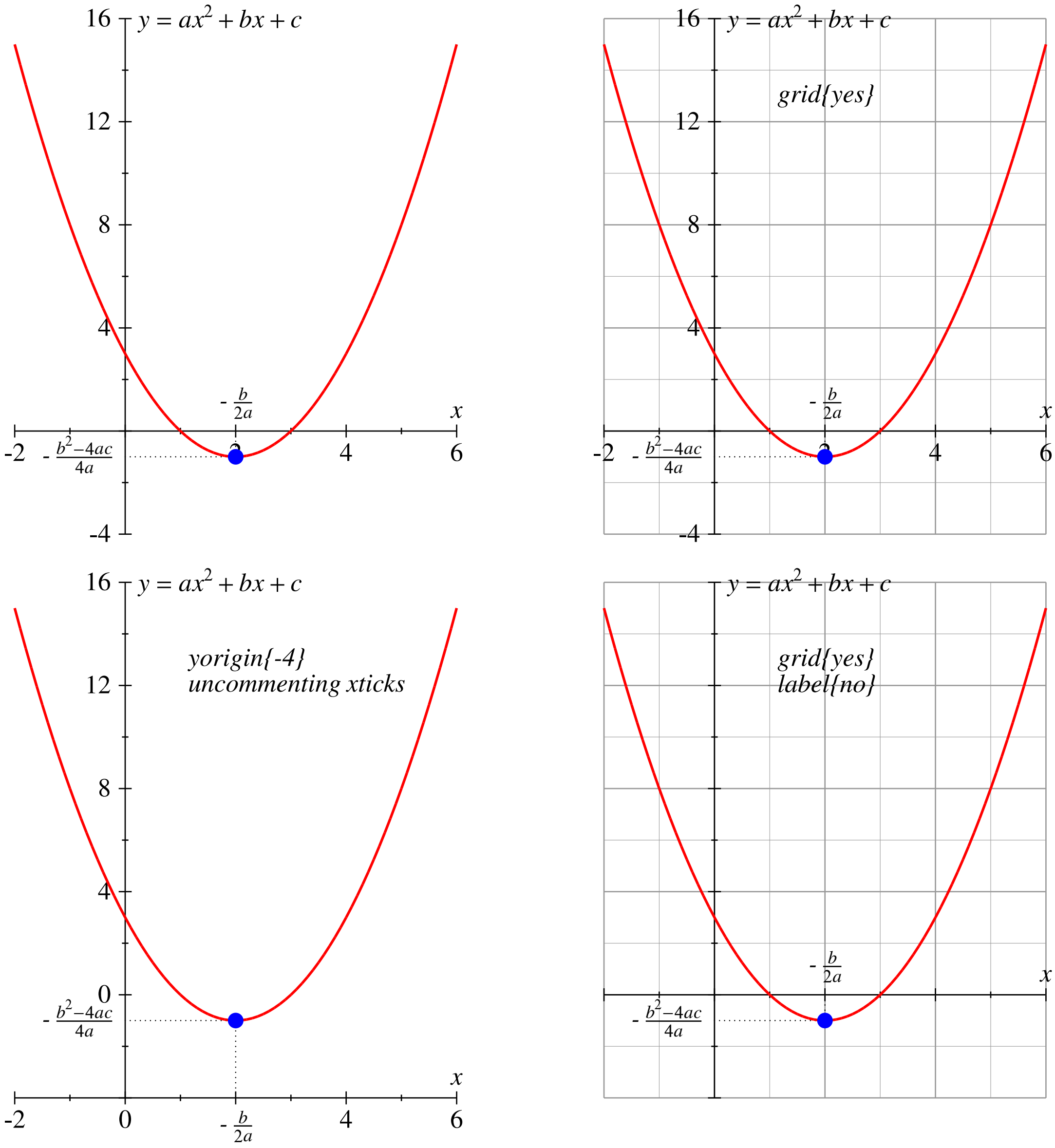
Lout Graph samples (modified @Graph – 17 August, 2019 – E.B.)

Ticks, labels and grid

```
@Graph
  style{axes} width{6c} height{7c}
  grid{no} label{yes}         # default values
  xorigin{0.0} yorigin{0.0}
  xticksep{2} xnsubtick{2}    # xticks{-2@ -1 0@ 1 2 () 3 4@ 5 6@}
  yticksep{4} ynsubtick{2}
{
  @Data pairs{dotted} linewidth{0.5p} { 0 -1 2 -1 2 0 }
  @Data color{red} pairs{solid} linewidth{1p} {
    xloop from {-2} to {6} by { 0.1 } do {
      x {{x - 2} * {x - 2} - 1}
    }
  }
  @Data color{blue} points{filledcircle} symbolsize{0.3f} { 2 -1 }
}
```

The above code doesn't show math objects appearing on the graph, but the source does.



Ticks can be defined as usual with *xmin*, *xmax*, *xticksep* or nothing for automatic ticks or *xticks* for manual ticks. a Tick is associated with a label displayed wth the tick.

The label is determined automatically according to the tick position on the axis. When defining ticks manually, for example

```
xticks{-2@ -1 0@ 1 2 (data1) 3 4@ 5 6@}
```

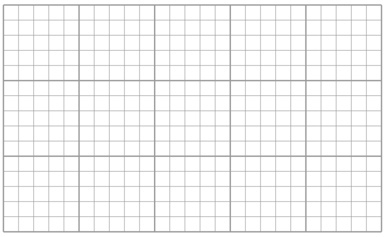
‘-2@’ identifies the tick at position *-2*, the @ indicating that the value -2 is to be displayed. ‘2 (data1)’ defines a tick at position *2* and with label *data1*. Using ‘2 ()’ is a way to display a tick without any label.

‘-1’ defines a subtick which appears smaller on the graph, and without any label. Subticks can be defined automatically using the *xnsubtick* which sets how many subranges would there be between two subsequent ticks.

Finally a *label* option, with possible values *yes*, *no*, *x* or *y*, controls globally if labels are displayed along the axis.

The *grid* option can take one the following values: *yes*, *no* (the default), *x* or *y*. It uses the ticks and subticks definitions allowing to have a coarse and a thiner grid. Hence, the graph

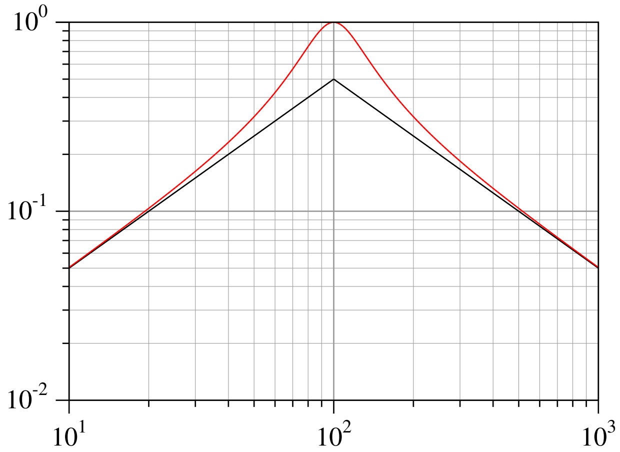
```
@Graph
  style{none} width{5c} height{3c} grid{yes}
  xmin{0} xmax{5} xticksep{1} xnsubtick{5}
  ymin{0} ymax{3} yticksep{1} ynsubtick{5}
{
}
```



can just display a 5x3 cm 1 cm-based grid with a 2 mm subgrid.

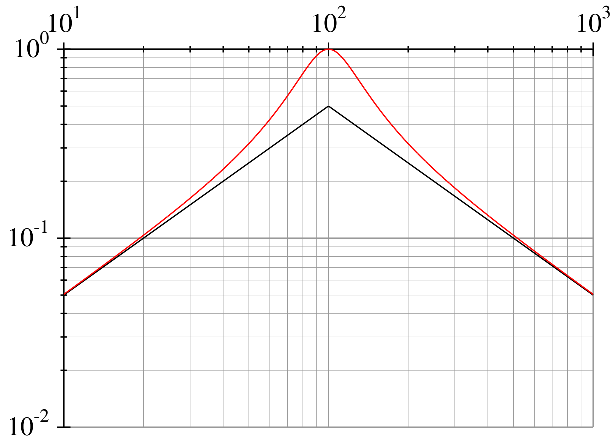
Styles supported are: *frame*, *axes*, *none*, *xygrid*, *xgrid*, *ygrid*. Note that when the *grid* option is used, it bypassed the grid styles. It is worth to use the *grid* option instead of the grid styles.

An example with a log scale, now.



```
@Graph
  width{7c} height{5c}
  style{frame} xextra{0c} yextra{0c}
  grid{yes}
  xlog{10} ylog{10}
{
  @Data pairs{solid} color{black}{
    10 0.05 100 0.5 1000 0.05
  }
  @Data pairs{solid} color{red} {
    # include data from "data/data1"
  }
}
```

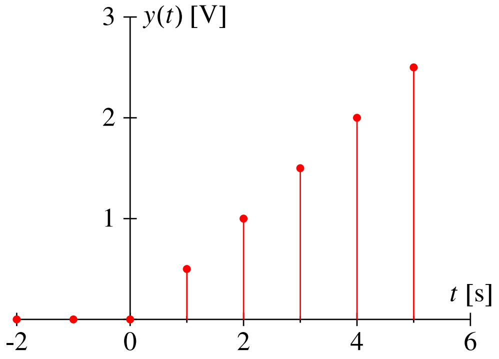
One has to note that, for surch a graph, the data have to be stored in a file that is generated by an external tool (computing environment, scripting languages, ...) and included. The data in the file must be put in ‘x y’ tuples.



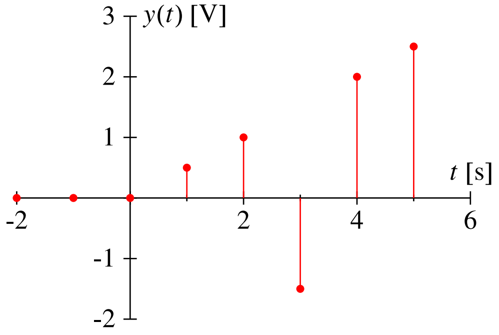
```
@Graph width{7c} height{5c}
  style{axes} xorigin{10} yorigin{1}
  grid{yes}
  xlog{10} ylog{10}
{
  @Data pairs{solid} color{black}{
    10 0.05 100 0.5 1000 0.05
  }
  @Data pairs{solid} color{red} {
    # include data from "data/data1"
  }
}
```

Here, as well, a noteworthy feature of the `axes` style: when the x axis is at the top of the graph, the labels are displayed above the axis, else they appeare below the axis.

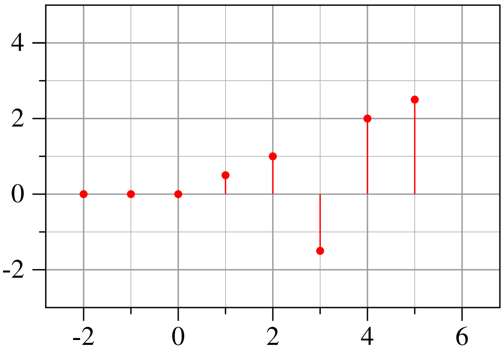
ycomb data vizualisation



```
@Graph
  style{axes} xorigin{0} yorigin{0}
  xticksep{2} xnsubtick{2}
  objects{
    @N at{6 0} margin{0.5f} @M{t`[@R{s}]}
    @E at{0 3} margin{0.5f} @M{y(t)`[@R{V}]}
  }
  color{red}
{
  @Data points{ycomb} {
    -2 0 -1 0 0 0 1 0.5 2 1 3 1.5 4 2 5 2.5
  }
}
```



The line drawn with each point is referenced to the position of the x axis (defined by the *yorigin* option).

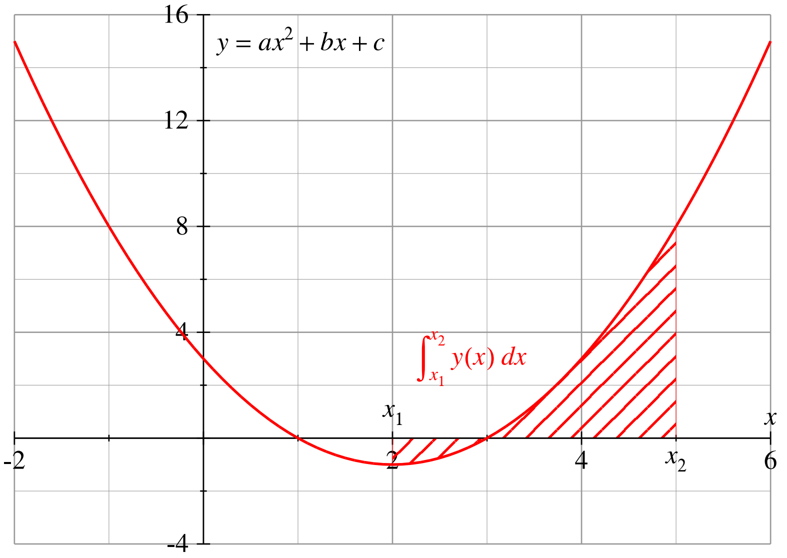


Even though the *axes* style is not used.

```
@Graph
  style{frame} grid{yes}
  xorigin{0} yorigin{0}
  xticksep{2} xnsubtick{2}
  yticksep{2} ynsubtick{2}
{
  @Data points{ycomb} color{red}{
    -2 0 -1 0 0 0 1 0.5 2 1 3 -1.5 4 2 5 2.5
  }
}
```

On the use of textures

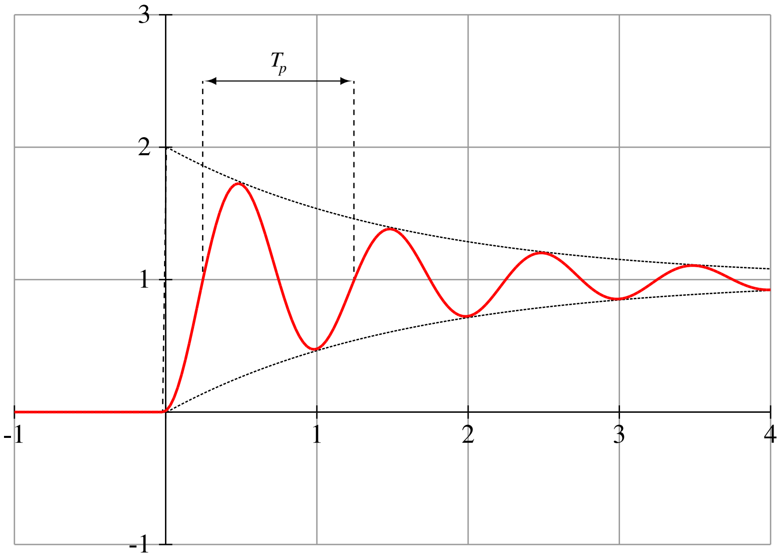
An example of using a texture to emphasize an area



```
@Graph
  style{axes} width{10c} height{7c}
  grid{yes}
  xorigin{0.0} yorigin{0.0}
  xticksep{2} xnsubtick{2}
  yticksep{4} ynsubtick{2}
  objects{
    @N at{6 0} margin{0.5f} @M{x}
    @E at{0 15} margin{0.5f} @M{y=a x sup 2 + b x +c}
    @N at{2 0} margin{0.5f} @M{x sub 1}
    @S at{5 0} margin{0.5f} @M{x sub 2}
    @E at{3.5 3} red @Color @Math{int from{x sub 1} to{x sub 2} y(x) ` dx}
  }
{
  @Data color{red} pairs{solid} linewidth{0.2p}
    texture{ striped angle { 45d } gap{5p}} paint{yes}{
    xloop from {2} to {5} by { 0.1 } do {
      x {{x - 2} * {x - 2} - 1}
    }
    # close the surface
    5 0 2 0 2 -1
  }
  # Redraw the x axis (nicer)
  @Data pairs{solid} {-2 0 6 0}
  @Data color{red} pairs{solid} linewidth{1p} {
    xloop from {-2} to {6} by { 0.1 } do {
      x {{x - 2} * {x - 2} - 1}
    }
  }
}
```

We draw the curve two times. The first is to draw the textured area. Note that we terminate by adding points to close the surface. The second time, we draw the curve on the full *x* axis.

Using @Diag to add information onto a graph



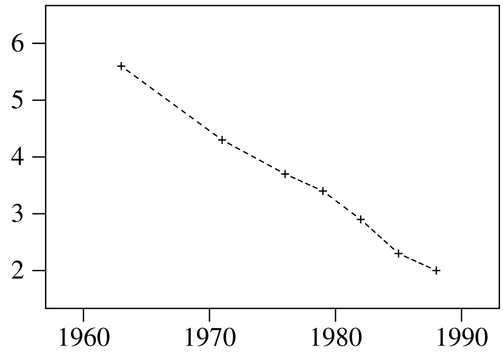
```
@Diag{
  @Graph
    style{axes} width{10c} height{7c}
    grid{yes}
    xorigin{0.0} yorigin{0.0}
    objects {
      @CTR at{0.245 2.5} {A::@Node margin{0.01f} outlinestyle{noline}}
      @CTR at{1.245 2.5} {B::@Node margin{0.01f} outlinestyle{noline}}
    }
  {
    @Data pairs{dashed} {0.245 1 0.245 2.5}
    @Data pairs{dashed} {1.245 1 1.245 2.5}
    @Data pairs{dashed} {
      # include {"data/data3"}
    }
    @Data pairs{dashed} {
      # include {"data/data4"}
    }
    @Data color{red} pairs{solid} linewidth{1p} {
      include {"data/data2"}
    }
  }
  @Link arrow{both} arrowstyle{curvedsolid} backarrowstyle{curvedsolid}
    ylabel{@M{T tsub p}} from{A} to{B}
}
```

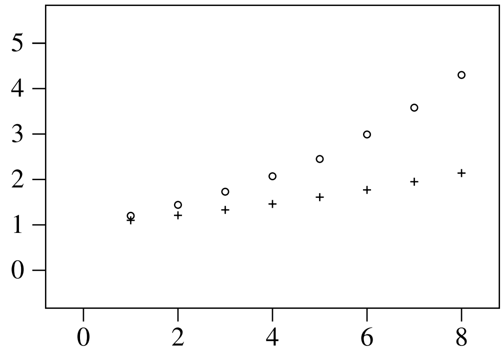
Arrows can't be added through the @Graph symbol, so we use the @Diag symbol, define the two points *A* and *B*, and use them to draw the arrow between thes two points.

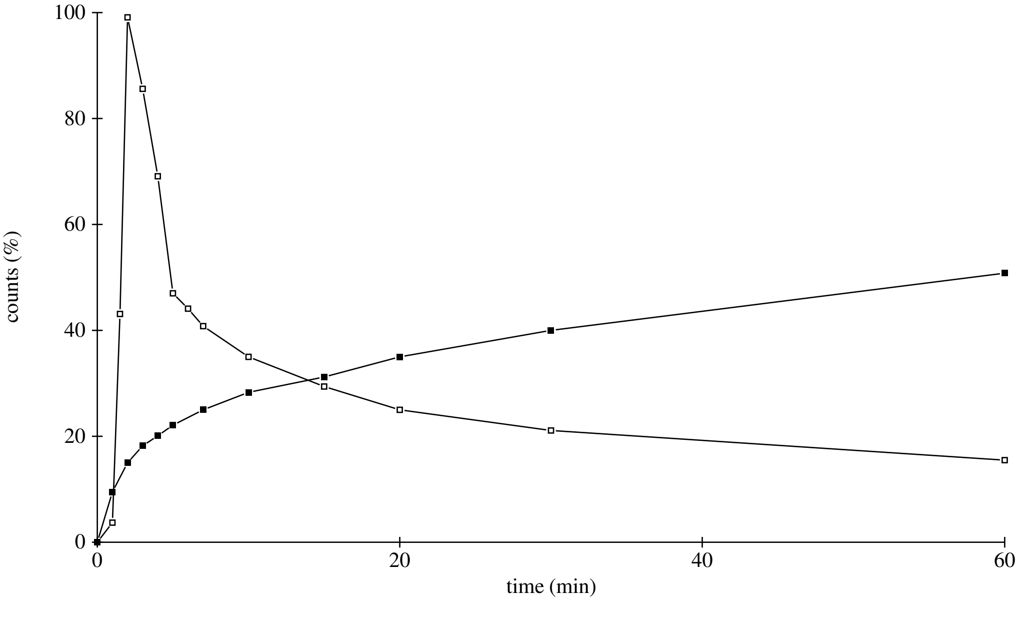
Lout User's Guide examples

For memory, the figures used in the User's Guide rendered by the modified module.

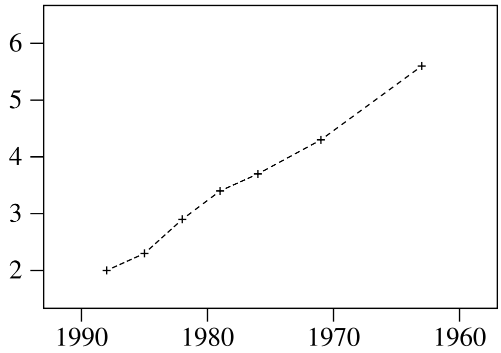
New South Wales road deaths, 1960–1990
(fatalities per 100 million vehicle km)

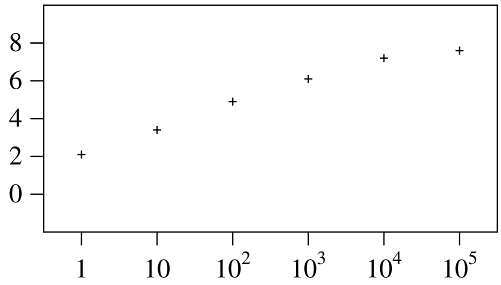


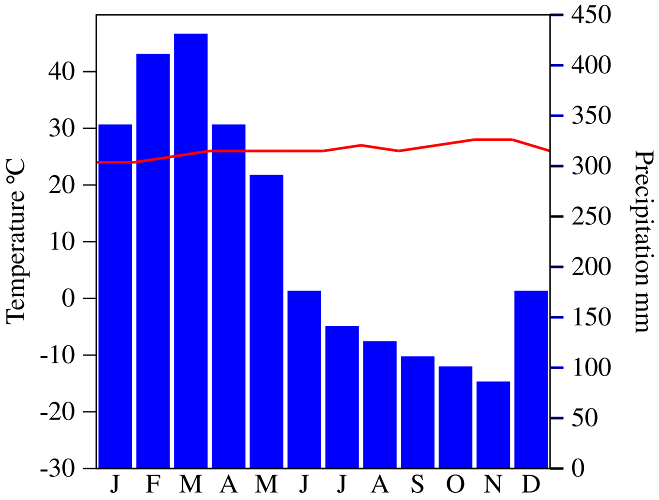




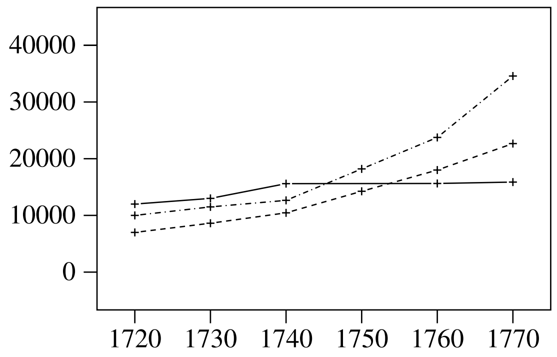
New South Wales road deaths, 1960–1990
(fatalities per 100 million vehicle km)



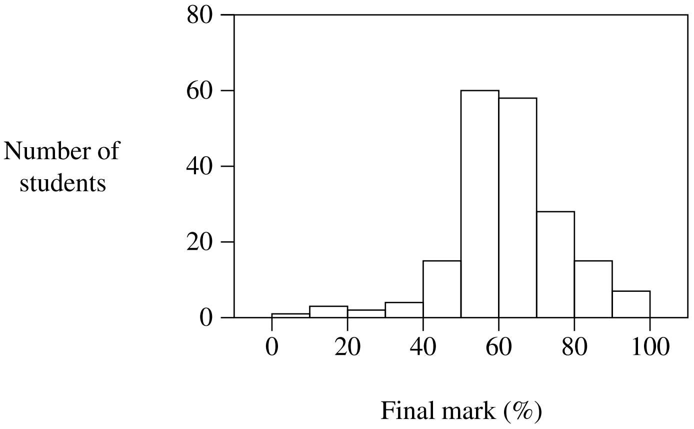




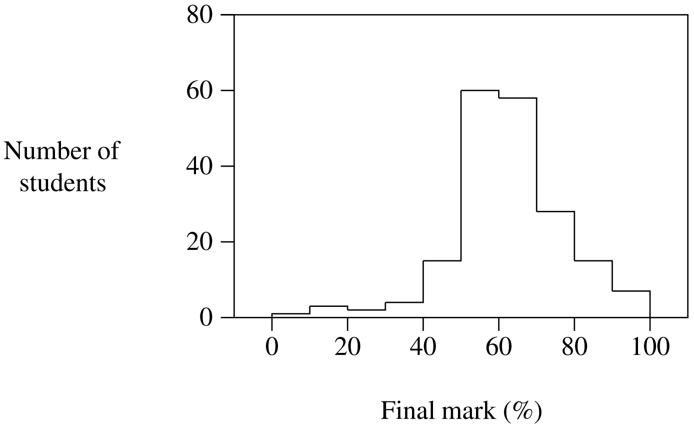
Estimated population of Boston, New York, and Philadelphia



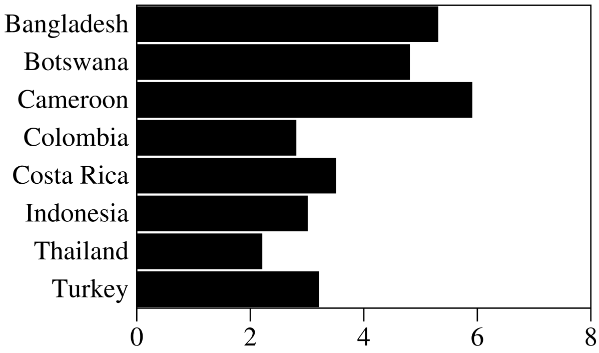
Computer Science 3 Results (1993)

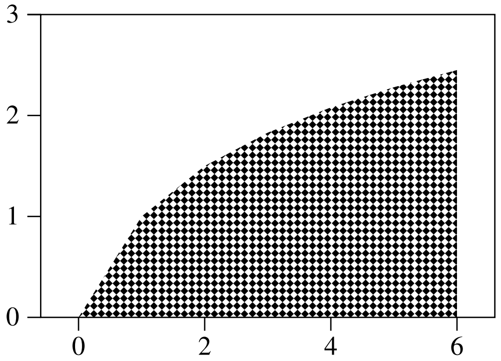


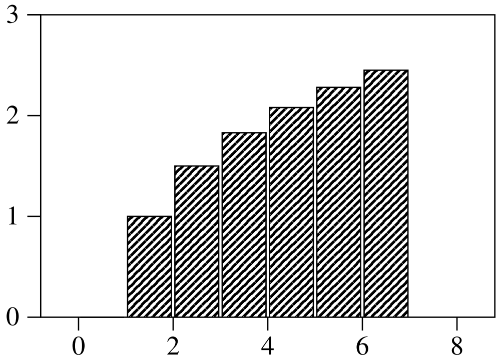
Computer Science 3 Results (1993)

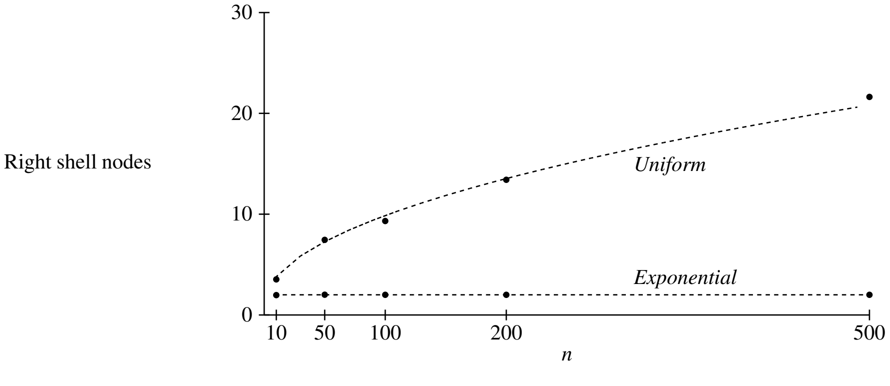


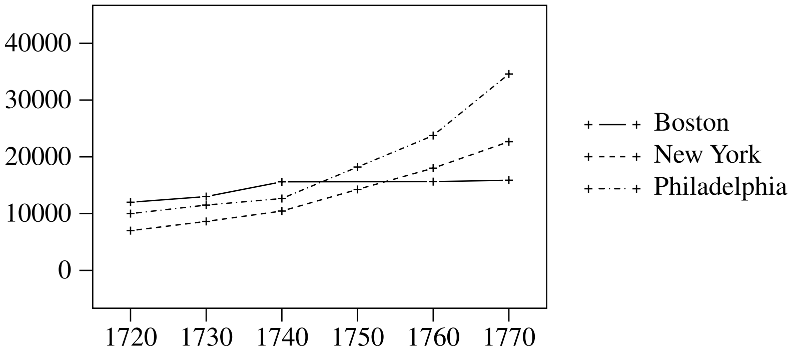
Fertility rates in some developing countries










40000
30000
20000
10000
0
1720
1730
1740
1750
1760
1770
Boston
New York
Philadelphia