Reproducible, flexible and high throughput data extraction from primary literature: The metaDigitise R package

Joel L. Pick[1,*], Shinichi Nakagawa[1], Daniel W.A. Noble[1]

[1] Ecology and Evolution Research Centre, School of Biological, Earth and Environmental Sciences, University of New South Wales, Kensington, NSW 2052, Sydney, AUSTRALIA

*Corresponding Author: joel.l.pick@gmail.com

Abstract

Research synthesis, especially in the form of meta-analysis, requires data extraction from primary studies. Meta-analysis synthesizes effect sizes, often calculated from summary statistics of studies. However, exact values of such statistics are commonly hidden in figures. The R package **metaDigitise** extracts descriptive statistics such as means, standard deviations and, if applicable, correlations from the four types of plots: 1) mean/error plots (e.g. bar graphs with standard errors), 2) box plots, 3) scatter plots and 4) histograms. The package interactively guides the user through data extraction process. Notably, it enables a large-scale extraction using image files, letting the user stop processing, edit and add to the resulting data fame at any point. Further, it facilitates reproducible data extraction from plots with little inter-observer bias, thus, allowing a group of people to participate the extraction of data collaboratively.

Keywords: meta-analysis, comparative analysis, data extraction, R, reproducibility, figures, images, summary statistics

1 Introduction

In many different contexts, researchers need to make use of data presented in primary literature. Most notably, this includes meta-analysis, which is becoming increasingly common in many research fields. Meta-analysis uses effect size estimates and their sampling variance, taken from many studies, to understand whether particular effects are common across studies and to explain variation among these effects (Glass, 1976; Borenstein et al., 2009; Koricheva, Gurevitch & Mengersen, 2013; Nakagawa et al., 2017). Meta-analysis therefore relies foremost on data extracted from primary literature, and more specifically, descriptive statistics (e.g. means, standard deviations, sample sizes, correlation coefficients) that have been reported in the text or tables of research papers. Descriptive statistics are also, however, frequently presented in figures and so need to be manually extracted using digitising programs. While inferential statistics (e.g., t- and F-statistics) are often presented along side descriptive statistics and can be used to derive effect sizes, descriptive statistics are much more appropriate to use because sources of non-independence in experimental designs can be dealt with more easily (Noble et al., 2017). Although there are several existing tools to perform tasks like this (e.g. DataThief (Tummers, 2006), GraphClick (Arizona-Software, 2008), WebPlotDigitizer (Rohatgi, 2017)), these tools are not appropriate for the needs of meta-analysis for three main reasons.

First, they typically only provide the user with calibrated x,y coordinates from imported figures, and do not differentiate between common plot types that are used to present data. This means that a large amount of downstream data manipulation is required, that is different across plots types. For example, data are frequently presented in mean/error plots (Figure 1A), for which the user wants a mean and error estimate for each group presented in the figure. With existing programs, x,y coordinates of means and errors are returned, to which the user must manually discern between mean and error coordinates and assign points to groups. The error then needs to be calculated as the deviation from the mean, and then transformed to a standard deviation, depending on the type of error presented. Second, digitising programs do not easily allow the integration of metadata at the time of data extraction, such as experimental group or variable names, and sample sizes. This makes the downstream calculations more laborious, as the information has to be added later, in most cases using different software. Finally, existing programs do not import a set of images and allow the user to systematically work through them. Instead they require the user to manually import images one by one, and export data into individual files, that need to be imported and edited using different software. In essence, existing software does not provide an optimized research pipeline to facilitate data extraction, editing and reproducibility.

These present major issues because extracting from figures can be an incredibly time-consuming process. Furthermore, although meta-analysis is an important tool in consolidating the data from multiple studies, many of the processes involved in data extraction are opaque and difficult to reproduce, making extending studies problematic. Having a tool that facilitates reproducibility in meta-analyses will increase transparency and aid in resolving the reproducibility crises seen in many fields (Peng, Dominici & Zeger, 2006; Peng, 2011; Sandve et al., 2013; Parker et al., 2016; Ihle et al., 2017).

Here, we present an interactive R package, **metaDigitise** (available at xxx), which is designed for large scale, reproducible data extraction from figures, specifically catering to the the needs of meta-analysts. To this end, we provide tools specific to data extraction from common plot types (mean/error plots, box plots, scatter plots and histograms, see Figure 1). **metaDigitise** operates within the R environment making data extraction, analysis and export more streamlined. The necessary calculations are carried out on processed data immediately after extraction so that comparable summary statistics can be obtained quickly. Summary data from multiple figures is returned into a single data frame which can be can easily exported or use in downstream analysis within R. Calibrated data is automatically saved for each digitised figure. Users can therefore redraw their digitisations on figures, make corrections and access calibration and proceeded data. This makes sharing figure digitisation and reproducing the work of others simple and easy, and allows meta-analysts to update meta-analyses more efficiently.

2 Directory Structure and Reproducibility

The **metaDigitise** package was created with the idea that users would have multiple images to extract from and therefore operates in the same way whether the user has one or multiple images. There is one main function, `metaDigitise()`, which interactively takes the user through the process of extracting data from figures. `metaDigitise()` works on a directory containing images of figures copied from primary literature, in .png, .jpg, .tiff, .pdf format, specified to `metaDigitise()` through the `dir` argument. The user needs to think carefully about their directory structure early on in their project, especially if they plan to share the extractions with collaborators or publish the project. Although different directory structures may be used, we would recommend having all files for one project in a single directory with an informative and unambiguous naming scheme for images to help identify the paper and figure that data come from, for example:

```
* Main project directory
    + FiguresToExtract
        + Paper1_Figure1_trait1.png
        + Paper1_Figure2_trait2.png
        + Paper2_Figure1_trait3.png
```

When `metaDigitise()` is run, it recognizes all the images in a directory and automatically imports them one by one, allowing the user to extract the relevant information about a figure as they go. Having all figures in one directory therefore expedites digitising figures by preventing users from having to constantly change directories and / or open new images. The data from each completed image is automatically saved as a `metaDigitise` object in a separate .RDS file to a `caldat` directory that is created within the parent directory when first executing `metaDigitise()`. These files enable re-plotting and editing of images at a later point (see below).

`metaDigitise()` also identifies images within a directory that have been previously digitised and only import images that have not been digitised in previous calls of the function. All figures do not, therefore, need to be extracted at one time and new figures can be added to the directory as the project develops. After each image is extracted, the user is asked whether they wish to continue or quit the extraction process. Upon rerunning `metaDigitise()`, previously digitised figures are ignored during processing, but their data is automatically integrated into the final output.

This directory structure allows the complete digitisation process to be reproduced at a later stage, shared with collaborators and presented as supplementary materials for a publication. As long as all the images and the caldat directory are still in the directory, `metaDigitise()` will be able to reproduce all figure extractions, regardless of the computer it is run on. For an analysis to be updated, new figures can simply be added to the directory and `metaDigitise()` run to incorporate the new data.

3 Image Processing

Running `metaDigitise()` presents the user with three options; 'Process new images', 'Import existing data' or 'Edit existing data'. Selecting 'Process New Images' starts the digitisation process on images within the directory that have not preciously been digitised; the other functions are discussed below.

For all plot types, `metaDigitise()` requires the user to calibrate the axes in the figure, by clicking on two known points on the axis in question, and entering the value of those points (Figure 1). `metaDigitise()` then calculates the value of any clicked points in terms of the figure axes. For mean/error and box plots, only the y-axis is calibrated (Figure 1A,B), assuming the x-axis is redundant. For scatter plots and histograms both axes are calibrated (Figure 1C,D).

As figures may have been copied from older, scanned publications, they may not be perfectly orientated. This makes calibration of the points in the figure problematic. `metaDigitise()` allows users to rotate the image (Figure 2A,B). Furthermore, mean/error plots, box plots and histograms, may be presented with horizontal bars. `metaDigitise()` assumes that bars are vertical, but allows the user to flip the image to make the bars are vertical (Figure 2C,D).

metaDigitise recognises four main types of plot; Mean/error plots, box plots, scatter plots and histograms, shown in Figure 1. All plot types can be extracted in a single call of `metaDigitise()` and integrated into one output. Alternatively, users can process different plot types separately, using separate directories. All four plot type are extracted slightly differently (outlined below). After completing all images, or upon quitting, either summarised or processed data is returned (specified by the user). From all plot types, summarised data consists of a mean, standard deviation and sample size, for each identified group within the plot (should multiple groups exist). In the case of scatter plots, the correlation coefficient between the points within each identified group is also returned. Alternatively choosing processed data will return a list with a slots for each of the four figure types, containing the calibrated points that the user has clicked. This may be particularly useful in the case of scatter plots.

3.1 Mean/Error and Box Plots

`metaDigitise()` handles mean/error and box plots in a very similar way. For each mean/box, the user is enters group names and sample sizes. If the user does not enter a sample size at the time of data extraction (if, for example, the information is not readily available) a standard deviation (SD) is not calculated. This can, however, be entered at a later time (see below). For mean/error plots, the user clicks on an error bar and the mean. Error bars above or below the mean can be clicked, as sometimes one is clearer than the other. `metaDigitise()` assumes that the error bars are symmetrical. This is deliberate as it is not clear how best to derive SD from asymmetrical error bars, not least as they represent different things in different figures. Points are displayed where the user has clicked, with the error in a different colour to the mean (Figure 1A). The user also enters the type of error used in the figure: standard deviation (SD), standard error (SE) or 95% confidence intervals (CI95). For box plots, the user clicks on the maximum, upper quartile, median, lower quartile and minimum. `metaDigitise()` will return a warning if the maximum is not greater than the minimum. For both plot types, the user can add, edit or remove groups. Three functions, `error_to_sd()`, `rqm_to_mean()` and `rqm_to_sd()`, that convert different error types to SD, box plot data to mean and box plot data SD, respectively, are also available in the package (see SM for further details of these conversions).

3.2 Scatter plots

Users can extract points from multiple groups from scatter plots. Points added by mistake can be deleted. The user can add more groups, edit groups (add or remove points) or delete groups. Different groups are plotted in different colours and shapes to enable them to be distinguished, with a legend at the bottom of the figure (Figure 1C). Mean, SD and sample size are calculated from the clicked points, for each group. Often data points will overlap with each other making it impossible to know whether points have been missed. However, a user may realise that the sample size from the digitisation conflicts with what is reported in the paper. For example, in Figure 1C only 49 points have been clicked when the sample size is known to be 50. Hence, **metaDigitise** also provides the user with the option to input a known sample sizes directly. Nonetheless, it is important to recognise the impact that overlapping points can have on summary statistics, and in particular on sampling variance.

3.3 Histograms

The user clicks on the top corners of each bar. A line is drawn, in alternate colours, at the top of these bars (Figure 1D). These are numbered to allow the the user to edit them. As with scatter plots, if the sample size from the extracted data does not match a known sample size, the user can enter an alternate sample size. The calculation of mean and SD from this data is shown in the SM.

4 Importing and Editing Previously Digitised data

metaDigitise is also able to re-import, edit and re-plot previously digitised figures. When running `metaDigitise()`, the user can choose to 'Import existing data', which returns previously digitised data, for single or all images. Users can also choose to 'Edit existing data' which allows them to re-plot or edit information or digitisations that have previously be done.

4.1 Adding Sample Sizes to Previous Digitisations

In many cases important information, such as sample sizes, may not be readily available when digitising figures. Such information does not, therefore, have to be added a the time of digitisation. To expedite finding and adding these sample sizes at a later point, `metaDigitise()` has a specific edit option that allows users to enter previously omitted sample sizes. This first identifies missing sample sizes in the digitised output, re-plots the relevant figures and prompts the user to enter the sample sizes for the relevant groups in the figure, one by one.

5 Software Validation

5.1 Inter-observer variability in digitisations

In order to evaluate the consistency of digitisation using **metaDigitise** between users, we simulated a dataset of two variables with two groups ($n = 10$ within groups). Each variable was plotted twice for each plot type (figures were modified slightly to give users a sense that they were digitising new data) generating a total of 14 figures. 14 independent digitisers (including the authors) were provided with a directory with all 14 figures in a randomised order. Digitisers ran **metaDigistise** on their own computers, across different operating systems (including Mac, Windows and Linux). Digitisers varied in their level of experience, from people with experience of meta-analyses or comparative work to those without any science background. We asked users to digitise all 14 figures and collected the mean, standard deviation and correlation coefficient (r, for scatter plots) generated by `metaDigitise()` for every plot digitised ($n = 28$ per digitiser per metric, $n = 4$ for r).

As a measure of bias, we calculated the percentage differences from the true summary statistics as

$$\frac{\theta - \hat{\theta}}{\hat{\theta}} \tag{1}$$

where θ is the estimate and $\hat{\theta}$ is the true value. The deviation from the true value of r was not further standardised, as it is already on a standardised scale. We also took the absolute values of these standardised differences as a measure of precision. The resulting data was used to assess between- and within- user variability (i.e., the intra-class correlation coefficient - ICC). This was done using linear mixed effect models with user identify as a random effect using **lme4** (Bates et al., 2015) in R. Standardised mean, standard deviation and correlation coefficients were used as response variables in separate models. Sampling variance for ICC estimates was generated based on 1000 parametric bootstraps of the model and the significance was tested using likelihood ratio tests, using **rptR** (Stoffel, Nakagawa & Schielzeth, 2017).

If digitisations were consistent across users then we should find no significant between user variability in the data. Indeed, across plot types we found no evidence for any inter-observer variability in digitisations for the mean (ICC = 0, 95% CI = 0 to 0.029, p = 1), standard deviation (ICC = 0, 95% CI = 0 to 0.033, p = 0.5) or correlation coefficient (ICC = 0.053, 95% CI = 0 to 0.296, p = 0.377). There were was little bias between digitised and true values, on average 1.63% (mean = 0.02%, SD = 4.9%, r = -0.03%) and overall there were only small absolute differences between digitised and true values, deviating, on average 2.18% (mean = 0.40%, SD = 5.81%, r = 0.33%) across all three summary statistics. SD estimates from digitisations are clearly more prone to error than means or correlation coefficients. If the mean absolute difference is calculated for each plot type, we can see that this effect is driven mainly by extraction from box plots and histograms (% difference; box plot: 15.805, histogram: 5.210, mean/error: 1.500, scatter plot: 0.433). SD estimation from box plot summary statistics is known to be more error prone, especially at small sample sizes (Wan et al., 2014).

5.2 Testing the accuracy of digitisations

To test how accurate **metaDigitise** is at matching clicked points to their true values, we generated four random scatterplots, each with 20 data points, and digitised these with `metaDigitise()`. This was done by one digitiser (J.L.P.), as there is no detectable between user variation. Data digitised using **metaDigitise** was essentially perfectly correlated with the true simulated data for both the x-variable (Pearson's correlation; $r = 0.9999915$, $t = 2137.4$, $df = 78$, $p < 0.001$) and y-variable ($r = 0.9999892$, $t = 1897.8$, $df = 78$, $p < 0.001$).

6 Limitations and Future Extensions

Although **metaDigitise** is already very flexible, and provides functionality not seen in any other package (Table S1), it is clear that there are some functions that it does not perform. Notably **metaDigitise** lacks is automated point detection, available in several packages (Table S1). However, from our experience, manual digitising is more reliable and often equally as fast. Given the variation in image quality, calibration for automatic point detection needs to be done for each plot individually. Additionally, auto-detection often misses points which then need to be manually added. Based on tests of **metaDigitise** (see above), figures can be extracted in around 1-2 minutes, including the entry of metadata. As a result, we do not believe that current automated point detection techniques provides substantial benefits in terms of time or accuracy.

metaDigitise also (currently) lacks the ability to zoom in on plots. Zooming may enable users to gain greater accuracy when clicking on points. However, from our own experience (and indeed from the results above), if you are using a reasonably sized screen then the accuracy is already high, and so there is not much gain to be had from zooming in on points.

In contrast to some other packages, **metaDigitise** does not extract lines from figures. In our own experience, line extraction is not particularly useful for meta-analysis, although we recognise that it may be useful in other fields. Should a user like to extract lines with **metaDigitise**, we would recommend extracting data as a scatter plot, and clicking along the line in question. A model can then be fitted to these points (accessed by choosing to return processed rather than summary data) to estimate the parameters needed.

Descriptive statistics are usually the most robust sources of information for calculating effect size statistics (Noble et al., 2017). These are most often presented in figures. Users may therefore also want to compare effect size estimates from inferential statistics with those derived from descriptive statistics (obtained for example using **metaDigitise**) from a paper. Comparing these different effects sizes can be useful in identifying uncertainties and problems within a paper. In the future, we hope to provide functions to easily convert inferential statistics to standardised effect size estimates, which can seamlessly be integrated with summary statistics from **metaDigitise**, to calculate equivalent standardised effect size estimates and sampling variance.

7 Conclusions

Increasing the reproducibility of figure extraction for meta-analysis and making this laborious process more streamlined, flexible and integrated with existing statistical software will go a long way in facilitating the production of high quality meta-analytic studies that can be updated in the future. We believe that **metaDigitise** will improve this research synthesis pipeline, and will hopefully become an integral package that can be added to the meta-analysts toolkit.

Acknowledgments

We thank the I-DEEL group and colleagues at UNSW for for testing, providing feedback and digitising including: Rose O'Dea, Fonti Kar, Malgorzata Lagisz, Julia Riley, Diego Barneche, Erin Macartney, Ivan Beltran, Gihan Samarasinghe, Dax Kellie, Jonathan Noble, Yian Noble and Alison Pick. J.L.P. was supported by a Swiss National Science Foundation Early Mobility grant (P2ZHP3_164962), D.W.A.N. was supported by an Australian Research Council Discovery Early Career Research Award (DE150101774) and UNSW Vice Chancellors Fellowship and S.N. an Australian Research Council Future Fellowship (FT130100268).

Author Contributions

J.L.P. and D.W.A.N. conceived the study and J.L.P., S.N. and D.W.A.N. developed the idea. J.L.P. and D.W.A.N. developed the R-package. J.L.P. and D.W.A.N. wrote the first draft of the paper and J.L.P., S.N. and D.W.A.N. contributed substantially to subsequent revisions of the manuscript and gave final approval for publication.

References

Arizona-Software (2008) *GraphClick Software, Version 3.0.*

Bates, D., Maechler, M., Bolker, B. & Walker, S. (2015) Fitting Linear Mixed-Effects Models Using lme4. *Journal of Statistical Software*, **67**, 1–48.

Borenstein, M., Hedges, L., Higgens, J. & Rothstein, H. (2009) Introduction to meta-analysis. *John Wily Sons. Ltd. West Sussex, UK.*

Glass, G. (1976) Primary, secondary, and meta-analysis research. *Educational Researcher*, **5**, 3–8.

Ihle, M., Winney, I.S., Krystalli, A. & Croucher, M. (2017) Striving for transparent and credible research: practical guidelines for behavioral ecologists. *Behavioral Ecology*, **28**, 348–354.

Koricheva, J., Gurevitch, J. & Mengersen, K. (2013) Handbook of Meta-Analysis in Ecology and Evolution. *Princeton University Press, Princeton, New Jersey.*

Nakagawa, S., Noble, D.W., Senior, A.M. & Lagisz, M. (2017) Meta-evaluation of meta-analysis: ten appraisal questions for biologists. *BMC Biology*, **15**, 18; DOI 10.1186/s12915–017–0357–7.

Noble, D.W., Lagisz, M., O'Dea, R.E. & Nakagawa, S. (2017) Nonindependence and sensitivity analyses in ecological and evolutionary meta-analyses. *Molecular Ecology*, **26**, 2410–2425.

Parker, T.H., Forstmeier, W., Koricheva, J., Fidler, F., Hadfield, J., En Chee, Y., Kelly, C.D., Gurevitch, J. & Nakagawa, S. (2016) Transparency in Ecology and Evolution: Real Problems, Real Solutions. *Trends in Ecology and Evolution*, **31**, 711–719.

Peng, R.D. (2011) Reproducible research in computational science. *Science*, **334**, 1226.

Peng, R.D., Dominici, F. & Zeger, S.L. (2006) Reproducible epidemiologic research. *American Journal of Epidemiology*, **163**, 783–789.

Rohatgi, A. (2017) *WebPlotDigitizer Software, Version 4.0.* Austin, Texas, USA.

Sandve, G.K., Nekrutenko, A., Taylor, J. & Hovig, E. (2013) Ten simple rules for reproducible computational research. *PLoS Computational Biology*, **9**, e1003285.

Stoffel, M.A., Nakagawa, S. & Schielzeth, H. (2017) rptR: repeatability estimation and variance decomposition by generalized linear mixed-effects models. *Methods in Ecology and Evolution*, **8**, 1639–1644.

Tummers, B. (2006) *DataThief Software, Version 3.0.*

Wan, X., Wang, W., Liu, J. & Tong, T. (2014) Estimating the sample mean and standard deviation from the sample size, median, range and/or interquartile range. *BMC Medical Research Methodology*, **14**, 135.

Figures

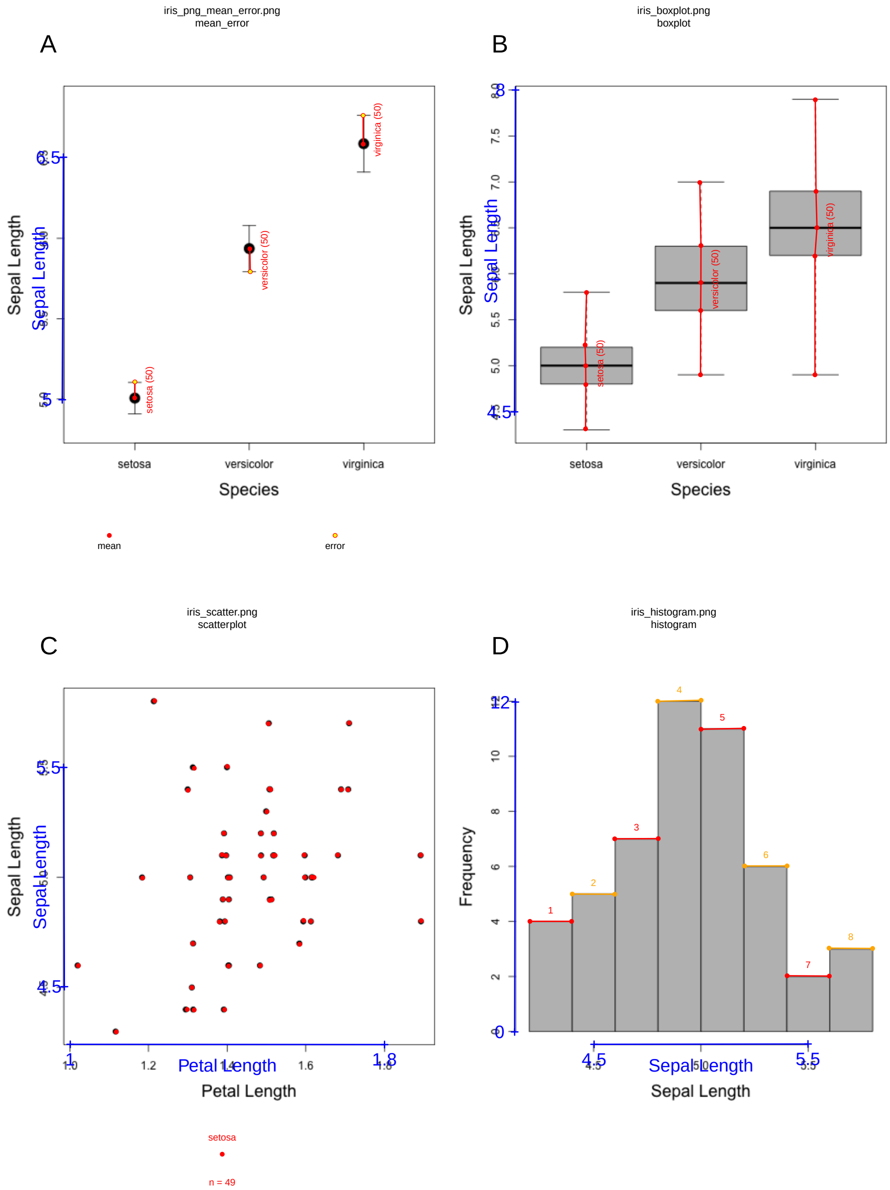

Figure 1: Four plot types that **metaDigitise** is designed to extract data from: A) mean/error plot, B) box plot, C) scatter plot and D) histogram. Data is taken from the iris dataset in R. A and B are plotted with the whole dataset, C and D are just the data for the species *setosa*. Digitisation of the images is shown. All figures are clearly labelled at the top to remind users of the filename and plot type. This reduces errors throughout the digitisation process. Names of the variables and calibration (in blue) are plotted alongside the digitised points. In A) and B), user entered group names and sample sizes are displayed beside the relevant points. In C) the names and sample sizes for each group are shown below the figure.

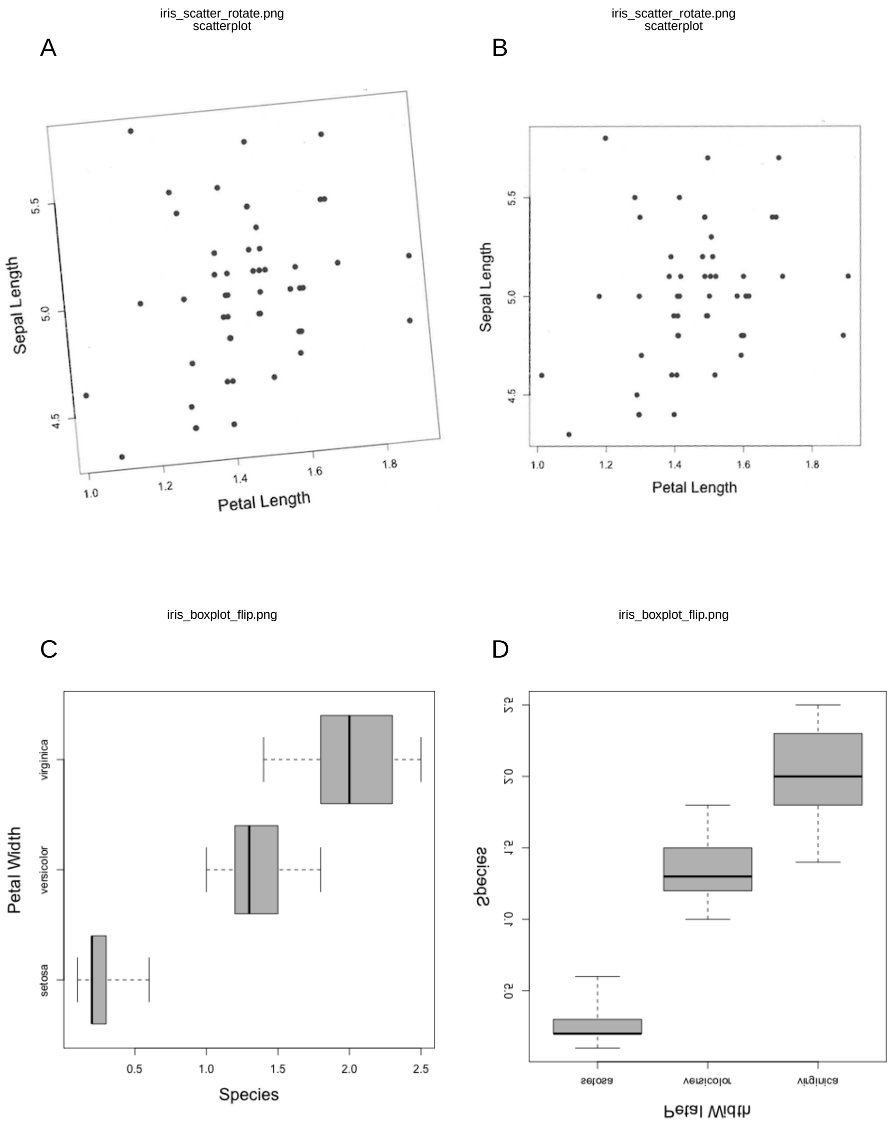

Figure 2: Figure rotation. A) and B) show how non-aligned images can be realigned through user defined rotation. C) and D) show how figures can be re-orientated so as to aid data input.