

Following are instructions on how to use the function supplied for this project to create plots of your raw data. It assumes you have installed RStudio as described in the document “Setup for RStudio”. See the document “Data management in RStudio” for how to manage your data in data frames.

The ability of R to produce publication-quality pdf plots of statistical data is a major benefit of the software. The graphics engine is based on a software package called `ggplot2` written by Hadley Wickham of Rice University. It is an implementation of Wilkinson’s *The Grammar of Graphics* (2005), which specifies a graphics programming language. The `gg` in `ggplot2` stands for Grammar of Graphics.

The philosophy of `ggplot2` is to give the user the flexibility of creating custom plots by combining or layering the plotting components from a general set of components. The advantage of this approach is that you are not restricted to a limited set of predetermined plot types. Instead, you can combine the components in an unlimited way to produce virtually any type of plot. The disadvantage is that it takes more effort to compose your plot. You can use the function provided for this course to create your plots of the raw data without understanding any of the details of `ggplot2`. Inspect the function source in `dp4dsFunctions.R` if you are interested in the details.

A few of the types of components that you can combine are

- Aesthetics — the general visual attributes of plot elements. Examples include x and y position of elements, size of elements, shape of elements, and color of elements.
- Geometries — the general type of plot elements. Examples include points, lines, line segments, bars, and text.
- Statistics — the results of computations from the data that are displayed in the plot. Examples include max, min, and mean displayed in a box plot and smooth displayed as a best-fit continuous curve.
- Coordinates — the coordinate system of the plot. Examples are cartesian, polar for pie charts, and map for cartographic displays.

1. *To plot raw data*

Suppose your data frame is named `asgnData` and you have an independent variable called `NumSorted` and five dependent variables called `InsertAsgn`, `SelectAsgn`, `HeapAsgn`, `MergeAsgn`, and `QuickAsgn`. (The following data are fictitious.)

```
> asgnData
  NumSorted InsertAsgn SelectAsgn HeapAsgn MergeAsgn QuickAsgn
1         500     999999     899999     799999     699999     599999
2        1000     944444     844444     744444     644444     544444
3        1500     988888     888888     788888     688888     588888
4        2000     911111     811111     711111     611111     511111
5        2500     922222     822222     722222     622222     522222
6        3000     933333     833333     733333     633333     533333
7        3500     944444     844444     744444     644444     544444
8        4000     955555     855555     755555     655555     555555
9        4500     966666     866666     766666     666666     566666
10       5000     977777     877777     777777     677777     577777
11       5500     988888     888888     788888     688888     588888
12       6000     999999     899999     799999     699999     599999
```

The function `dp4dsRawPlot()` provided for this project makes plots a family of raw data in a line plot for a given data frame assuming the first column is the independent (x) variable and the remaining columns are the dependent (y) variables. It has five parameters:

- A data frame.
- The label for the x-axis.
- The label for the y-axis.
- The label for the legend.
- A vector of labels of the dependent variables for the legend.

2. *To install the function*

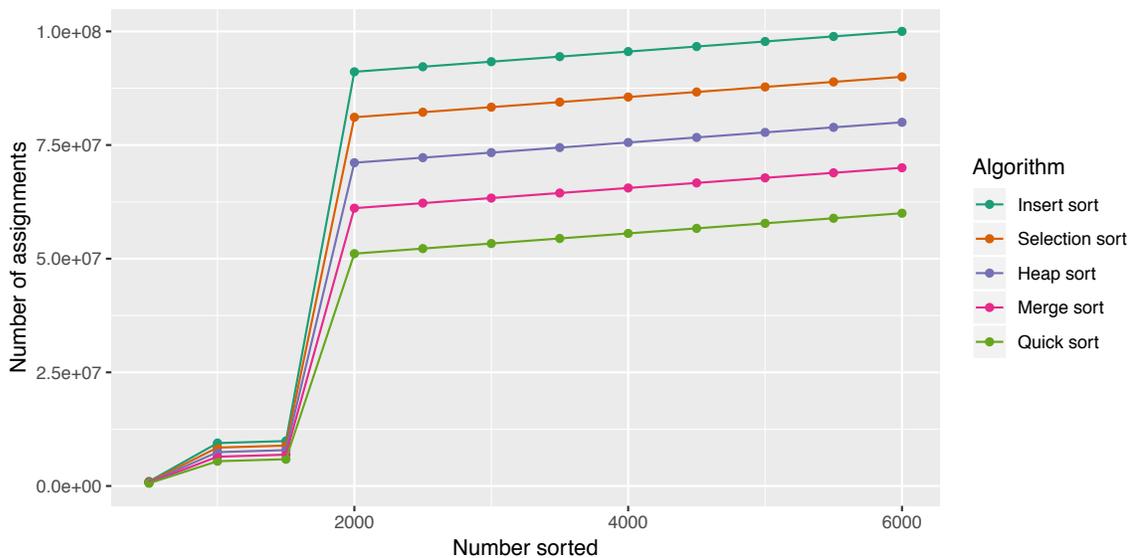
Open the file `dp4dsFunctions.R` in RStudio. Click the source button in the upper right corner of the edit pane.

3. *To call the function*

Enter the following on the RStudio command line. Do not enter the `+`. It is the R command line continuation prompt.

```
> dp4dsRawPlot(asgnData, "Number sorted", "Number of assignments", "Algorithm",
+ c("Insert sort", "Selection sort", "Heap sort", "Merge sort", "Quick sort"))
```

The following plot is produced.



4. *To save the plot*

With the Plot tab selected and the plot showing in the pane, click `Export → Save as PDF ...`. A useful feature is the ability to set a custom size for the plot in the PDF size field. This feature will scale the plot without distorting the text. Because the plot is generated with vector graphics, you can zoom in to inspect the details of the plot without encountering any jaggies that would be present if you saved it as a screenshot.

5. *To plot a different set of data*

If you want to plot a different set of data, make a new data frame with the data you want to plot. Keep the first variable as the independent variable. See the document “Data management in RStudio”.