

Reproducible Research Dynamic Document Example

Andy Hector & Yvonne Buckley

i I. Background

This document is intended to provide a simple example of using Quarto and R (via RStudio) to produce a reproducible research dynamic document. Reproducible research documents combine input code, software (R) generated output and author-written text to explain the analysis so that it can be understood and repeated by others. *Journal of Ecology* now encourages the submission of dynamic documents for reproducible research as supplementary information (see the *Journal of Ecology* instructions to authors). Quarto is the successor to the similar R markdown software but also works with Python and other languages (alternatives like Jupyter notebooks can also be used). The Quarto and R Markdown software is available automatically when using RStudio.

i II. Setting document-wide options

Document-wide Quarto options are set by a the ‘YAML header’ at the start of a Qmd file. Document-wide options for individual packages can also be set at the start of the Qmd file by a ‘chunk’ of R code. The first chunk of R code (below) sets these document wide package options. These options are largely cosmetic choices that could be omitted (the default behaviour will then be applied) and are not intended to distract new users of Quarto who may want to skip over this chunk. The code sets the options to: Clear R’s workspace; Set the text width; Turn off significance stars in the output; Set the black and white theme for ggplot2 package graphics; Set the figure size; Tidy chunk text; and set comment output symbols). By default, a HTML document will be produced when the Quarto document is rendered—to produce a PDF file you will need to install some additional software such as the tinytex R package.

```
rm(list = ls(all.names = TRUE)) # clear workspace
options(width = 58) # text width
```

```
options(show.signif.stars = FALSE) # R text options
library(ggplot2) # ggplot2 package, set options
theme_set(theme_bw()) # ggplot2 black and white figure theme
library(knitr) # knitr package, to set options
opts_chunk$set(fig.width = 4.5, fig.height = 2.5) # fig dimensions
opts_chunk$set(tidy = TRUE, tidy.opts = list(width.cutoff = 54)) # tidy code
opts_chunk$set(comment = "#>") # output prefix
```

Linear model ANOVA of Darwin's maize data

This document presents a simple linear model analysis of a data set collected by Charles Darwin.

1 Packages, data and exploratory graphics

1.1 R packages:

The following R packages are used (they must be installed before they can be used—see the RStudio Tools menu):

```
library(arm)
library(marginaleffects)
library(SMPracticals)
library(tidyverse, quietly = TRUE)
```

i III. Setting the working directory

Putting the Quarto (Qmd) file and data file(s) together in the same directory automatically sets that location as the working directory. Alternatively, the working directory should be set by each analyst for their computer via the menu options (not with R code that will only work for the machine it was written on).

1.2 Darwin's maize pollination data

In *The effects of cross and self-fertilization in the vegetable kingdom* Darwin (1876) describes how he produced seeds of maize (*Zea mays*) that were fertilized with pollen from the same individual or from a different plant. Pairs of seeds taken from self-fertilized and cross-pollinated plants were then grown in pots and the height of the seedlings measured as a surrogate for

their evolutionary fitness. Darwin wanted to know whether inbreeding reduced the fitness of the selfed plants.

1.2.1 Experimental design

Darwin's work pre-dates the development of formal experimental design in the early twentieth century and therefore has some shortcomings that we ignore in this example. For example, randomisation was not used. Darwin's experiment compares 15 maize seedlings grown from seeds from a self-pollinated mother plant with 15 seedlings grown from seeds from a cross-pollinated mother plant. Pairs of seedlings from cross- and self-pollinated seeds were planted in pots. Ideally, one pair of plants would have been grown in each of 15 pots. In practice a variable numbers of pairs (3, 4 or 5) were planted in only four pots. For simplicity, in this example we ignore the pairing and analyse the data as if it were a modern fully-randomized design. We use the `lm()` function to compare the heights of seedlings from self or cross-pollinated seeds using linear model ANOVA.

The data are available from the `SMPracticals` R package:

```
darwin # library(SMPracticals)
```

Convert `pot`, `pair` and `type` to factors:

```
darwin$pot <- factor(darwin$pot)
darwin$pair <- factor(darwin$pair)
darwin$type <- factor(darwin$type)
```

No transformation of the response variable is necessary to meet the assumptions of the linear model (see below: Assumption checking).

The `glimpse` function gives a précis of the dataset:

```
glimpse(darwin) # library(tidyverse)
```

```
#> Rows: 30
#> Columns: 4
#> $ pot      <fct> I, I, I, I, I, I, II, II, II, II, II, II,~
#> $ pair     <fct> 1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7,~
#> $ type     <fct> Cross, Self, Cross, Self, Cross, Self, Cr~
#> $ height   <dbl> 23.500, 17.375, 12.000, 20.375, 21.000, 2~
```

A summary of the dataframe shows that the design is balanced with 15 pairs of cross- and self-pollinated plants; The response variable biomass has no zeros or missing values:

```
summary(darwin)
```

```
#>   pot      pair      type      height
#>  I   : 6      1       : 2   Cross:15   Min.    :12.00
#>  II  : 6      2       : 2   Self :15   1st Qu.:17.53
#>  III:10      3       : 2                Median :18.88
#>  IV  : 8      4       : 2                Mean   :18.88
#>                5       : 2                3rd Qu.:21.38
#>                6       : 2                Max.   :23.50
#>                (Other):18
```

i IV. Dynamic documents

Reproducible research documents are dynamic. For example, if the initial stages reveal errors in the data these can be corrected as part of the code and the changes will be applied when the file is run in future.

A graph (using the ggplot2 package) of the raw data:

```
xlabel <- expression(paste("Pollination treatment"))
ylabel <- expression(paste("Height (decimal inches)"))
```

```
Fig_1 <- ggplot(data = darwin, aes(x = type, y = height)) +
  geom_point() + labs(x = xlabel, y = ylabel)
Fig_1
```

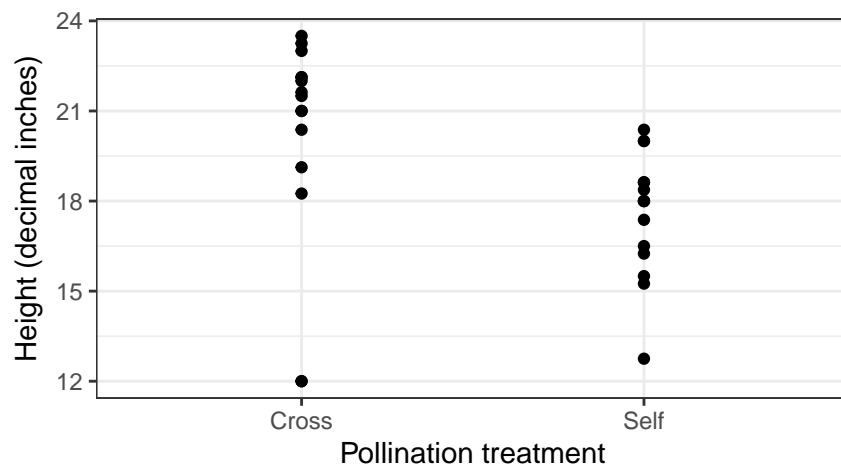


Figure 1 Plant height (decimal inches) as a function of the cross- and self-pollinated treatments.

Analysis

2.1 Linear model

Linear model ('one-way ANOVA') analysing biomass in relation to the fixed treatment factor :

```
model_1 <- lm(height ~ type, data = darwin)
```

The coefficients with their standard errors:

```
display(model_1) # library(arm)
```

```
#> lm(formula = height ~ type, data = darwin)
#>               coef.est coef.se
#> (Intercept)  20.19      0.76
#> typeSelf    -2.62      1.07
#> ---
#> n = 30, k = 2
#> residual sd = 2.94, R-Squared = 0.18
```

The report of the number of observations, n, and number of parameters, k, confirms the intended model has been fitted.

The output gives the coefficients and standard errors for the mean for the cross-pollinated treatment ('Intercept') and the difference between treatment means ('typeSelf'). The design is balanced with no missing values so the two treatment means have the same standard error. Seedlings from cross pollinated parent plants are 20.19 inches tall on average. The seedlings from self-pollinated parent plants are 2.62 inches shorter.

The confidence intervals for these point estimates are:

```
confint(model_1, oldNames = FALSE)
```

```
#>               2.5 %      97.5 %
#> (Intercept) 18.63651 21.7468231
#> typeSelf    -4.81599 -0.4173433
```

The two means with their standard standard errors and 95% confidence intervals (using the marginal effects package):

```
predictions(model_1, by = "type") # library(marginaleffects)

#>
#>   type Estimate Std. Error    z Pr(>|z|)    S 2.5 %
#> Cross     20.2     0.759 26.6 <0.001 515.3 18.7
#> Self      17.6     0.759 23.1 <0.001 391.4 16.1
#> 97.5 %
#>    21.7
#>    19.1
#>
#> Columns: rowid, type, estimate, std.error, statistic, p.value, s.value, conf.low, conf.high
#> Type: response
```

A graph of the two means and 95% confidence intervals:

```
plot_predictions(model_1, by = "type")
```

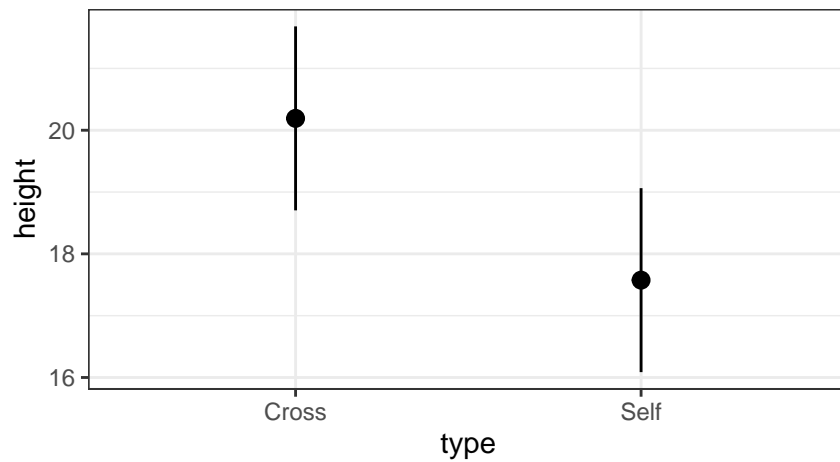


Figure 2 Mean heights (decimal inches) of seedlings from cross- and self-fertilized seeds with 95% confidence intervals.

Differences in means

Confidence intervals (using the arm package `coefplot()` function) for the difference in height:

```
coefplot(model_1, xlim = c(-5, 0)) # library(arm)
```

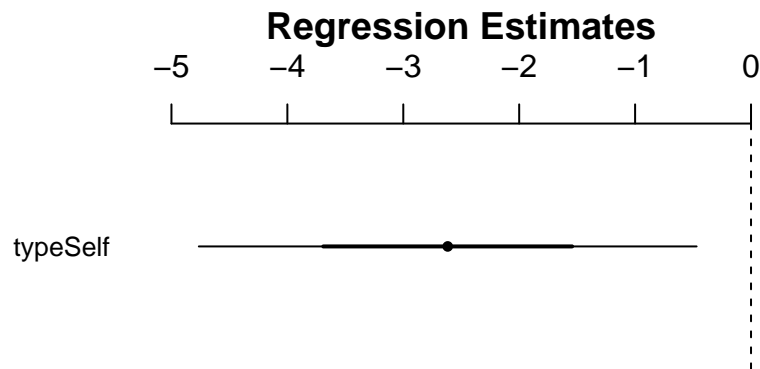


Figure 3 Coefficient plot of the difference in treatment means ± 1 and 2 standard errors of the difference (SED).

3. Assumption checking

The residuals show no indication of notable violations of the assumptions of homogeneity of variance or normality:

```
plot(model_1, which = 1)
```

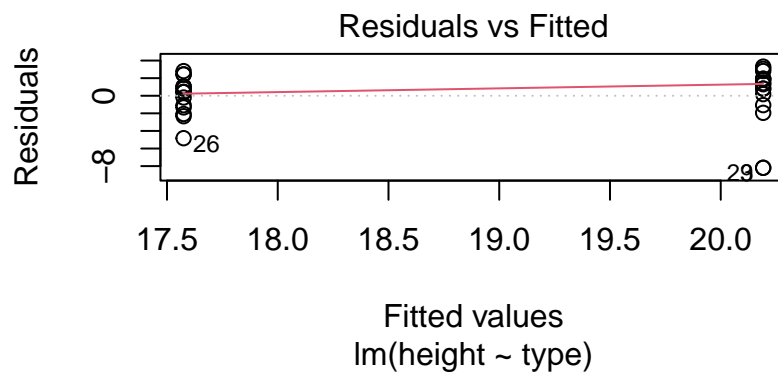


Figure 3a Diagnostic plot of Pearson residuals versus fitted values.

```
qqnorm(resid(model_1))  
qqline(resid(model_1))
```

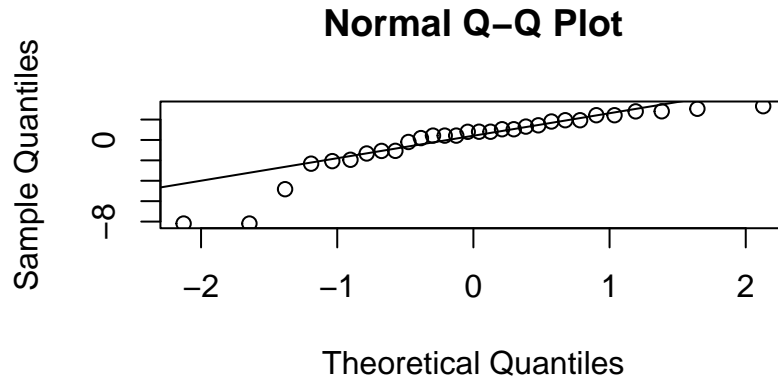


Figure 3b Diagnostic plot of the quantiles of the observed residuals versus the theoretical quantiles.

Based on these diagnostic plots no transformations were applied to the response variable.

4. Results and conclusions

- There are statistically detectable effects of the pollination treatment on seedling height.
- Self pollination is associated with a decrease in height of the resulting seedlings.

The main manuscript text, table and figures report the treatment means and differences between means together with confidence intervals (as given above).

5. Software versions

4. Recording software choices

To reproduce an analysis it may be necessary to know the versions of the software used, especially with modern analytical methods that are still in development. An easy, basic way to do this is with the `sessionInfo()` function:

```
sessionInfo()
```

```
#> R version 4.4.1 (2024-06-14)
#> Platform: aarch64-apple-darwin20
#> Running under: macOS 15.0.1
#>
#> Matrix products: default
#> BLAS:   /Library/Frameworks/R.framework/Versions/4.4-arm64/Resources/lib/libRblas.0.dylib
#> LAPACK: /Library/Frameworks/R.framework/Versions/4.4-arm64/Resources/lib/libRlapack.dylib
#>
#> locale:
#> [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
#>
#> time zone: Europe/London
#> tzcode source: internal
#>
#> attached base packages:
#> [1] stats      graphics  grDevices  utils      datasets
#> [6] methods    base
#>
#> other attached packages:
#> [1] lubridate_1.9.3      forcats_1.0.0
#> [3] stringr_1.5.1        dplyr_1.1.4
#> [5] purrr_1.0.2          readr_2.1.5
#> [7] tidyr_1.3.1          tibble_3.2.1
#> [9] tidyverse_2.0.0      SMPracticals_1.4-3.1
#> [11] ellipse_0.5.0        marginaleffects_0.20.1
#> [13] arm_1.14-4           lme4_1.1-35.3
#> [15] Matrix_1.7-0         MASS_7.3-60.2
#> [17] knitr_1.46           ggplot2_3.5.1
#>
#> loaded via a namespace (and not attached):
#> [1] utf8_1.2.4           generics_0.1.3       stringi_1.8.4
```

```

#> [4] lattice_0.22-6      hms_1.1.3           digest_0.6.35
#> [7] magrittr_2.0.3       timechange_0.3.0    evaluate_0.23
#> [10] grid_4.4.1          fastmap_1.2.0       jsonlite_1.8.8
#> [13] backports_1.5.0     survival_3.6-4      formatR_1.14
#> [16] fansi_1.0.6         scales_1.3.0        abind_1.4-5
#> [19] cli_3.6.2           rlang_1.1.3         munsell_0.5.1
#> [22] splines_4.4.1       withr_3.0.0         yaml_2.3.8
#> [25] tools_4.4.1         tzdb_0.4.0          checkmate_2.3.1
#> [28] nloptr_2.0.3        coda_0.19-4.1       minqa_1.2.7
#> [31] colorspace_2.1-0    boot_1.3-30         vctrs_0.6.5
#> [34] R6_2.5.1            lifecycle_1.0.4     insight_0.19.11
#> [37] pkgconfig_2.0.3     pillar_1.9.0        gtable_0.3.5
#> [40] glue_1.7.0          data.table_1.15.4   Rcpp_1.0.12
#> [43] xfun_0.44           tidyselect_1.2.1    rstudioapi_0.16.0
#> [46] farver_2.1.2        htmltools_0.5.8.1   nlme_3.1-164
#> [49] labeling_0.4.3      rmarkdown_2.27      compiler_4.4.1

```