Homework Advice

David M. Rocke

January 28, 2026

- For document in Word or the equivalent, code and output will look funny unless you change to a monospace font. Default fonts have characters of variable width and different spaces between characters. These are called proportionally spaced. The standard monospace font in Window is Courier New. For Macs, the traditional monospace font is SF Mono Regular, though there are others. If you copy from R to a document with a monospaced font, it does not change the appearance.
- Choose No Spacing on the Home tab.
- Highlight the whole document (at the start or later) with Ctrl-a.
- Choose Courier New and 11 or 12 points. There are 72 points to the inch.
- Choose paragraph on the home tab. Make sure line spacing is single and spacing before/after is 0.

- Alternatively, you can put the code and text output into a text file such as Notepad in Windows or TextEdit in iOS.
- R Markdown allows code, text output, graphic output, and your comments and explanations all to be in one document.
- Please realize that the Reader has to go through your submission to see if you have done things right or made mistakes. Don't make this difficult for him—he is a second-year BME graduate student, as you will be next year.

- If a variable is coded with numbers, such as $1, 2, 3, \ldots$, then ask yourself if it is a measurement of something. If not, then make it a factor.
- In a clinical study, weight is a numeric variable, systolic blood pressure is a numerical variable, but patient number is not a numerical variable and needs to be a factor.

- Your output should match the code you submit. If you have a document with code and output, and if the code would not produce the output you included, what would anyone think?
- Do not type in numbers which are already in a file you have loaded or should have loaded. This leads to frequent errors.

- You should almost never use the equal variance t-test. If the variances of two groups really are equal, the performance of the equal variance t-test is a little better than that of the Welch t-test which does not make that assumption.
- If the variances truly are not equal, the performance of the equal variance t-test will be worse, sometime catastrophically worse.
- It is often hard to tell if the variances really are equal
- This makes the equal variance test a bad gamble, with small possible gain, and large possible loss.
- For more than two groups, there is no good alternative to the equal variance analysis, but check if there are large variations in the variance.

- Statistical tests for normality, equality of variance, serial correlation and others are often not very informative. They generally have low statistical power, which is the chance of a significant result if the equality etc. is not true.
- Conditional use of these tests is usually bad: test for equal variance; if $p < 0.05$ use the Welch t-test; otherwise use the equal variance t-test. Usually, these compound procedures perform worse than the the default test without a pre-test.
- An even worse procedure is testing for normality: if significant, use a rank test; if not, use a t-test. You see this in academic papers in which the person doing the statistics does not know much.
- Procedures like the Welch t-test or linear regression often work well even if the exact assumptions fail. Exact assumptions are always false! Aproximately true is usually good enough.

- If there is an outlier from the plots, as there might be in some homework assignments, it would make sense to see what the analysis looks like without that point, but not necessarily permanently delete the point. Sometimes an "outlier" is the most important observation in the experiment.
- If there is a trend in the results when plotted in the time order the experiments were done, this is disturbing, even with randomization.

Don't edit the output too much: summary(lm) is much better than print(lm).

```
> print(lm(y~method+run, data = yield))

Call:
lm(formula = y ~ method + run, data = yield)

Coefficients:
(Intercept)      methodB          run
    43.9459      20.0789       0.7723
```

```
> summary(lm(y~method+run, data = yield))

Call:
lm(formula = y ~ method + run, data = yield)

Residuals:
    Min      1Q  Median      3Q     Max
-9.9589 -2.9716 -0.3824  4.1236  8.9689

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  43.9459     2.7972  15.711 1.48e-11 ***
methodB      20.0789     2.3984   8.372 1.96e-07 ***
run           0.7723     0.2080   3.714  0.00173 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.36 on 17 degrees of freedom
Multiple R-squared:  0.828,     Adjusted R-squared:  0.8077
F-statistic: 40.91 on 2 and 17 DF,  p-value: 3.181e-07
```

- The data from the textbook are in the `data.zip` file.
- Files beginning with `fig` are from figures in the book.
- Files beginning with `prb` are from problems in the book. These are at the end of each chapter.
- Files beginning with `exe` are from exercises in the book. These are for self-study at places in the textbook where you can practice.
- Files beginning with `tab` are from tables in the book.
- This information is in the file `ReadMe.text`. You should always read files named `readme`!