A Minimal Book Example for BIOL8700

Rachael Remington; Anthony Davidson

2022-05-24

Contents

Resources for class 2022 and future BIOL8700 classes

Chapter 1

Contents

Chapter 2

Packages needed

Resources for class 2022 and future 8700 classes

```
library(rmarkdown)
library(rticles)
library(bookdown)
library(bookdownplus)
```

2.1 Keeping your book upto date

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(jtools)
```

This workshop will help you apply the principles taught in BIOL8291 to your own experiment. To prepare, you will create an outline of your experimental plan, focusing on the statistical framework of your design.

Chapter 3

Experimental plan template

Now that you have been developing a research question over the last few months, the next step is to design a set of experiments that will specifically test your research aims.

We are often focused on the “cookbook” aspect of experiments – the protocols and steps required to conduct each experiment. However, it’s critical to spend time designing your overall experimental approach and the finer details to ensure that your research will produce robust data that can be clearly analysed without bias. When we test specific questions, we want to avoid statistical issues such as “noise” and “confounding” factors.

Terry Neeman will be delivering a workshop to help you strengthen your experimental plan

– both in terms of your proposed design, and to help you more clearly and accurately explain the rationale and set-up of your experiments.

Below is an outline of questions for you to answer/justify for each part of your experimental plan. You will also need to draw two figures (digital drawings preferred) for each aim that show:

1) a simple overview of the experimental plan related to the research aim,
2) a detailed “snap-shot” of the experimental set up (i.e., how will the plates, plants, etc. be arranged? Will there be a row-column design? Blocking? Randomization? What treatments will be applied and how many replicates will be tested?)

You can access the template for the experimental plan on Wattle or a copy can be found within this repository. DOWNLOAD NOW.

The template is laid out below as follows:

3.1 Research Question

```
#input question here
```

3.2 Experimental Aims

For each experiment explain the overall experimental approach (1-3 sentences + overview diagram) list:

1) Response/Outcome measure for the experiment;

```
#input question here
```

2) The experimental factor(s) of interest;

```
#input question here
```

3) The experimental conditions (groups for comparisons);

```
#input question here
```

4) The number of replicates/sample size for each condition;

```
#input question here
```

5) The experimental control(s);

```
#input question here
```

6) Are their any potentially confounding factors ("nuisance factors")?

```
#input question here
```

7) Briefly explain how they will be tracked/or mitigated.

```
#input question here
```

Briefly explain the design of the experiment and provide a diagram that shows a "snap shot" of the experimental set-up (e.g., how all the plants under different experimental conditions will be arranged, all the plates in the lab, a flow chart of computational steps, etc.). Make sure to consider and include relevant design aspects like blocking, randomisation, as well as to clearly indicate treatments, replicate numbers, and controls.

> TIP: Start thinking about how you will analyse your data: "what statistical tests would you use?"

Chapter 4

Sampling design

Outcome vs experimental factors matter. How we construct models is directly related to the underlying structure of the data we have collected.

```
#general packages used
library(tidyverse)
library(knitr)
library(gridExtra)
library(here)
```

4.1 Terry's lecture in RMarkdown

Understanding variance is very important for building statistical models. But what it is really about is structuring and generating reproducible experiments.

```
knitr::include_graphics("study_experiment_design/Experimental-plan-workshop Terry Neeman 17 May 2
```

Experimental Plan in Biology Research

A Statistical Perspective

Terry Neeman

Australian National University

May 2022

We should all know what these are and how to render a report in RMarkdown. Next we will produce a RMarkdown document for the question we have been working on ready to add data and other sampling design information.

1. Outcome and predictor variables
2. Statistical model fitting
3. Visual results

4.1.1 How would you design your experiment?

What will the actual design look like? How are the treatments and samples laid out within the experimental layout.

4.1.2 Model selection

Defining models and the notation associated with the aims and hypotheses of the research project is an important part of communicating reproducing results.

4.1.2.1 Model construction

Went we are testing hypothesis we can only disprove the null hypothesis. To do this we can fit a series of linear models increasing in complexity as we ask our research questions.

4.1.2.1.1 Null model In statistical tests the null model can be defined as almost anything however most commonly this is the simplist model that assumes no relationship between the outcome and the treatments.

$$nullmodel : lm(predictor = 0)$$

##### Different mean

4.1.2.1.2 Interaction More complex model.

$$model_1 = lm(activity \equiv GBP8 + (1|litter) + (1|mouseID))$$

Chapter 5

Descriptive information

As we have discussed there will and are a collection of attributes that your research will have without/before collecting data. These things are important for prelimary analysis and things like power analysis. This chapter helps create descriptive statistics for our research.

We should all know what these are and how to render/generate a report or document in RMarkdown.

The next we will produce a RMarkdown document for the question we have been working on ready to add data and other sampling design information.

```
#general packages used
library(tidyverse)
```

5.1 Importing data

From the model above it is possible to generate a dataset to produce a model fit and outcomes generated.

```
datRawIn <- read.csv(here::here("notes2022/data/Analysis_ardMods.csv"))
```

Select variables from dataset and check the numerical structure is correct in R (numerical, factor etc).

```
# 3what response
responseACT <- c(datRawIn$Delta.Ct.Mean)
responseACT2 <- c(datRawIn$EQ.Ct.Mean)



 p1 <- hist(responseACT)
 p2 <- hist(responseACT2)
```

```
par(mfrow = c(2,2))
hist(responseACT2)
hist(responseACT)
```

What does this tell us about how RProjects and other funky things work?

3. Data import

```
# dat <- read.csv("data/Analysis_ardMods.csv")
dat <- read.csv("notes2022/data/AnalysisSimple2.csv")

glimpse(dat)
```

```
## Rows: 18
## Columns: 13
## $ ï..qPCR.ID  <chr> "Sample 1", "Sample 2", "Sample 3", "Sample 4", "Sample 5"~
## $ Sample.ID   <chr> "Ab39.4-4", "Ab39.4-5", "Ab39.4-6", "Ab39.4-7", "Ab39.4-8"~
## $ Gapdh.CT    <dbl> 12.90898, 14.25320, 13.53162, 13.37721, 14.24936, 15.05268~
## $ IL.12p40.CT <dbl> 29.03335, 29.49289, 29.35403, 30.03361, 28.45837, 29.93511~
## $ DeltaCT     <dbl> 16.12437, 15.23969, 15.82242, 16.65640, 14.20900, 14.88243~
## $ Yform       <dbl> 1.39985e-05, 2.58461e-05, 1.72575e-05, 9.68108e-06, 5.2803~
## $ NormY       <dbl> 1.3998531, 2.5846142, 1.7257488, 0.9681077, 5.2803716, 3.3~
## $ CT          <dbl> 12.90898, 14.25320, 13.53162, 13.37721, 14.24936, 15.05268~
## $ IL.6.CT     <dbl> 24.87968, 30.88497, 25.32136, 22.90003, 26.01840, 27.08113~
## $ K.J         <dbl> 11.970696, 16.631767, 11.789745, 9.522829, 11.769031, 12.0~
## $ Yform.1     <dbl> 0.0002491500, 0.0000098478, 0.0002824440, 0.0013593860, 0.~
## $ NormY.1     <dbl> 0.249150312, 0.009847804, 0.282444495, 1.359386108, 0.2865~
## $ treatment   <chr> "red", "red", "red", "red", "red", "blue", "blue", "blue",~
```

```
variable.names(dat)
```

```
##  [1] "ï..qPCR.ID"  "Sample.ID"   "Gapdh.CT"    "IL.12p40.CT" "DeltaCT"
##  [6] "Yform"       "NormY"       "CT"          "IL.6.CT"     "K.J"
## [11] "Yform.1"     "NormY.1"     "treatment"
```

```
dat1 <- dat %>%
  select(CT,NormY, treatment)

dat2 <- dat1 %>%
          mutate(treatment = as.factor(treatment))

glimpse(dat2)
```

```
## Rows: 18
## Columns: 3
## $ CT        <dbl> 12.90898, 14.25320, 13.53162, 13.37721, 14.24936, 15.05268, ~
## $ NormY     <dbl> 1.3998531, 2.5846142, 1.7257488, 0.9681077, 5.2803716, 3.310~
## $ treatment <fct> red, red, red, red, red, blue, blue, blue, blue, blue, blue,~
```

```
table(dat2$treatment)
```

```
##
## blue  red
##    9    9
```

```
levels(dat2$treatment)
```

```
## [1] "blue" "red"
```

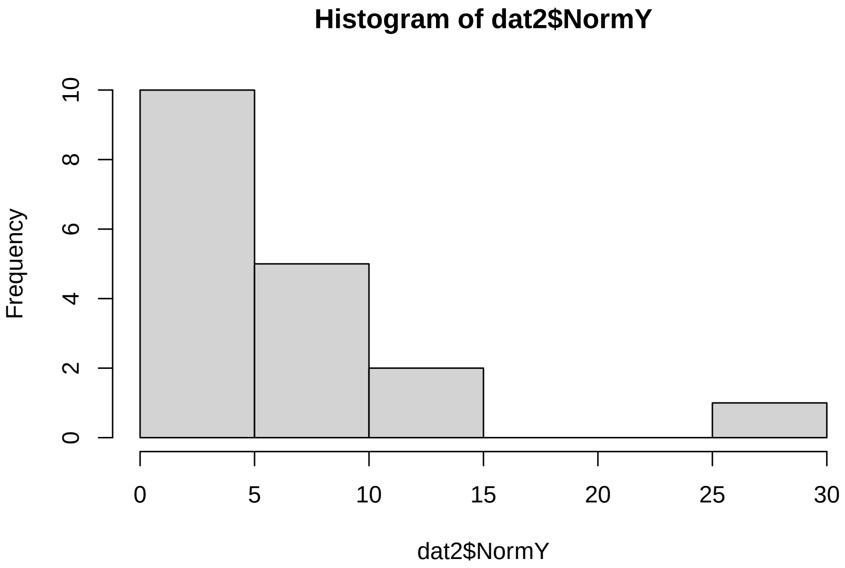
```
table(dat2$NormY)
```

```
##
## 0.968107711 1.399853134 1.725748792 2.347926871 2.521406058 2.584614246
##           1           1           1           1           1           1
## 2.941109452 3.112983596 3.310870251  4.15262465 5.170244383 5.280371649
##           1           1           1           1           1           1
## 7.145824462  7.43600363 9.711078137  11.0225451 13.68043761 29.95071366
##           1           1           1           1           1           1
```

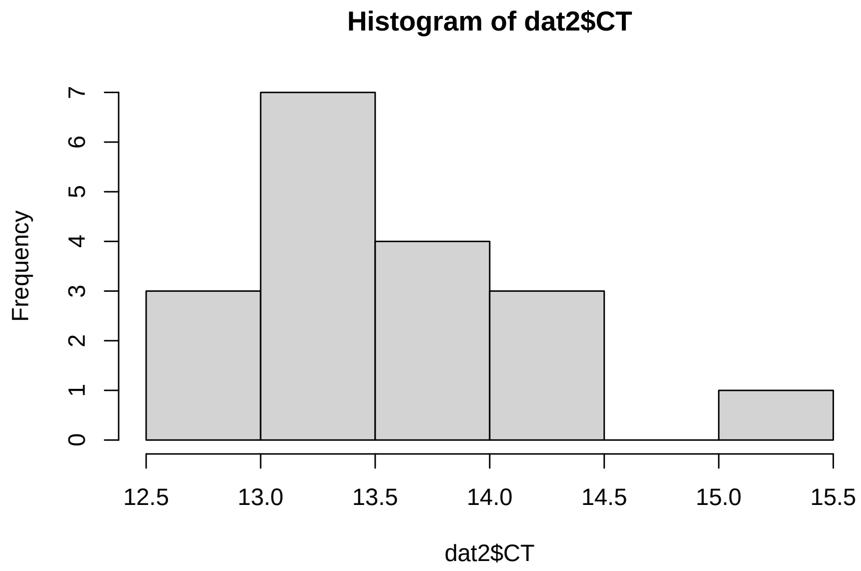
```
levels(dat2$NormY)
```

```
## NULL
```

```
hist(dat2$NormY)
```



```
hist(dat2$CT)
```



```
# removedData <- dat %>%
#   filter(Omit != "false")
# # length(unique(dat$Well.Position))
# # outcome <- dat$
# table(dat$Omit)
# head(removedData)
```

5.1.0.1 Visualising data

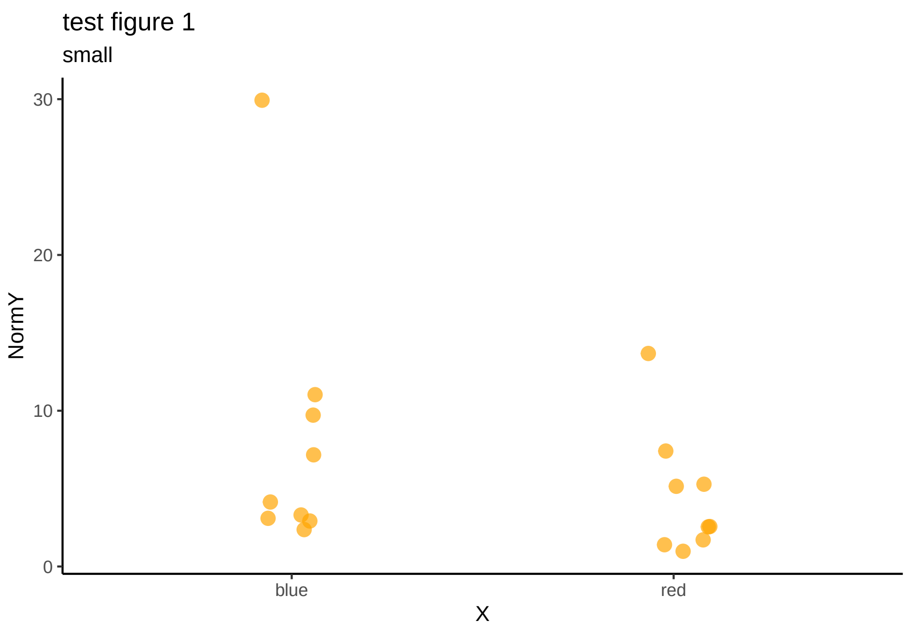
Data visualisation

```
variable.names(dat2)
```

```
## [1] "CT"        "NormY"     "treatment"
```

```
p1 <- ggplot(dat) +
  # geom_boxplot(aes(x = treatment, y = NormY),outlier.size = 0) +
  geom_jitter(aes(x = treatment, y = NormY), size = 3, colour = "orange", alpha = 0.7,
  theme_bw() +
  theme_classic() +
  xlab("X") +
  ggtitle("test figure 1", subtitle = "small")


p1
```



```
# # table(dat$Well.Position)
# mean(dat$Delta.Ct.Mean, na.rm = TRUE)
# sd(dat$Delta.Ct.Mean, na.rm = TRUE)
#
# hist(dat$Delta.Ct.Mean)
```

5.1.1 Scatter plot

3. Data import

```
# dat <- read.csv("data/Analysis_ardMods.csv")
```

4. Data visualisation

```
# table(dat$Well.Position)
# mean(dat$Delta.Ct.Mean, na.rm = TRUE)
# sd(dat$Delta.Ct.Mean, na.rm = TRUE)
#
# hist(dat$Delta.Ct.Mean)
```

5. Tidyverse approach

```
# variable.names(dat)
#
# #sample name
# # table(dat$Sample.Name)
# table(dat$Target.Name)
# mean(as.numeric(dat$CT), na.rm = TRUE)

##sumarise over Target Name and find n, mean, mode, median, sd, etc for each of the ta
```

6. Nicer plots using `ggplot`

7. Tidyverse approach

```
#tidy data
#tibble
```

6. ggplot

Much easier to work this this and tidyverse

7. Read a cool sampling design/issue paper

5.1.1.1 Reporting

```
#general packages used
library(tidyverse)
```

We should all know what these are and how to render a report in RMarkdown. Next we will produce a RMarkdown document for the question we have been working on ready to add data and other sampling design information.

1. Outcome and predictor variables
2. Other study examples and code
3. Other studies with same sampling design
4. Other reference material.
5. Read a cool sampling design/issue paper

Chapter 6

Power analysis

Watch this short (ish video) as a summary of what you should understand so far. What does this tell us about how RProjects and other funky things work?

```
library(stats)
power.anova.test(groups = 4, n = 5, between.var = 1, within.var = 3)
```

```
##
##      Balanced one-way analysis of variance power calculation
##
##          groups = 4
##               n = 5
##     between.var = 1
##      within.var = 3
##       sig.level = 0.05
##           power = 0.3535594
##
## NOTE: n is number in each group
```

```
# Power = 0.3535594

power.anova.test(groups = 4, between.var = 1, within.var = 3,
                 power = .80)
```

```
##
##      Balanced one-way analysis of variance power calculation
##
##          groups = 4
##               n = 11.92613
##     between.var = 1
##      within.var = 3
##       sig.level = 0.05
##           power = 0.8
##
## NOTE: n is number in each group
```

```
# n = 11.92613

## Assume we have prior knowledge of the group means:
groupmeans <- c(120, 130, 140, 150)
power.anova.test(groups = length(groupmeans),
                 between.var = var(groupmeans),
                 within.var = 500, power = .90) # n = 15.18834
```

```
##
##      Balanced one-way analysis of variance power calculation
##
##          groups = 4
##               n = 15.18834
##     between.var = 166.6667
##      within.var = 500
##       sig.level = 0.05
##           power = 0.9
##
## NOTE: n is number in each group
```

I can not find the additional code from last year but this is a slight variation on the project.

```
knitr::include_app("https://mathiasharrer.shinyapps.io/power_calculator_meta_analysis/
```

```
## PhantomJS not found. You can install it with webshot::install_phantomjs(). If it is
```

6.1 Meta-analysis power test

```
# The included numbers will per calculate power for a meta-analysis to detect a summar

es <- 0.2 # Enter your summary effect size
as <- 50  # Average per number per group
mk <- 15  # Number of effect sizes
hg <- 1   # Heterogeniety (".33" for small, "1" for moderate, & "3" for large)

eq1 <- ((as+as)/((as)*(as))) + ((es^2)/(2*(as+as)))
eq2 <- hg*(eq1)
eq3 <- eq2+eq1
eq4 <- eq3/mk
eq5 <- (es/sqrt(eq4))
Power <- (1-pnorm(1.96-eq5)) # Two-tailed
```

```
Power
```

```
## [1] 0.7798811
```

6.1.0.1 Next steps

Collect some sample data and application.

z

One aspect that can be challenging when working with RMarkdown documents for manuscripts is references.

The references for a bookdown or rmarkdown file can be included using the following information in the `yml` header of the index file.

6.2 Manual references

6.3 Packages

The goal of BIOL877_setup_2022 repository here is to provide a starting point for using RMarkdown and RStudio to undertake research.

6.4 Download project and files

We checked and loaded an RMarkdown

This is a *sample* book written in **Markdown** and extended from the original RMD file. You can use anything that Pandoc's Markdown supports; for example, a math equation $a^2 + b^2 = c^2$.

6.5 Usage

Each **bookdown** chapter is an .Rmd file, and each .Rmd file can contain one (and only one) chapter. A chapter *must* start with a first-level heading: `# A good chapter`, and can contain one (and only one) first-level heading.

Use second-level and higher headings within chapters like: `## A short section` or `### An even shorter section`.

The `index.Rmd` file is required, and is also your first book chapter. It will be the homepage when you render the book.

6.6 Render book

You can render the HTML version of this example book without changing anything:

1. Find the **Build** pane in the RStudio IDE, and
2. Click on **Build Book**, then select your output format, or select "All formats" if you'd like to use multiple formats from the same book source files.

Or build the book from the R console:

```
bookdown::render_book()
```

To render this example to PDF as a `bookdown::pdf_book`, you'll need to install XeLaTeX. You are recommended to install TinyTeX (which includes XeLaTeX): https://yihui.org/tinytex/.

6.7 Preview book

As you work, you may start a local server to live preview this HTML book. This preview will update as you edit the book when you save individual .Rmd files. You can start the server in a work session by using the RStudio add-in "Preview book", or from the R console:

```
bookdown::serve_book()
```

6.8 Output files

One of the benefits of working in RMarkdown is that the 'gitbook' extension is that it can easily be hosted on github. To do this with the least hurdles is to publish the output files as a github pages site from the `.docs/`.