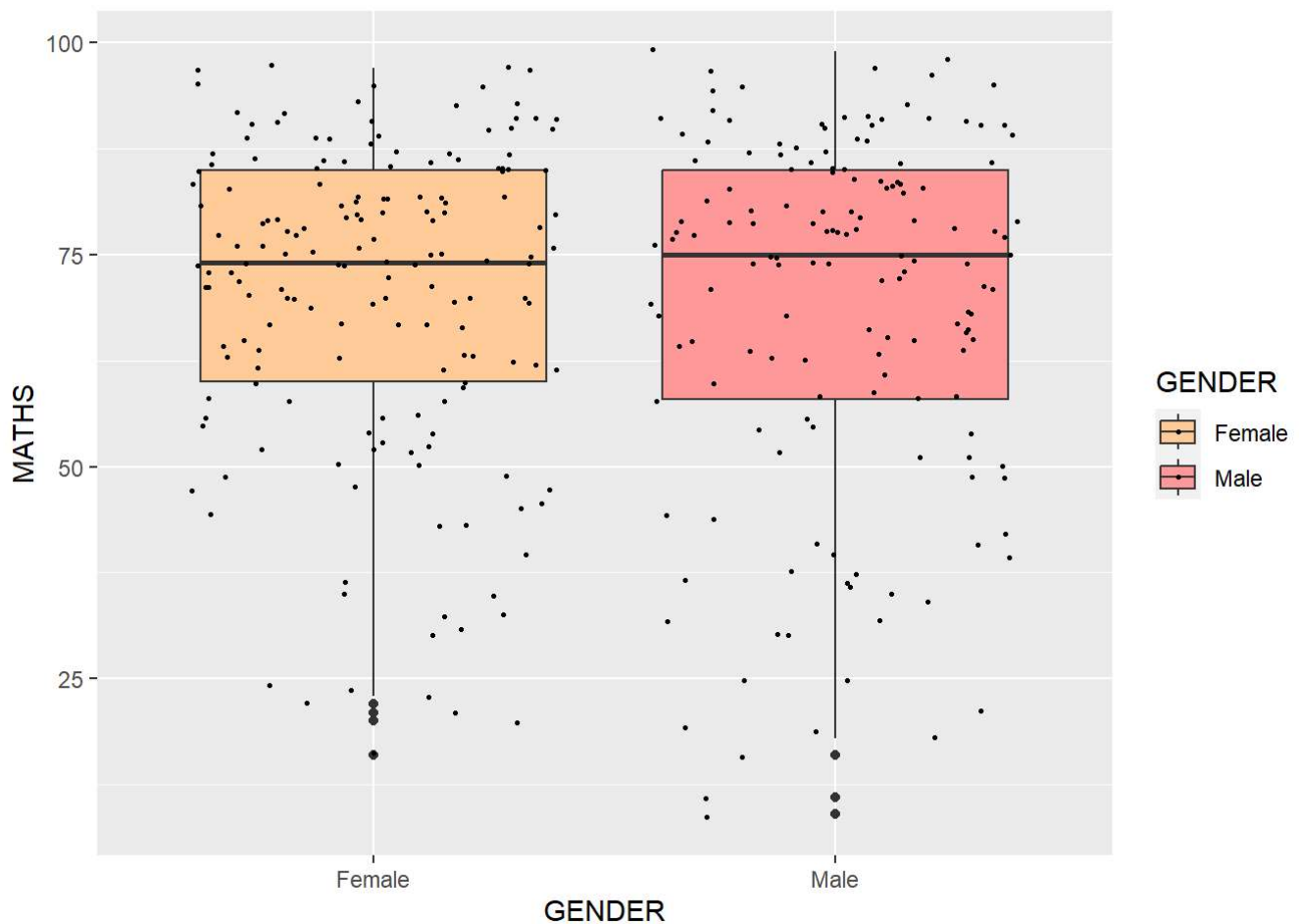
► Code



## geom objects can be combined

The code chunk below plots the data points on the boxplots by using both `geom_boxplot()` and `geom_point()`.

► Code



## Essential Grammatical Elements in ggplot2: stat

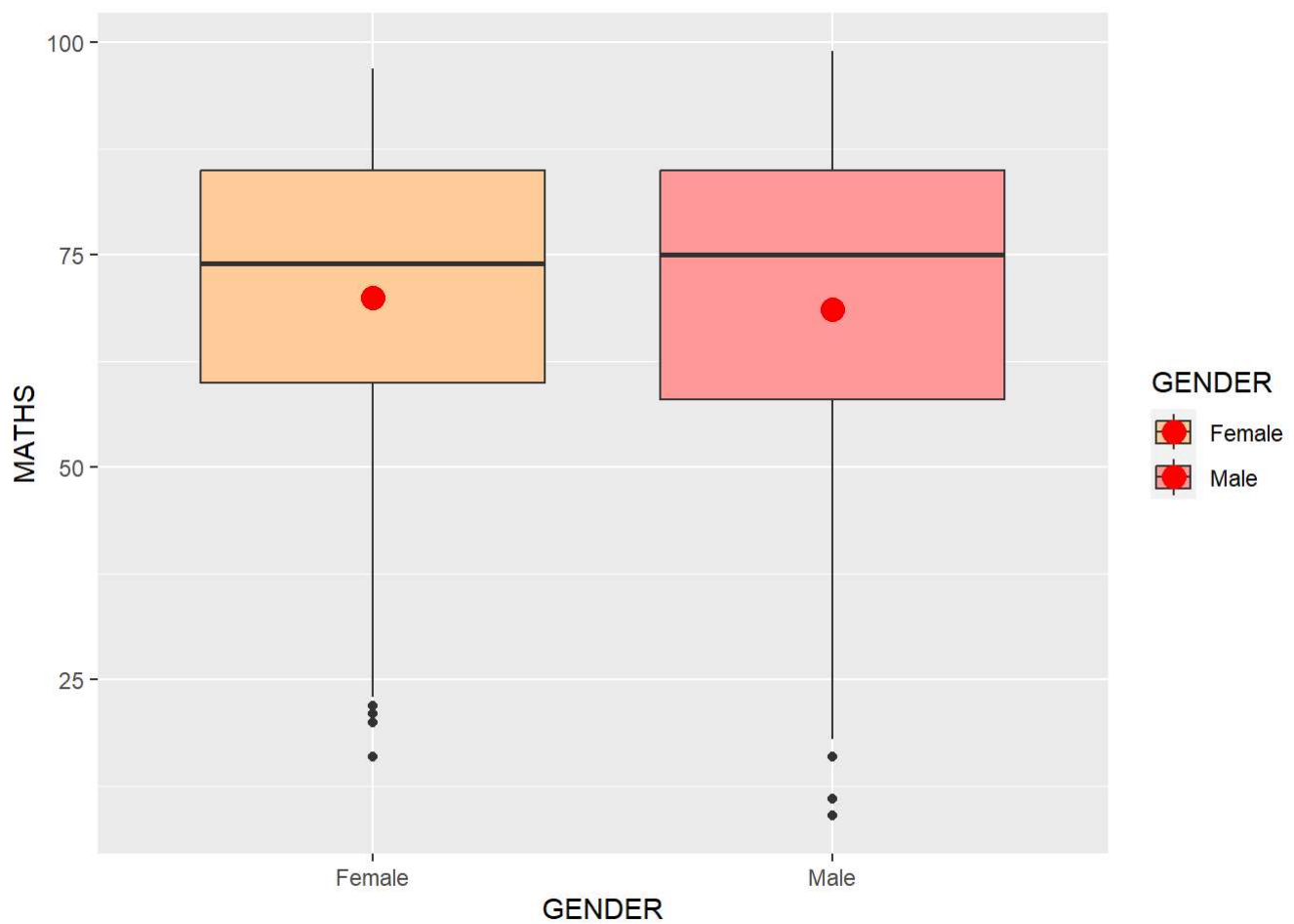
The [Statistics functions](#) statistically transform data, usually as some form of summary. For example:

- frequency of values of a variable (bar graph)
  - a mean
  - a confidence limit

### the `stat_summary()` method

The code chunk below adds mean values by using [stat\\_summary\(\)](#) function and overriding the default geom.

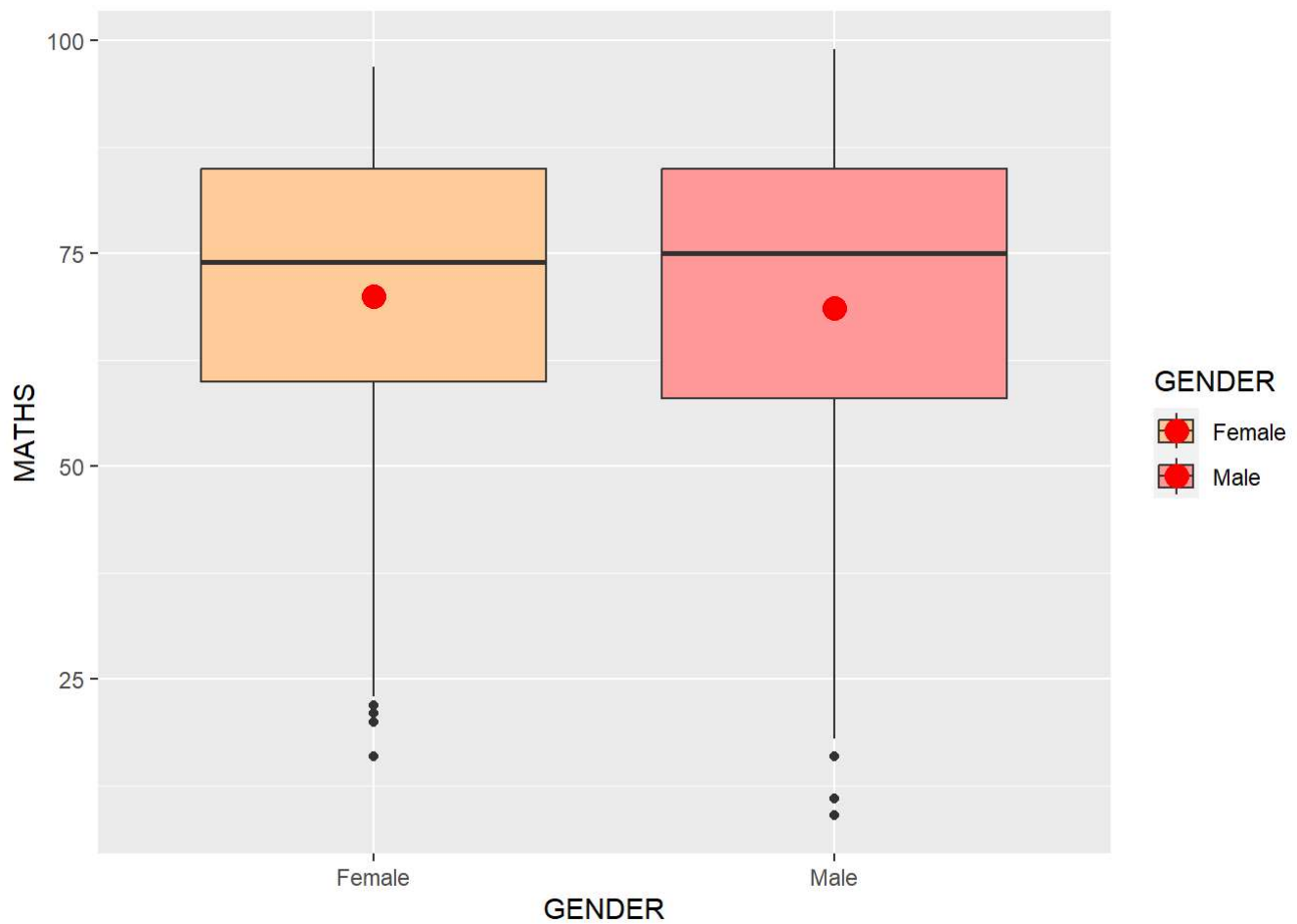
► Code



## the `geom()` method

The code chunk below adding mean values by using `geom_()` function and overriding the default stat.

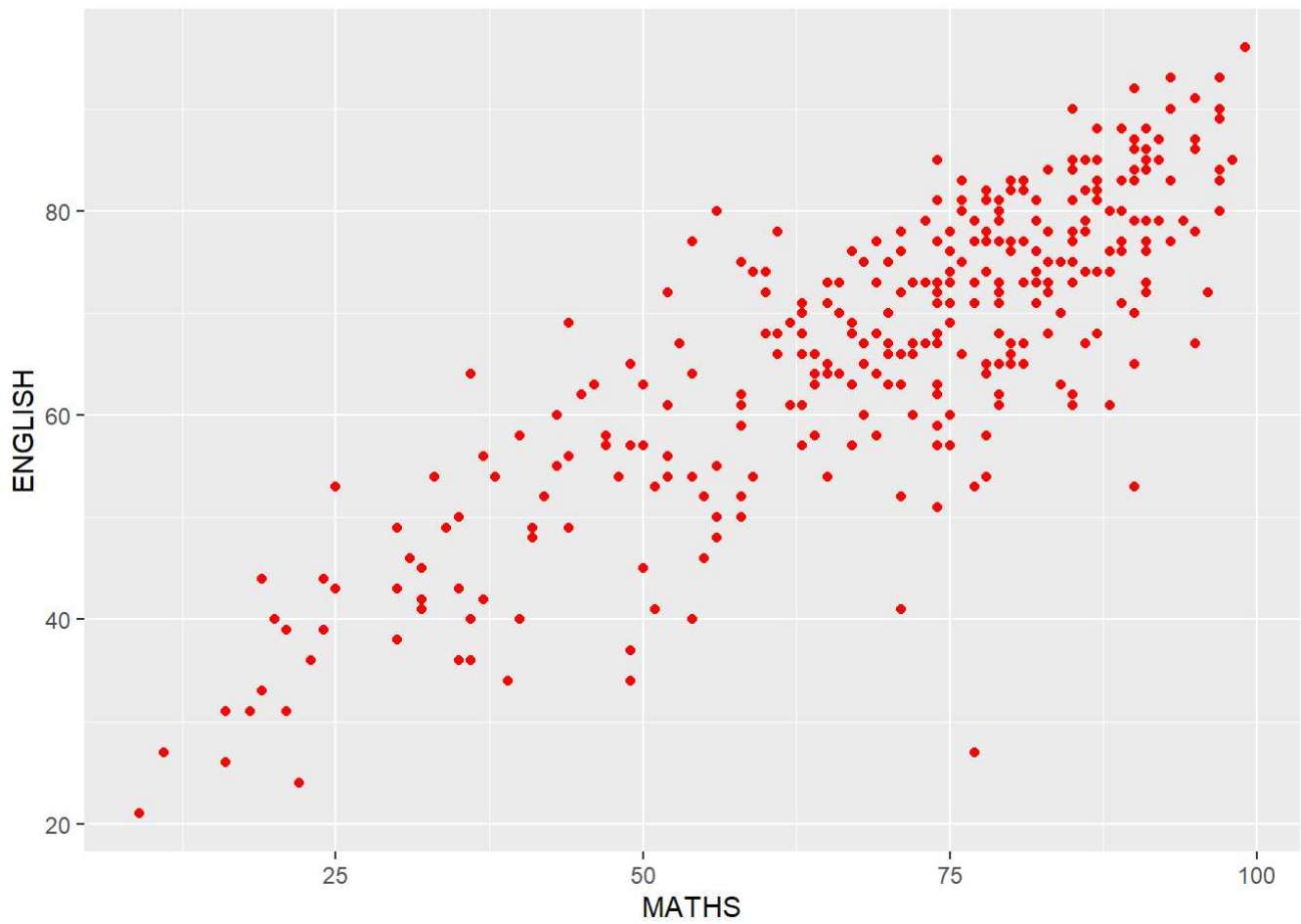
► Code



## Scatter Plot - `geom_point()`

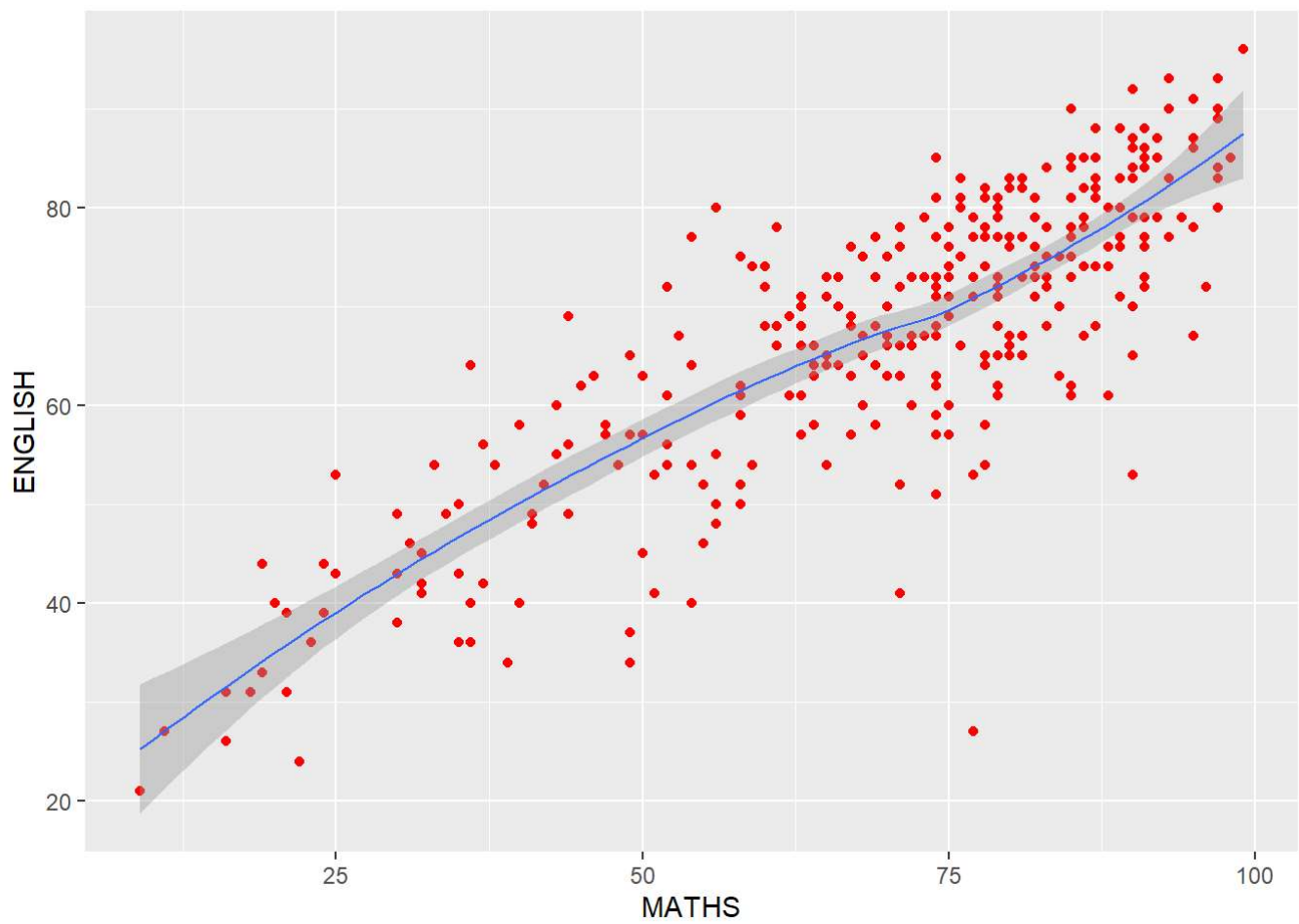
The scatterplot below shows the relationship of Maths and English grades of pupils. The interpretability of this graph can be improved by adding a best fit curve.

► Code



In the code chunk below, `geom_smooth()` is used to plot a best fit curve on the scatterplot.

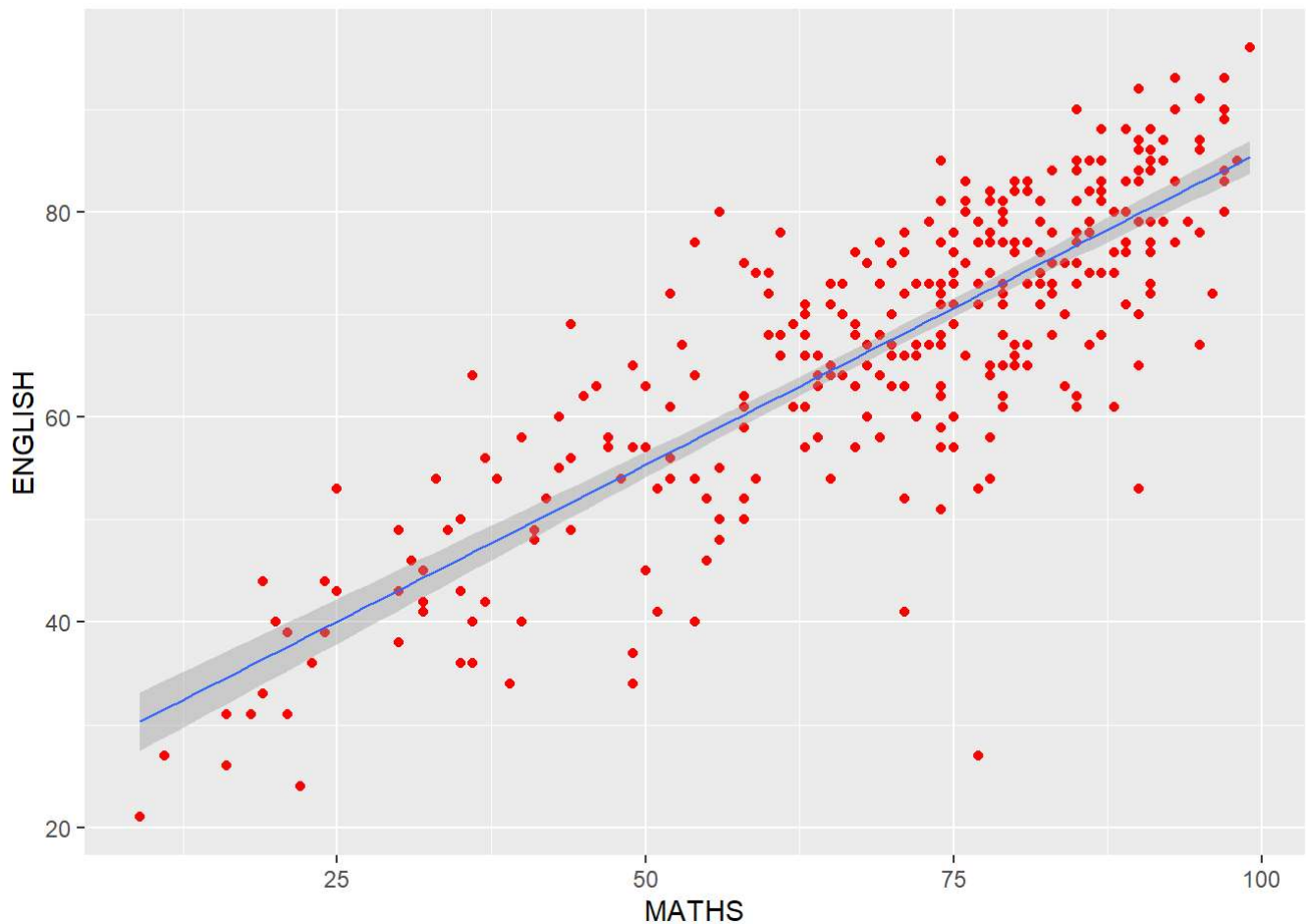
► Code



The default smoothing method can be overridden as shown below.

► Code





## Essential Grammatical Elements in ggplot2: Facets

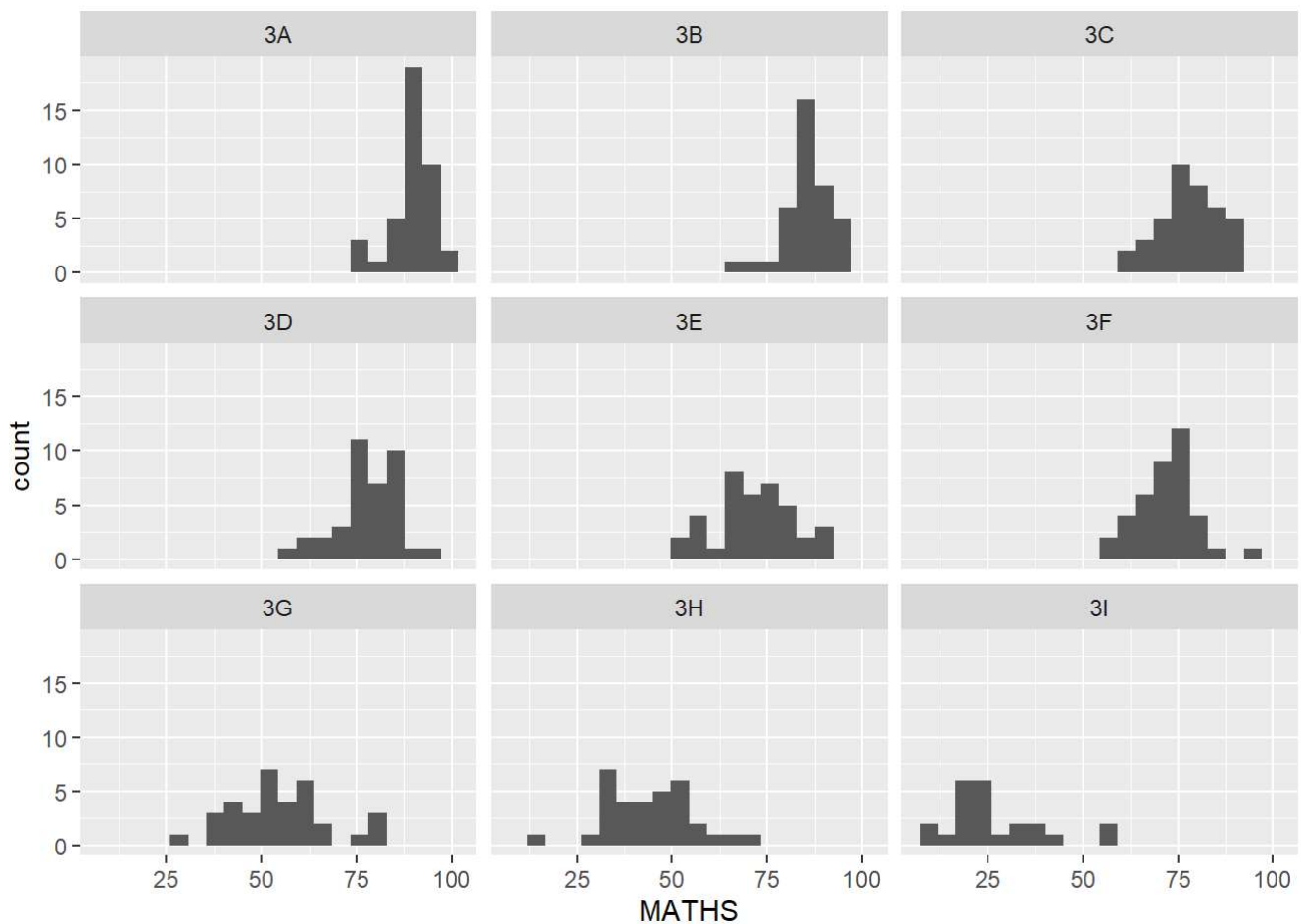
Facetting generates small multiples (sometimes also called trellis plot), each displaying a different subset of the data. They are an alternative to aesthetics for displaying additional discrete variables. ggplot2 supports two types of faceting, namely: `facet_grid()` and `facet_wrap()`.

### `facet_wrap()` function

`facet_wrap` wraps a 1d sequence of panels into 2d. This is generally a better use of screen space than `facet_grid` because most displays are roughly rectangular.

The code chunk below plots a trellis plot using `facet_wrap()`.

► Code

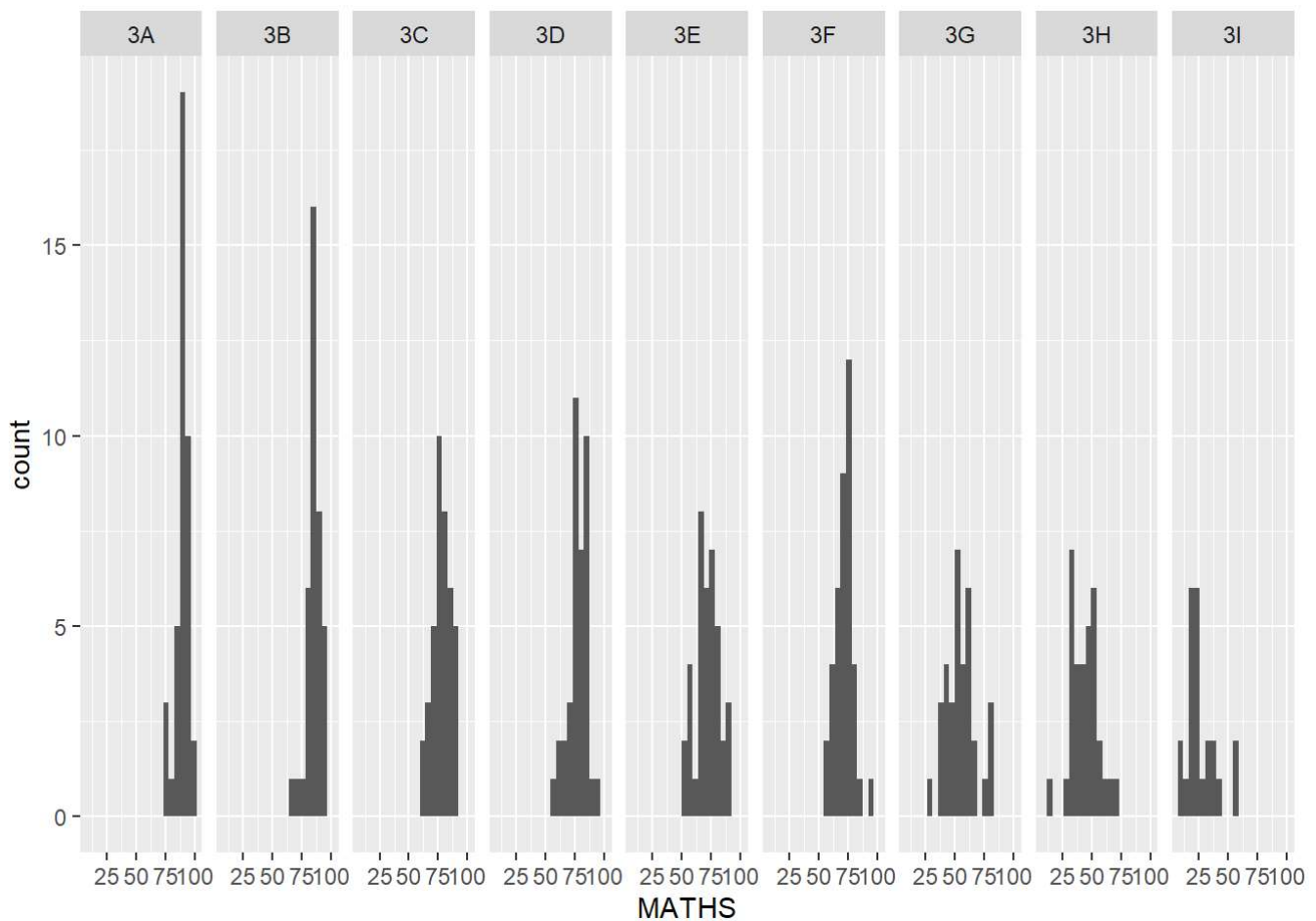


## facet\_grid() function

[facet\\_grid\(\)](#) forms a matrix of panels defined by row and column facetting variables. It is most useful when you have two discrete variables, and all combinations of the variables exist in the data.

The code chunk below plots a trellis plot using `facet_grid()`.

► Code



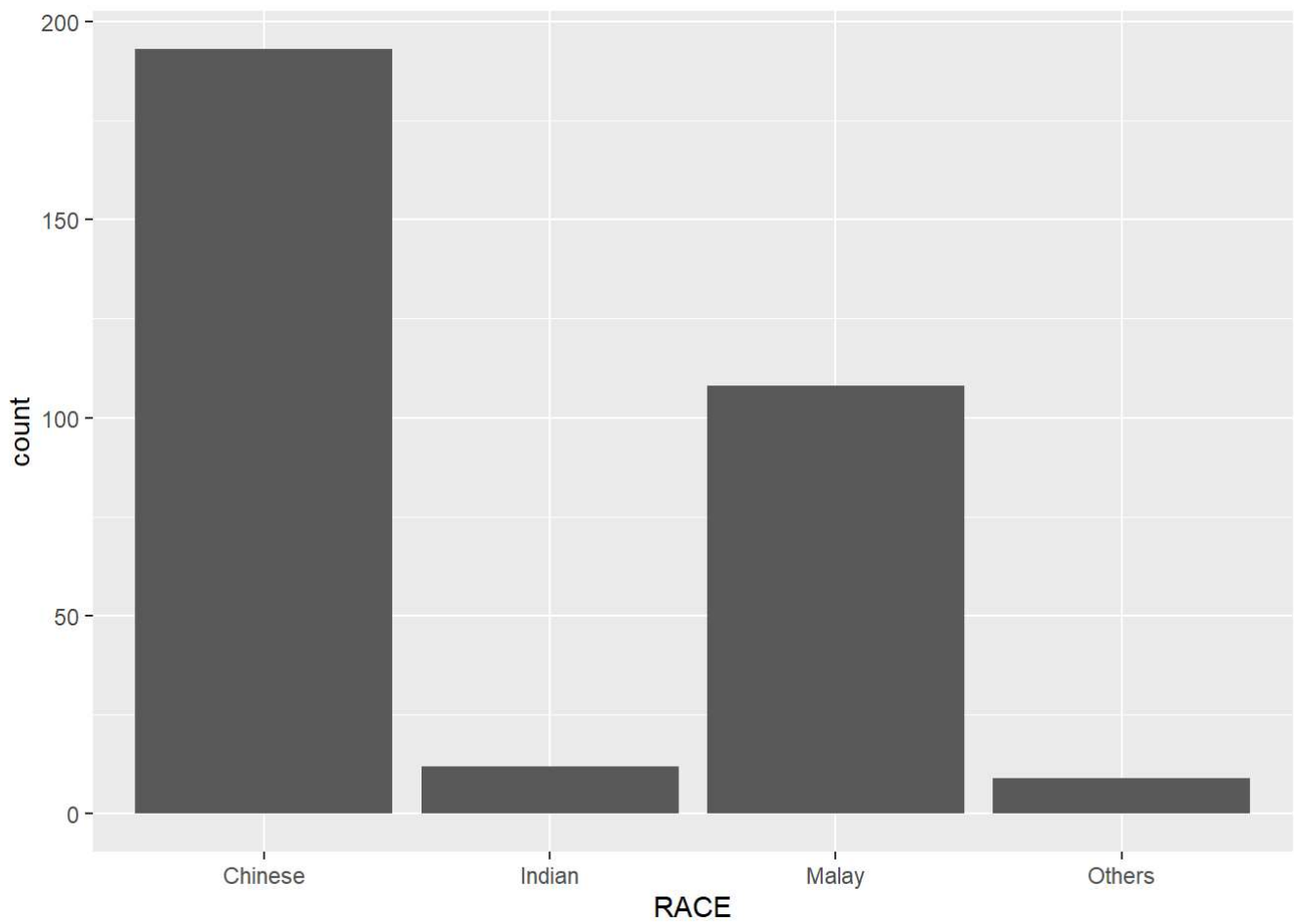
## Essential Grammatical Elements in ggplot2: Coordinates

The *Coordinates* functions map the position of objects onto the plane of the plot. There are a number of different possible coordinate systems to use

### Working with Coordinate

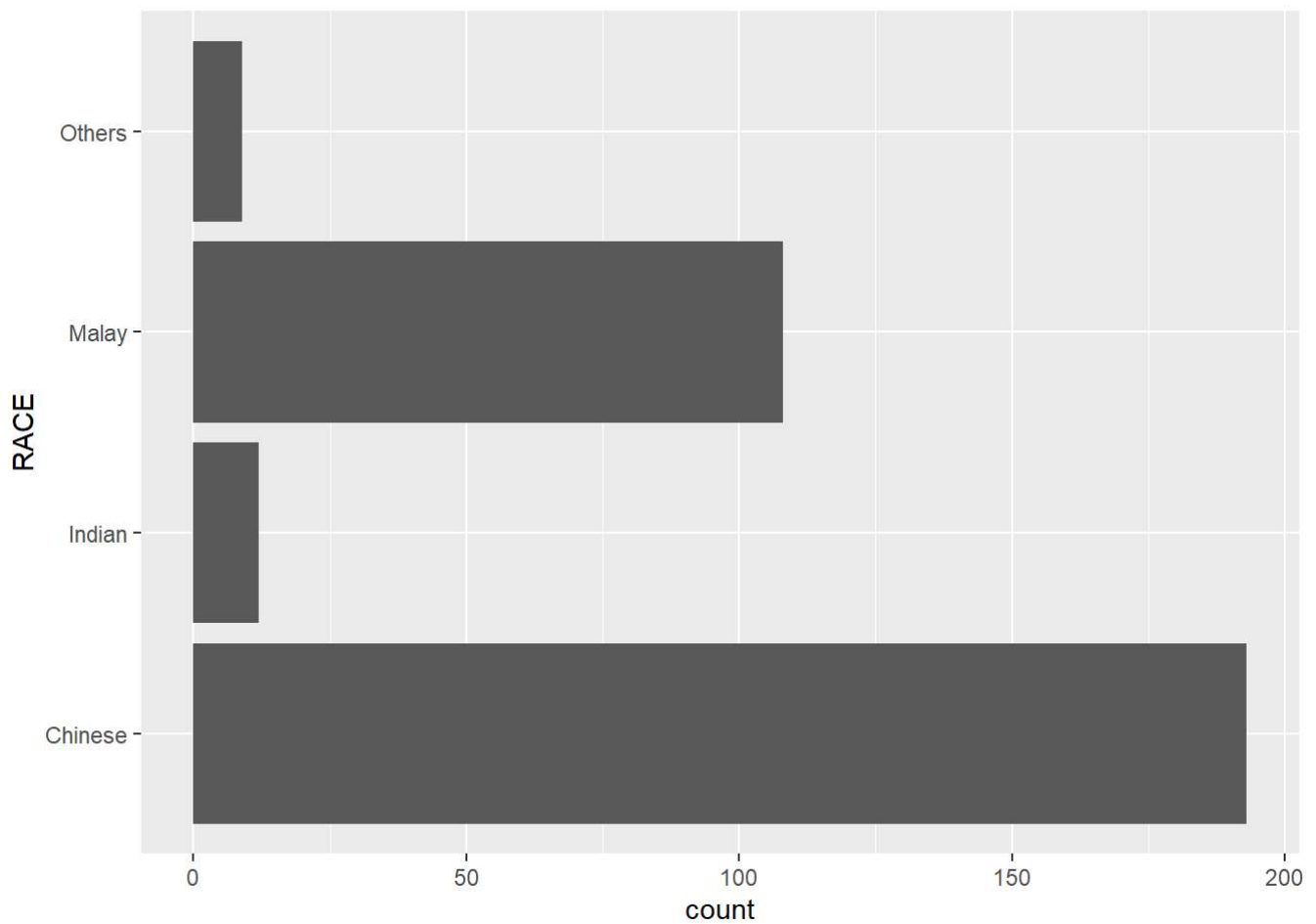
By the default, the bar chart of ggplot2 is in vertical form.

► Code



The code chunk below flips the horizontal bar chart into vertical bar chart by using `coord_flip()`.

► Code

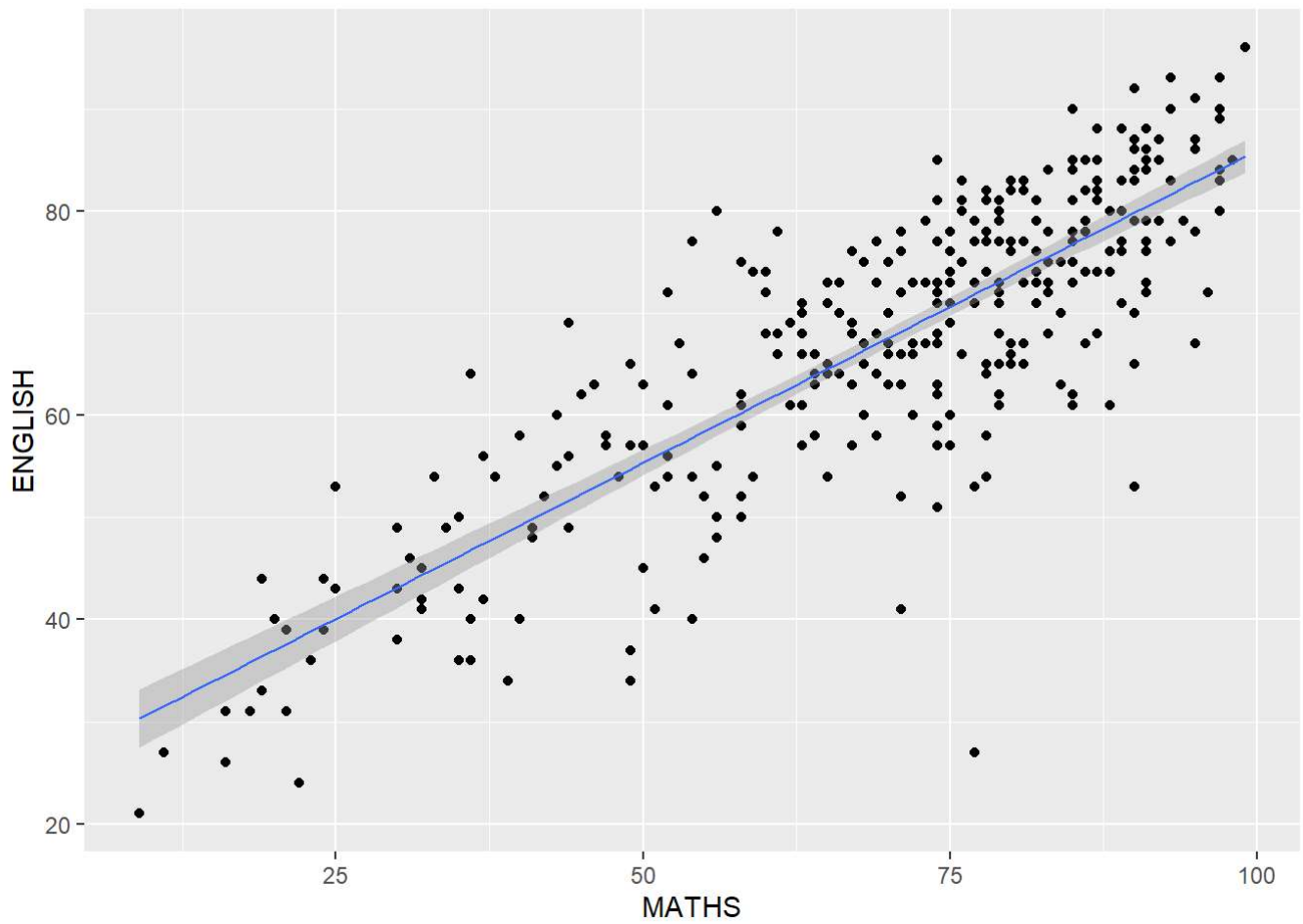


## Changing the y- and x-axis range

---

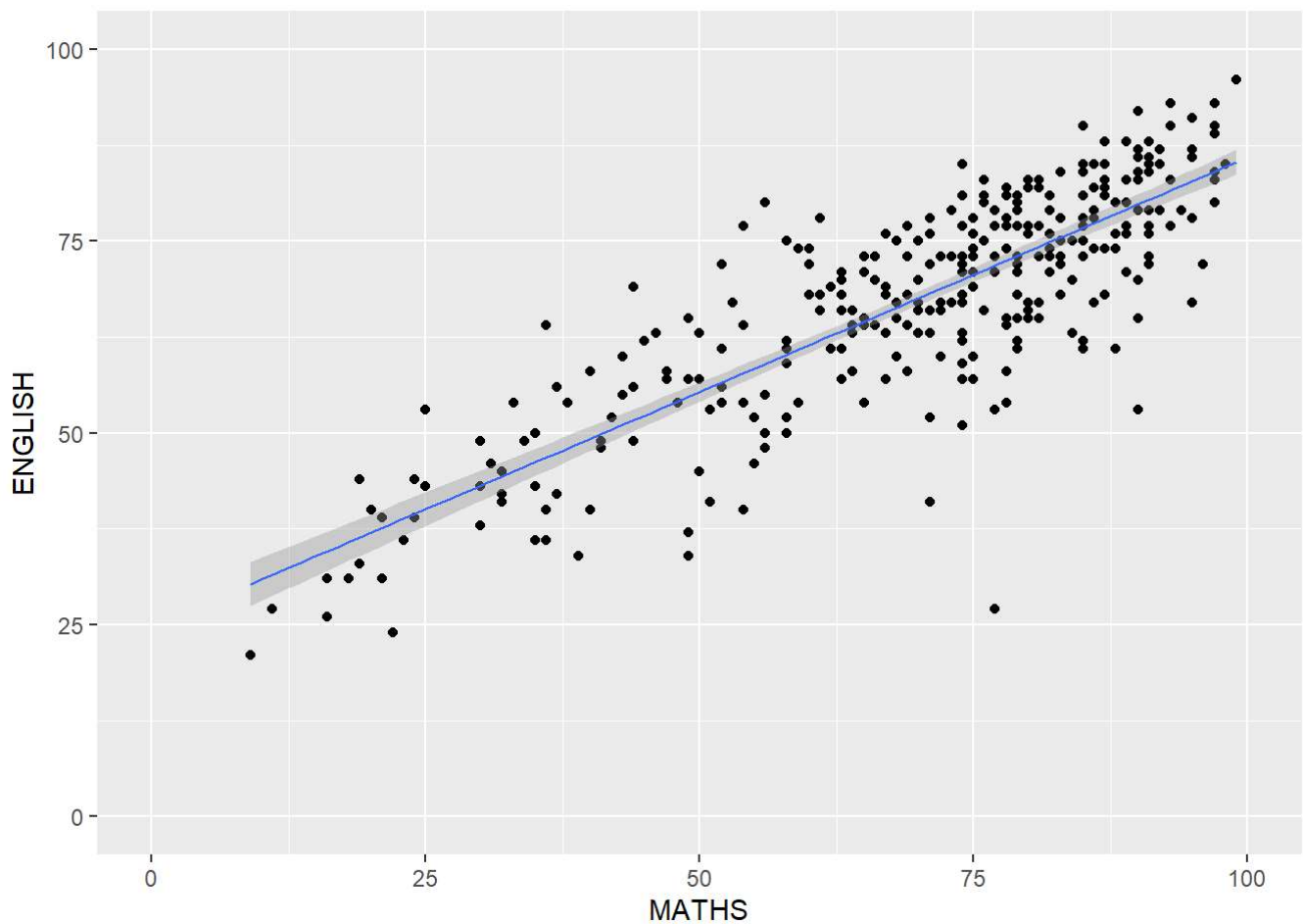
The scatterplot on the right is slightly misleading because the y-axis and x-axis range are not equal.

► Code



The code chunk below fixed both the y-axis and x-axis range from 0-100.

► Code



## Essential Grammatical Elements in ggplot2: themes

Themes control elements of the graph not related to the data. A list of theme can be found at this [link](#). Each theme element can be conceived of as either a line (e.g. x-axis), a rectangle (e.g. graph background), or text (e.g. axis title).

Gray

Classic

Minimal

The code chunk below plot a horizontal bar chart using `theme_gray()`.

► Code

