CSC 120 (R Section, L0201), Spring 2015 — Assignment #1

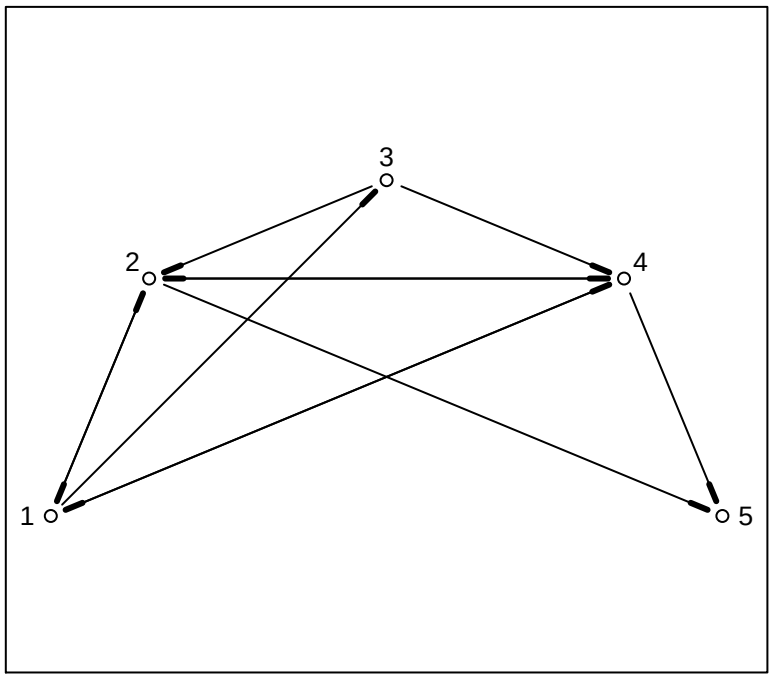
Worth 10% of the course grade. Due by the start of class on February 10, by email (see end of this handout). This assignment may be handed in late, with a 20% penalty, by start of class on February 13. Assignments will not usually be accepted after that. Contact the instructor as soon as possible if you have a legitimate excuse (eg, documented illness) for handing in the assignment late.

This assignment is to be done by each student individually. You may discuss it in general terms with other students, but the work you hand in should be your own. In particular, you shouldn't leave a discussion with someone else with any written notes (either paper or electronic).

Revised version: Corrects how long the thick line is (0.15, not 15% of distance, but you can do it the old way if you like). Also says how to submit the assignment.

The patterns of links from one web site to another are of great interest to search engine companies such as Google. In this assignment, you will write an R program to display the connections amongst a (small) set of web sites. This will provide practice in basic R programming facilities such as `if` and `for`, in R's plotting facilities, and in how to divide a program into several functions that can be understood separately, and perhaps be used in other programs.

Here is an example of a plot such as your program should produce:



This shows the links amongst five web sites, identified by the numbers 1 to 5. A link from web site i to web site j is shown as a line from the circle labeled i to the circle labeled j in which the part of the line near the circle labeled i is thicker than the middle of the line. If two web sites link to each other, the line connecting them will be thick at both ends. For instance, this plot shows that web site 5 links to web site 4, but web site 4 does not link to web site 5. Also web sites 1 and 2 link to each other, and web site 1 does not link to web site 5 and neither does web site 5 link to web site 1.

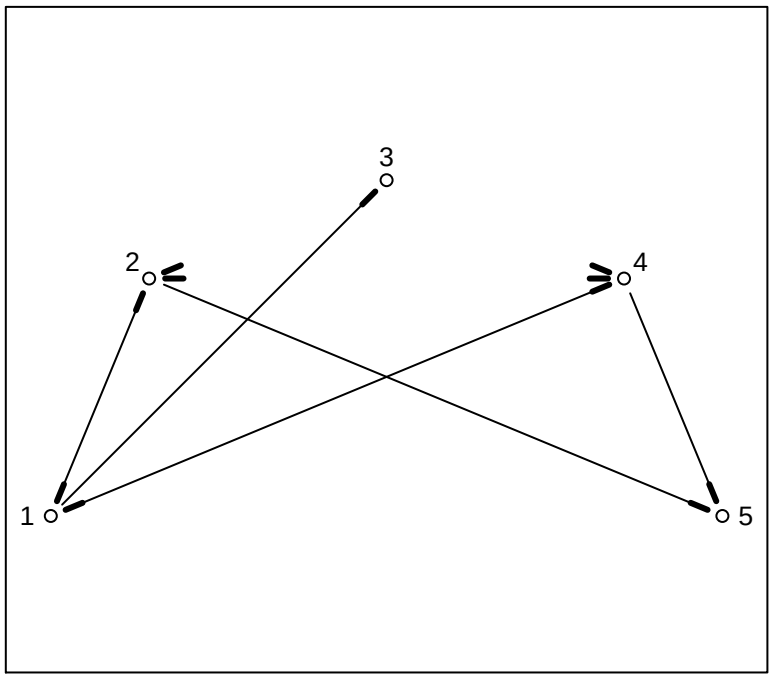
You will get the data displayed in such a plot from a set of files (on the course webpage), one file for each web site. The names of these files will all begin with some string of characters, followed immediately (no space) by the number identifying a web site (starting at 1). The file for web site i will contain the numbers of all the web sites that web site i links to.

For example, the data for producing the plot above came from the files with the following names and contents:

```
slink1    2 4
slink2    1 4 3
slink3    1
slink4    1 3 2
slink5    2 4
```

When the number of web sites is large, the plot may be hard to read, with links from web sites with many links obscuring the links from other web sites. Your program should therefore have an option for suppressing the thin part of a line representing a link from a website with more than a specified number of links to other websites. The initial thick part of the line should still be shown, however.

Here is the plot that would be produced for the above data if links from web sites with more than two links to other sites are not drawn:



Web sites 2 and 4 have more than two links to other web sites, so these links are shown only as the short and thick initial part of the line. Notice that these web sites may still have incoming lines from other web sites that link to them, such as the links from site 5 to site 2 and from site 1 to site 4.

You should write two R functions that can be used in an R script to produce plots such as those above — one for reading link data from files, and one for producing a plot of such data. Their definitions should begin as shown below:

```
read_links <- function (nsites, file)

show_link_data <- function (link_data, max_keep=1000000000)
```

The `read_links` function should read data on `nsites` (a positive integer) web sites from files (or web pages specified by URL) that begin with the string `file`, with numbers from 1 up appended to this string. Each such file will contain numbers identifying the web sites that it links to. Your `read_links` function should check that these numbers are not less than one or greater than `nsites`. If an invalid number is found, you should call the `stop` function passing it the string `"Invalid link"`. This will print this message and then stop your program, returning you to the R console prompt (if you ran the program interactively).

The value returned by `read_links` should be a list of `nsites` numeric vectors, with element i of this list being a vector of the numbers of the web sites linked to by site i (in any order). Note that these vectors will be of varying length, equal to the number of sites linked to.

The `show_data` function should produce a plot, like those above, of the data given by its first argument, which should be a list of the form returned by `read_links`. The second argument of `show_data` should be the number of links to other sites beyond which a web site's links will not be shown as lines (except for the short thick part of the line). This second argument may be omitted in a call, in which case it defaults to the huge number 1000000000 (which is more than the number of links any web site would have).

The plot produced by `show_data` should have the same appearance as those above, with the points representing web sites positioned around a half-circle, and without labels or numbers on the axes. See the week 4 lecture slides for information on how to do this. However, when debugging your program, you might want to leave the numbers on the axes, so you can better see what might be causing any problems.

In order to implement the `show_data` function, you should write four additional functions (which might also be useful for other programs). These functions should have definitions that begin as follows:

```
plot_arc <- function (nsites)

plot_links <- function (link_data, max_keep=1000000000)

thick_start_line (a, b, thin_part)

arc_points <- function (nsites)
```

The `plot_arc` function should create a new plot, and plot circles and numbers representing `nsites` web sites at equally-spaced positions along a half-circle. You will need to use the `text` function described in the week 4 slides to draw the numbers. You should give `text` the argument `cex=0.7` in order to make the numbers smaller so they will fit better.

The `plot_links` function should take arguments like those of `show_link_data`. It will add to the plot created by `plot_arc` by drawing lines between circles representing web sites according to the information in `link_data`, using the `thick_start_line` function. Note that the number of sites will be equal to `length(link_data)`, so it does not have to be provided to `plot_links` as an additional argument.

The `thick_start_line` function should take two vectors of length two as its first two arguments, specifying the coordinates of a start point and an end point. If `thin_part` is `TRUE`, it should draw an ordinary (thin) line from `a` to `b`. Regardless of whether `thin_part` is `TRUE` or `FALSE`, a short thick line (extending a distance of 0.15 from `a` to `b`) should be drawn (on top of the thin line if it is present). Both types of lines should be drawn with the `lines` function, using `type="c"` in order to leave space for the circles drawn by `plot_arc`.

The `arc_points` function should return a list with two vector elements, named `x` and `y`, that give the horizontal and vertical coordinates of `nsites` points on a half-circle centred at the origin with radius one. You should call `arc_points` from both `plot_arc` and `plot_links`.

You should put the definitions of all these functions in a file, which you will submit as part of your assignment. For this assignment, you do not have to include comments before your function definitions that describe what the functions do, since this is completely specified in this assignment handout, but this will be required in future assignments.

You should also create a separate R script file, which should call the `read_links` and `show_link_data` functions in order to display two sets of data from the course web page, and perform other tests as described below. Your script should contain informative but concise comments describing what it does.

The first data set is what is shown in the examples above. It consists of only five web sites. You can read it from files that begin with

```
http://www.cs.utoronto.ca/~radford/csc120/slink
```

The second data set has 35 web sites, and is stored in files that begin with

```
http://www.cs.utoronto.ca/~radford/csc120/blink
```

Note that you have to append numbers 1, 2, etc. to the end of the above to get the full name of the file to read from, using `scan`.

For the first (small) data set, you should produce a plot of all links, and plot of links excluding those from web sites with more than two links. You should also display the data read with `read_links` as a list (ie, as R normally prints it). Finally, you should check that you correctly detect an error when a link is invalid by trying to read data on just the first three web sites in this data set (which should give an error since these refer to web site 4).

For the second (big) data set, you should produce a plot of all links, and a plot of links excluding those from web sites with more than ten links.

You should submit the output (including plots) of your script (with the commands echoed) as part of your assignment. You will do this using the function `knitr::spin`, which creates a

single HTML file as output. You should start your script as follows, in order to set the right options for `knitr::spin`:

```
#+ setup,include=FALSE
source("http://www.cs.utoronto.ca/~radford/csc120/options.r")
#+
```

To you submit your assignment, send an email to `radford@cdf.utoronto.ca`, with subject line "A1 your-family-name, your-given-name". The body of the email can be blank (but you can include a note if you like). You should attach two files. The first should be the .R file containing your function definitions. The second should be the .html file created by `knitr::spin` when you ran your script.