Chapter 5

Paired Experiments and Randomized Block Experiments

Contents

A **randomized block design** is often used instead of a completely randomized design in studies where there is extraneous variation among the experimental units that may influence the response. A significant amount of the extraneous variation may be removed from the comparison of treatments by partitioning the experimental units into fairly **homogeneous subgroups** or **blocks**.

For example, suppose you are interested in comparing the effectiveness of four antibiotics for a bacterial infection. The recovery time after administering an antibiotic may be influenced by a patient's general health, the extent of their infection, or their age. Randomly allocating experimental subjects to the treatments (and then comparing them using a one-way ANOVA) may produce one treatment having a "favorable" sample of patients with features that naturally lead to a speedy recovery. Alternatively, if the characteristics that affect the recovery time are spread across treatments, then the variation within samples due to these uncontrolled features can dominate the effects of the treatment, leading to an inconclusive result.

A better way to design this experiment would be to **block** the subjects into groups of four patients who are alike as possible on factors other than the treatment that influence the recovery time. The four treatments are then randomly assigned to the patients (one per patient) within a block, and the recovery time measured. The blocking of patients usually produces a more sensitive comparison of treatments than does a completely randomized design because the variation in recovery times due to the blocks is eliminated from the comparison of treatments.

A randomized block design is a **paired experiment** when two treatments are compared. The usual analysis for a paired experiment is a parametric or

non-parametric paired comparison.

Randomized block (RB) designs were developed to account for soil fertility gradients in agricultural experiments. The experimental field would be separated into strips (blocks) of fairly constant fertility. Each strip is partitioned into equal size plots. The treatments, say varieties of corn, are randomly assigned to the plots, with each treatment occurring the same number of times (usually once) per block. All other factors that are known to influence the response would be controlled or fixed by the experimenter. For example, when comparing the mean yields, each plot would receive the same type and amount of fertilizer and the same irrigation plan.

The discussion will be limited to randomized block experiments with one factor. Two or more factors can be used with a randomized block design. For example, the agricultural experiment could be modified to compare four combinations of two corn varieties and two levels of fertilizer in each block instead of the original four varieties. In certain experiments, each experimental unit receives each treatment. The experimental units are "natural" blocks for the analysis.

Example: Comparison of Treatments for Itching Ten[1] male volunteers between 20 and 30 years old were used as a study group to compare seven treatments (5 drugs, a placebo, and no drug) to relieve itching. Each subject was given a different treatment on seven study days. The time ordering of the treatments was randomized across days. Except on the no-drug day, the subjects were given the treatment intravenously, and then itching was induced on their forearms using an effective itch stimulus called cowage. The subjects recorded the duration of itching, in seconds. The data are given in the table below. From left to right the drugs are: papaverine, morphine, aminophylline, pentobarbitol, tripelenamine.

The volunteers in the study were treated as blocks in the analysis. At best, the volunteers might be considered a representative sample of males between the ages of 20 and 30. This limits the extent of inferences from the experiment. The scientists can not, without sound medical justification, extrapolate the results to children or to senior citizens.

```
library(tidyverse)

# load ada functions
source("ada_functions.R")

#### Example: Itching
dat_itch <-
  read_csv("http://statacumen.com/teach/ADA2/notes/ADA2_notes_Ch05_itch.csv")
## Parsed with column specification:
## cols(
##   Patient = col_double(),
##   Nodrug = col_double(),
##   Placebo = col_double(),
##   Papv = col_double(),
##   Morp = col_double(),
##   Amino = col_double(),
##   Pento = col_double(),
##   Tripel = col_double()
## )
```

[1] Beecher, 1959

	Patient	Nodrug	Placebo	Papv	Morp	Amino	Pento	Tripel
1	1	174	263	105	199	141	108	141
2	2	224	213	103	143	168	341	184
3	3	260	231	145	113	78	159	125
4	4	255	291	103	225	164	135	227
5	5	165	168	144	176	127	239	194
6	6	237	121	94	144	114	136	155
7	7	191	137	35	87	96	140	121
8	8	100	102	133	120	222	134	129
9	9	115	89	83	100	165	185	79
10	10	189	433	237	173	168	188	317

5.1 Analysis of a Randomized Block Design

Assume that you designed a randomized block experiment with I blocks and J treatments, where each treatment occurs once in each block. Let y_{ij} be the response for the j^{th} treatment within the i^{th} block. The model for the experiment is

$$y_{ij} = \mu_{ij} + e_{ij},$$

where μ_{ij} is the population mean response for the j^{th} treatment in the i^{th} block and e_{ij} is the deviation of the response from the mean. The population means are assumed to satisfy the additive model

$$\mu_{ij} = \mu + \alpha_i + \beta_j$$

where μ is a grand mean, α_i is the effect for the i^{th} block, and β_j is the effect for the j^{th} treatment. The responses are assumed to be independent across blocks, normally distributed and with constant variance. The randomized block model does not require the observations within a block to be independent, but does assume that the correlation between responses within a block is identical for each pair of treatments.

The model is sometimes written as

Response = Grand Mean + Treatment Effect + Block Effect + Residual.

Given the data, let $\bar{y}_{i\cdot}$ be the i^{th} block sample mean (the average of the responses in the i^{th} block), $\bar{y}_{\cdot j}$ be the j^{th} treatment sample mean (the average of the responses on the j^{th} treatment), and $\bar{y}_{\cdot\cdot}$ be the average response of all IJ observations in the experiment.

An ANOVA table for the randomized block experiment partitions the Model SS into SS for Blocks and Treatments.

Source	*df*	SS	MS
Blocks	$I-1$	$J\sum_i(\bar{y}_{i\cdot}-\bar{y}_{\cdot\cdot})^2$	
Treats	$J-1$	$I\sum_j(\bar{y}_{\cdot j}-\bar{y}_{\cdot\cdot})^2$	
Error	$(I-1)(J-1)$	$\sum_{ij}(y_{ij}-\bar{y}_{i\cdot}-\bar{y}_{\cdot j}+\bar{y}_{\cdot\cdot})^2$	
Total	$IJ-1$	$\sum_{ij}(y_{ij}-\bar{y}_{\cdot\cdot})^2$.	

A primary interest is testing whether the treatment effects are zero: $H_0 : \beta_1 = \cdots = \beta_J = 0$. The treatment effects are zero if in each block the population mean responses are identical for each treatment. A formal test of no treatment effects is based on the p-value from the F-statistic F_{obs} = MS Treat/MS Error. The p-value is evaluated in the usual way (i.e., as an upper tail area from an F-distribution with $J-1$ and $(I-1)(J-1)$ df.) This H_0 is rejected when the treatment averages $\bar{y}_{\cdot j}$ vary significantly relative to the error variation.

A test for no block effects ($H_0 : \alpha_1 = \cdots = \alpha_I = 0$) is often a secondary interest, because, if the experiment is designed well, the blocks will be, by construction, noticeably different. There are no block effects if the population mean response for an arbitrary treatment is identical across blocks. A formal test of no block effects is based on the p-value from the the F-statistic F_{obs} = MS Blocks/MS Error. This H_0 is rejected when the block averages $\bar{y}_{i\cdot}$ vary significantly relative to the error variation.

The randomized block model is easily fitted in the `lm()` function. Before illustrating the analysis on the itching data, let me mention five important points about randomized block analyses:

1. The F-test p-value for comparing $J = 2$ treatments is identical to the p-value for comparing the two treatments using a paired t-test.
2. The Block SS plus the Error SS is the Error SS from a one-way ANOVA comparing the J treatments. If the Block SS is large relative to the Error SS from the two-factor model, then the experimenter has eliminated a substantial portion of the variation that is used to assess the differences among the treatments. This leads to a more sensitive comparison of treatments than would have been obtained using a one-way ANOVA.
3. The RB model is equivalent to an additive or no interaction model for a two-factor experiment, where the blocks are levels of one of the factors. The analysis of a randomized block experiment under this model is the same analysis used for a two-factor experiment with no replication (one observation per cell). We will discuss the two-factor design soon.

4. Under the sum constraint on the parameters (i.e., $\sum_i \alpha_i = \sum_j \beta_j = 0$), the estimates of the grand mean, block effects, and treatment effects are $\hat{\mu} = \bar{y}_{\cdot\cdot}$, $\hat{\alpha}_i = \bar{y}_{i\cdot} - \bar{y}_{\cdot\cdot}$, and $\hat{\beta}_j = \bar{y}_{\cdot j} - \bar{y}_{\cdot\cdot}$, respectively. The estimated mean response for the $(i,j)^{th}$ cell is $\hat{\mu}_{ij} = \hat{\mu} + \hat{\alpha}_i + \hat{\beta}_j = \bar{y}_{i\cdot} + \bar{y}_{\cdot j} - \bar{y}_{\cdot\cdot}$.
5. The F-test for comparing treatments is appropriate when the responses within a block have the same correlation. This is a reasonable working assumption in many analyses. A multivariate repeated measures model can be used to compare treatments when the constant correlation assumption is unrealistic, for example when the same treatment is given to an individual over time.

RB Analysis of the Itching Data First we reshape the data to long format so each observation is its own row in the `data.frame` and indexed by the `Patient` and `Treatment` variables.

```
dat_itch_long <-
  dat_itch %>%
  pivot_longer(
    cols      = -1          # all but the first column
  , names_to  = "Treatment"
  , values_to = "Seconds"
  , values_drop_na = TRUE   # drop the NA values in long format
  ) %>%
  mutate(
    Patient   = factor(Patient)
  , Treatment = factor(Treatment)
  )

str(dat_itch_long)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame': 70 obs. of  3 variables:
##  $ Patient  : Factor w/ 10 levels "1","2","3","4",..: 1 1 1 1 1 1 1 2 2 2 ...
##  $ Treatment: Factor w/ 7 levels "Amino","Morp",..: 3 6 4 2 1 5 7 3 6 4 ...
##  $ Seconds  : num  174 263 105 199 141 108 141 224 213 103 ...
```

```
head(dat_itch_long, 3)
```

```
## # A tibble: 3 x 3
##   Patient Treatment Seconds
##   <fct>   <fct>       <dbl>
## 1 1       Nodrug        174
## 2 1       Placebo       263
## 3 1       Papv          105
```

```
tail(dat_itch_long, 3)
```

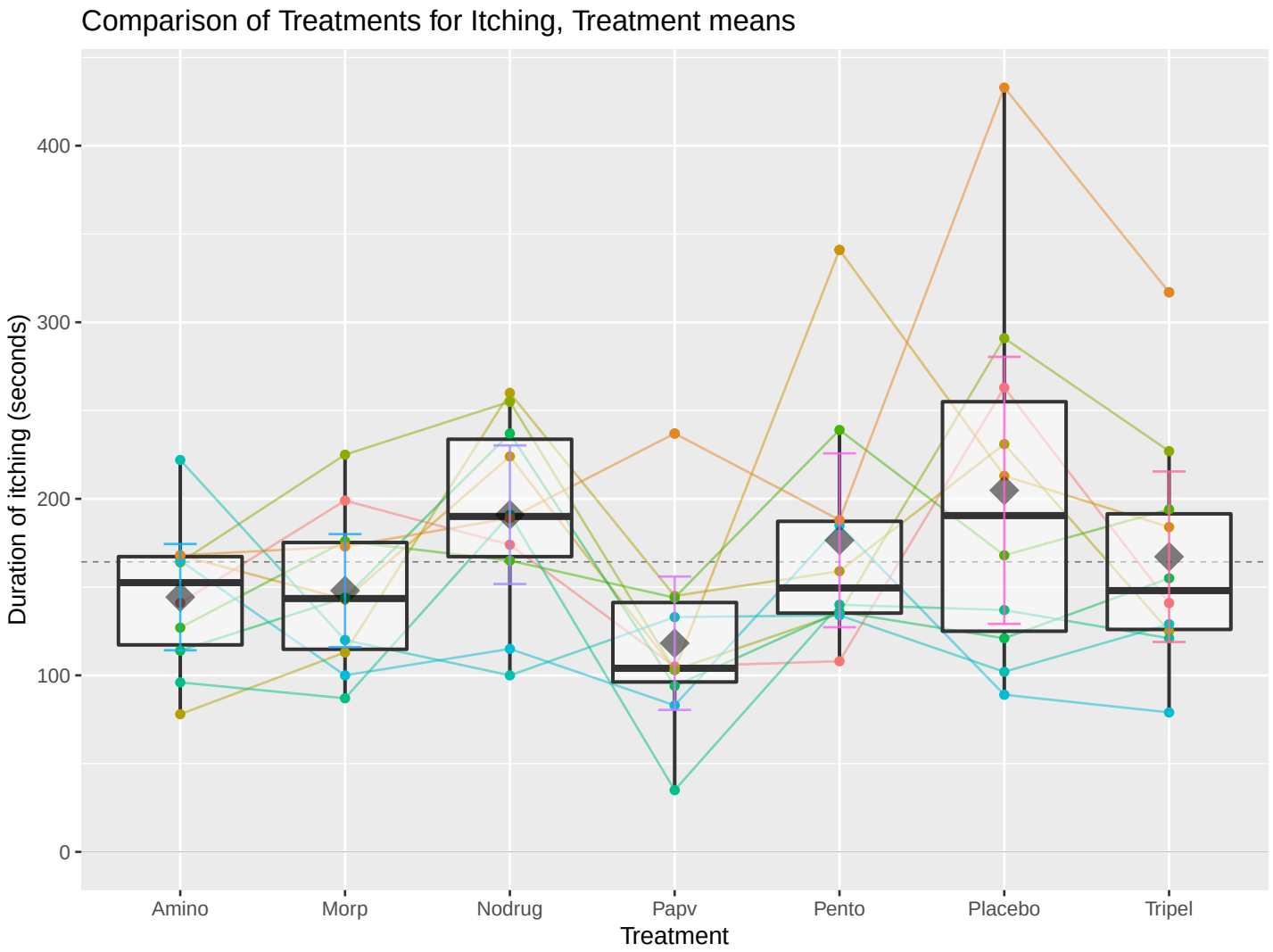
```
## # A tibble: 3 x 3
```

```
##   Patient Treatment Seconds
##   <fct>   <fct>       <dbl>
## 1 10      Amino         168
## 2 10      Pento         188
## 3 10      Tripel        317
```

As a first step, I made side-by-side boxplots of the itching durations across treatments. The boxplots are helpful for informally comparing treatments and visualizing the data. The differences in the level of the boxplots will usually be magnified by the F-test for comparing treatments because the variability within the boxplots includes block differences which are moved from the Error SS to the Block SS. The plot also includes the 10 Patients with lines connecting their measurements to see how common the treatment differences were over patients. I admit, this plot is a little too busy.

Each of the five drugs appears to have an effect, compared to the placebo and to no drug. Papaverine appears to be the most effective drug. The placebo and no drug have similar medians. The relatively large spread in the placebo group suggests that some patients responded adversely to the placebo compared to no drug, whereas others responded positively.

```
# Plot the data using ggplot
library(ggplot2)
p <- ggplot(dat_itch_long, aes(x = Treatment, y = Seconds))
# plot a reference line for the global mean (assuming no groups)
p <- p + geom_hline(aes(yintercept = 0),
                    colour = "black", linetype = "solid", size = 0.2, alpha = 0.3)
p <- p + geom_hline(aes(yintercept = mean(Seconds)),
                    colour = "black", linetype = "dashed", size = 0.3, alpha = 0.5)
# colored line for each patient
p <- p + geom_line(aes(group = Patient, colour = Patient), alpha = 0.5)
# boxplot, size=.75 to stand out behind CI
p <- p + geom_boxplot(size = 0.75, alpha = 0.5)
# points for observed data
p <- p + geom_point(aes(colour = Patient))
# diamond at mean for each group
p <- p + stat_summary(fun.y = mean, geom = "point", shape = 18, size = 6,
                      alpha = 0.5)
# confidence limits based on normal distribution
p <- p + stat_summary(fun.data = "mean_cl_normal", geom = "errorbar",
                      width = .2, aes(colour=Treatment), alpha = 0.8)
p <- p + labs(title = "Comparison of Treatments for Itching, Treatment means")
p <- p + ylab("Duration of itching (seconds)")
# removes legend
p <- p + theme(legend.position="none")
print(p)
```



To fit the RB model in `lm()`, you need to specify blocks (`Patient`) and treatments (`Treatment`) as **factor** variables, and include each to the right of the tilde symbol in the **formula** statement. The response variable **Seconds** appears to the left of the tilde.

```
lm_s_t_p <-
  lm(
    Seconds ~ Treatment + Patient
  , data = dat_itch_long
  )
library(car)
Anova(lm_s_t_p, type=3)
```

```
## Anova Table (Type III tests)
##
## Response: Seconds
##             Sum Sq Df F value    Pr(>F)
## (Intercept)  87704  1 28.3372 2.019e-06 ***
## Treatment    53013  6  2.8548  0.017303 *
```

```
## Patient     103280  9  3.7078  0.001124 **
## Residuals   167130 54
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(lm_s_t_p)
```

```
##
## Call:
## lm(formula = Seconds ~ Treatment + Patient, data = dat_itch_long)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -81.286 -34.800  -8.393  30.900 148.914
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       141.586     26.598   5.323 2.02e-06 ***
## TreatmentMorp       3.700     24.880   0.149  0.88233
## TreatmentNodrug    46.700     24.880   1.877  0.06592 .
## TreatmentPapv     -26.100     24.880  -1.049  0.29883
## TreatmentPento     32.200     24.880   1.294  0.20109
## TreatmentPlacebo   60.500     24.880   2.432  0.01838 *
## TreatmentTripel    22.900     24.880   0.920  0.36144
## Patient2           35.000     29.737   1.177  0.24436
## Patient3           -2.857     29.737  -0.096  0.92381
## Patient4           38.429     29.737   1.292  0.20176
## Patient5           11.714     29.737   0.394  0.69518
## Patient6          -18.571     29.737  -0.625  0.53491
## Patient7          -46.286     29.737  -1.557  0.12543
## Patient8          -27.286     29.737  -0.918  0.36292
## Patient9          -45.000     29.737  -1.513  0.13604
## Patient10          82.000     29.737   2.758  0.00793 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 55.63 on 54 degrees of freedom
## Multiple R-squared:  0.4832,Adjusted R-squared:  0.3397
## F-statistic: 3.367 on 15 and 54 DF,  p-value: 0.00052
```

The order to look at output follows the hierarchy of multi-parameter tests down to single-parameter tests.

1. The F-test at the bottom of the `summary()` tests for both no block effects and no treatment effects. If there are no block effects and no tretment effects then the mean itching time is independent of treatment and patients. The p-value of 0.0005 strongly suggests that the population mean

itching times are not all equal.

2. The ANOVA table at top from `Anova()` partitions the Model SS into the SS for Blocks (Patients) and Treatments. The Mean Squares, F-statistics, and p-values for testing these effects are given. For a RB design with the same number of responses per block (i.e., no missing data), the Type I and Type III SS are identical, and correspond to the formulas given earlier. The distinction between Type I and Type III SS is important for unbalanced problems, an issue we discuss later. The F-tests show significant differences among the treatments (p-value=0.017) and among patients (p-value=0.001).
3. The individual parameter (coefficient) estimates in the `summary()` are likely of less interest since they test differences from the baseline group, only. The multiple comparisons in the next section will indicate which factor levels are different from others.

Multiple comparisons Multiple comparison and contrasts are not typically straightforward in R, though some newer packages are helping make them easier. Below I show one way that I think is relatively easy.

The package `emmeans` is used to specify which factor to perform multiple comparisons over and which p-value adjustment method to use. The default is to use Tukey adjustments.

```
# Contrasts to perform pairwise comparisons
cont_s <-
  emmeans::emmeans(
    lm_s_t_p
  , specs = "Treatment"
  #, by = c("Patient")
  )

# Means and CIs
cont_s                                  # adjust = "tukey" is default
```

```
##  Treatment emmean   SE df lower.CL upper.CL
##  Amino        144 17.6 54    109.0      180
##  Morp         148 17.6 54    112.7      183
##  Nodrug       191 17.6 54    155.7      226
##  Papv         118 17.6 54     82.9      153
##  Pento        176 17.6 54    141.2      212
```

```
##  Placebo      205 17.6 54    169.5      240
##  Tripel       167 17.6 54    131.9      202
##
## Results are averaged over the levels of: Patient
## Confidence level used: 0.95
```

```
#confint(cont_s, adjust = "bonferroni")

# Pairwise comparisons
cont_s %>% pairs()  #adjust = "bonf")   # adjust = "tukey" is default
```

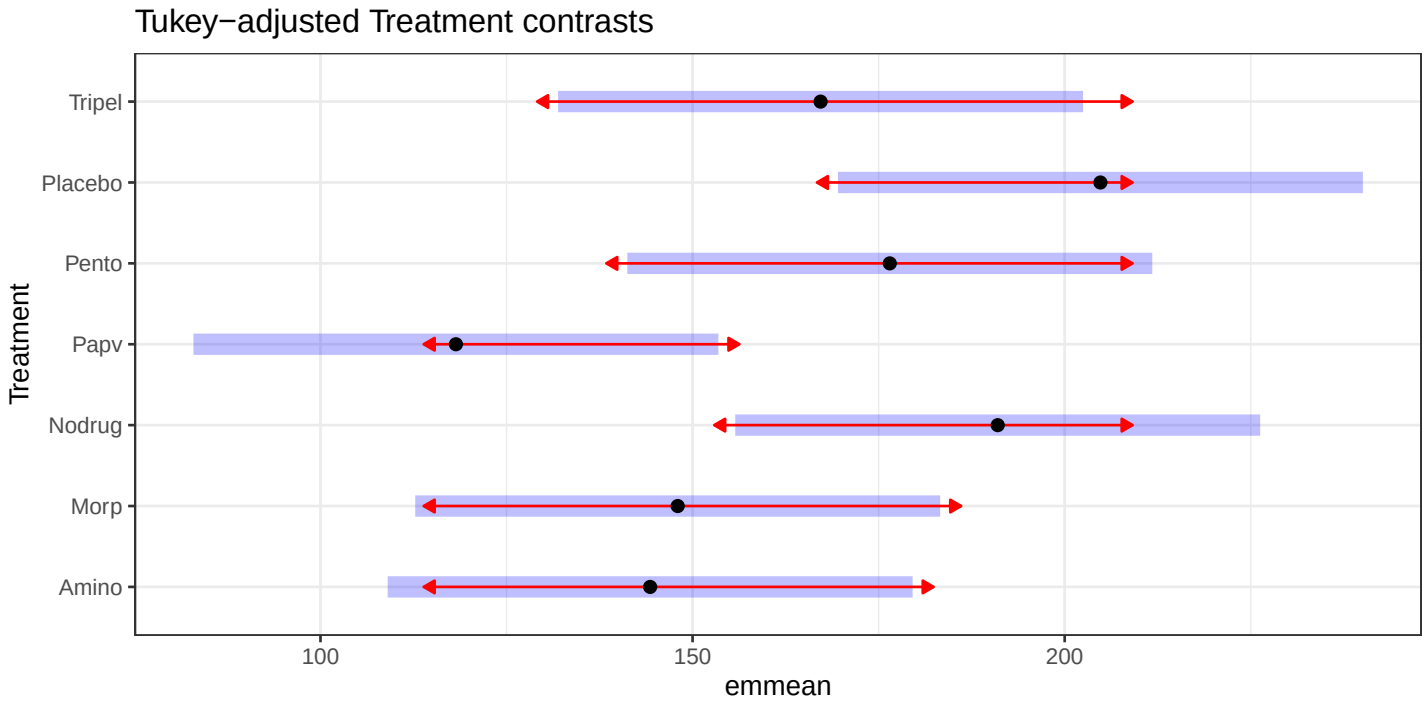
```
##  contrast         estimate   SE df t.ratio p.value
##  Amino - Morp         -3.7 24.9 54 -0.149  1.0000
##  Amino - Nodrug      -46.7 24.9 54 -1.877  0.5039
##  Amino - Papv         26.1 24.9 54  1.049  0.9398
##  Amino - Pento       -32.2 24.9 54 -1.294  0.8516
##  Amino - Placebo     -60.5 24.9 54 -2.432  0.2057
##  Amino - Tripel      -22.9 24.9 54 -0.920  0.9676
##  Morp - Nodrug       -43.0 24.9 54 -1.728  0.6006
##  Morp - Papv          29.8 24.9 54  1.198  0.8920
##  Morp - Pento        -28.5 24.9 54 -1.146  0.9108
##  Morp - Placebo      -56.8 24.9 54 -2.283  0.2710
##  Morp - Tripel       -19.2 24.9 54 -0.772  0.9867
##  Nodrug - Papv        72.8 24.9 54  2.926  0.0700
##  Nodrug - Pento       14.5 24.9 54  0.583  0.9971
##  Nodrug - Placebo    -13.8 24.9 54 -0.555  0.9978
##  Nodrug - Tripel      23.8 24.9 54  0.957  0.9610
##  Papv - Pento        -58.3 24.9 54 -2.343  0.2431
##  Papv - Placebo      -86.6 24.9 54 -3.481  0.0163
##  Papv - Tripel       -49.0 24.9 54 -1.969  0.4454
##  Pento - Placebo     -28.3 24.9 54 -1.137  0.9135
##  Pento - Tripel        9.3 24.9 54  0.374  0.9998
##  Placebo - Tripel     37.6 24.9 54  1.511  0.7369
##
## Results are averaged over the levels of: Patient
## P value adjustment: tukey method for comparing a family of 7 estimates
```

EMM plot interpretation

This **EMM plot (Estimated Marginal Means, aka Least-Squares Means)** is only available when conditioning on one variable. The **blue bars** are confidence intervals for the EMMs; don't ever use confidence intervals for EMMs to perform comparisons – they can be very misleading. The **red arrows** are for the comparisons among means; the degree to which the "comparison arrows" overlap reflects as much as possible the significance of the comparison of the two estimates. If an arrow from one mean overlaps an arrow from an-

other group, the difference is not significant, based on the adjust setting (which defaults to "tukey").

```
# Plot means and contrasts
p <- plot(cont_s, comparisons = TRUE)  #, adjust = "bonf")  # adjust = "tukey" is default
p <- p + labs(title = "Tukey-adjusted Treatment contrasts")
p <- p + theme_bw()
print(p)
```



With `summary()`, the p-value adjustment can be coerced into one of several popular methods, such as Bonferroni. Notice that the significance is lower (larger p-value) for Bonferroni below than Tukey above.

Recall how the **Bonferroni** correction works. A comparison of c pairs of levels from one factor having a family error rate of 0.05 or less is attained by comparing pairs of treatments at the $0.05/c$ level. Using this criteria, the population mean response for factor levels (averaged over the other factor) are significantly different if the p-value for the test is $0.05/c$ or less. The output actually adjusts the p-values by reporting p-value$\times c$, so that the reported adjusted p-value can be compared to the 0.05 significance level. Also, the red "comparison arrows" in the plot are longer making it more likely that Treatment comparison arrows overlap.

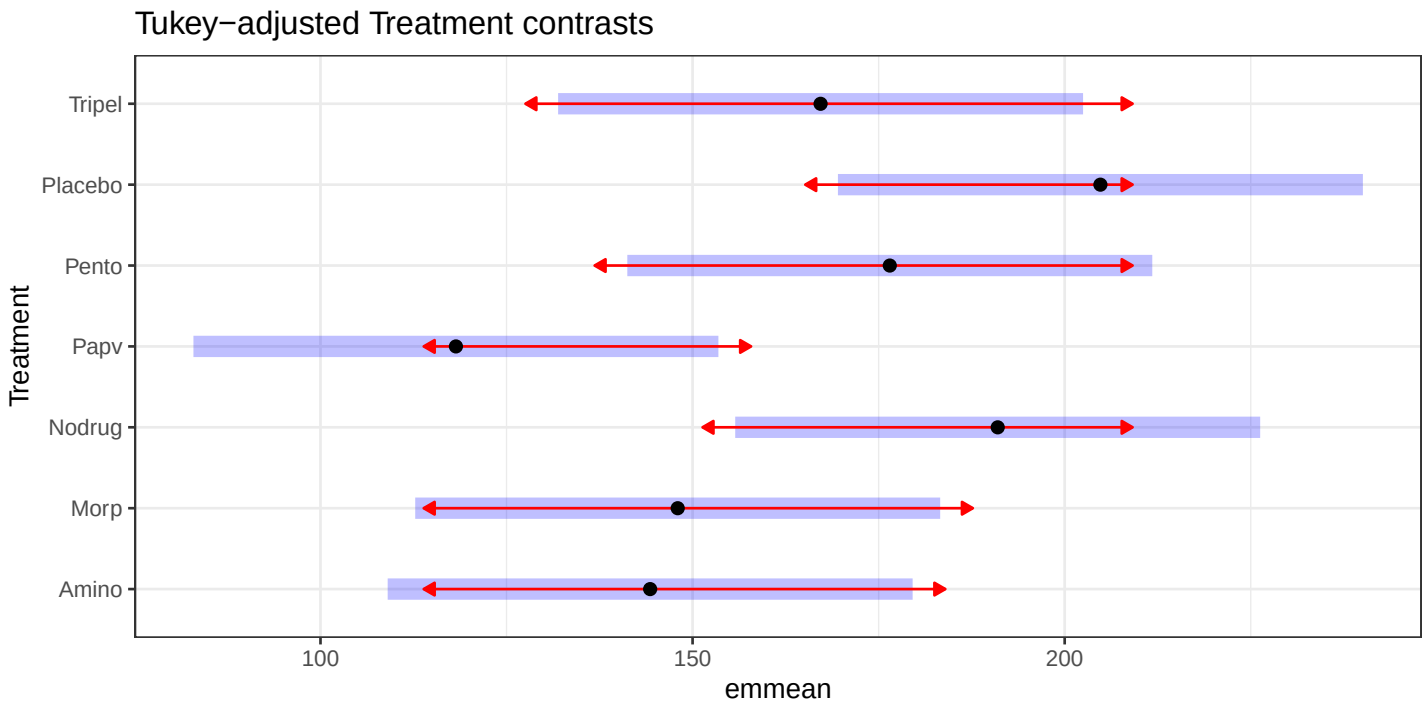
```
# Means and CIs
confint(cont_s, adjust = "bonferroni")
```

```
##  Treatment emmean   SE df lower.CL upper.CL
##  Amino        144 17.6 54     95.1      193
##  Morp         148 17.6 54     98.8      197
##  Nodrug       191 17.6 54    141.8      240
##  Papv         118 17.6 54     69.0      167
##  Pento        176 17.6 54    127.3      226
##  Placebo      205 17.6 54    155.6      254
##  Tripel       167 17.6 54    118.0      216
##
## Results are averaged over the levels of: Patient
## Confidence level used: 0.95
## Conf-level adjustment: bonferroni method for 7 estimates
```

```
# Pairwise comparisons
cont_s %>% pairs(adjust = "bonf")
```

```
##  contrast         estimate   SE df t.ratio p.value
##  Amino - Morp         -3.7 24.9 54 -0.149  1.0000
##  Amino - Nodrug      -46.7 24.9 54 -1.877  1.0000
##  Amino - Papv         26.1 24.9 54  1.049  1.0000
##  Amino - Pento       -32.2 24.9 54 -1.294  1.0000
##  Amino - Placebo     -60.5 24.9 54 -2.432  0.3859
##  Amino - Tripel      -22.9 24.9 54 -0.920  1.0000
##  Morp - Nodrug       -43.0 24.9 54 -1.728  1.0000
##  Morp - Papv          29.8 24.9 54  1.198  1.0000
##  Morp - Pento        -28.5 24.9 54 -1.146  1.0000
##  Morp - Placebo      -56.8 24.9 54 -2.283  0.5543
##  Morp - Tripel       -19.2 24.9 54 -0.772  1.0000
##  Nodrug - Papv        72.8 24.9 54  2.926  0.1053
##  Nodrug - Pento       14.5 24.9 54  0.583  1.0000
##  Nodrug - Placebo    -13.8 24.9 54 -0.555  1.0000
##  Nodrug - Tripel      23.8 24.9 54  0.957  1.0000
##  Papv - Pento        -58.3 24.9 54 -2.343  0.4794
##  Papv - Placebo      -86.6 24.9 54 -3.481  0.0210
##  Papv - Tripel       -49.0 24.9 54 -1.969  1.0000
##  Pento - Placebo     -28.3 24.9 54 -1.137  1.0000
##  Pento - Tripel        9.3 24.9 54  0.374  1.0000
##  Placebo - Tripel     37.6 24.9 54  1.511  1.0000
##
## Results are averaged over the levels of: Patient
## P value adjustment: bonferroni method for 21 tests
```

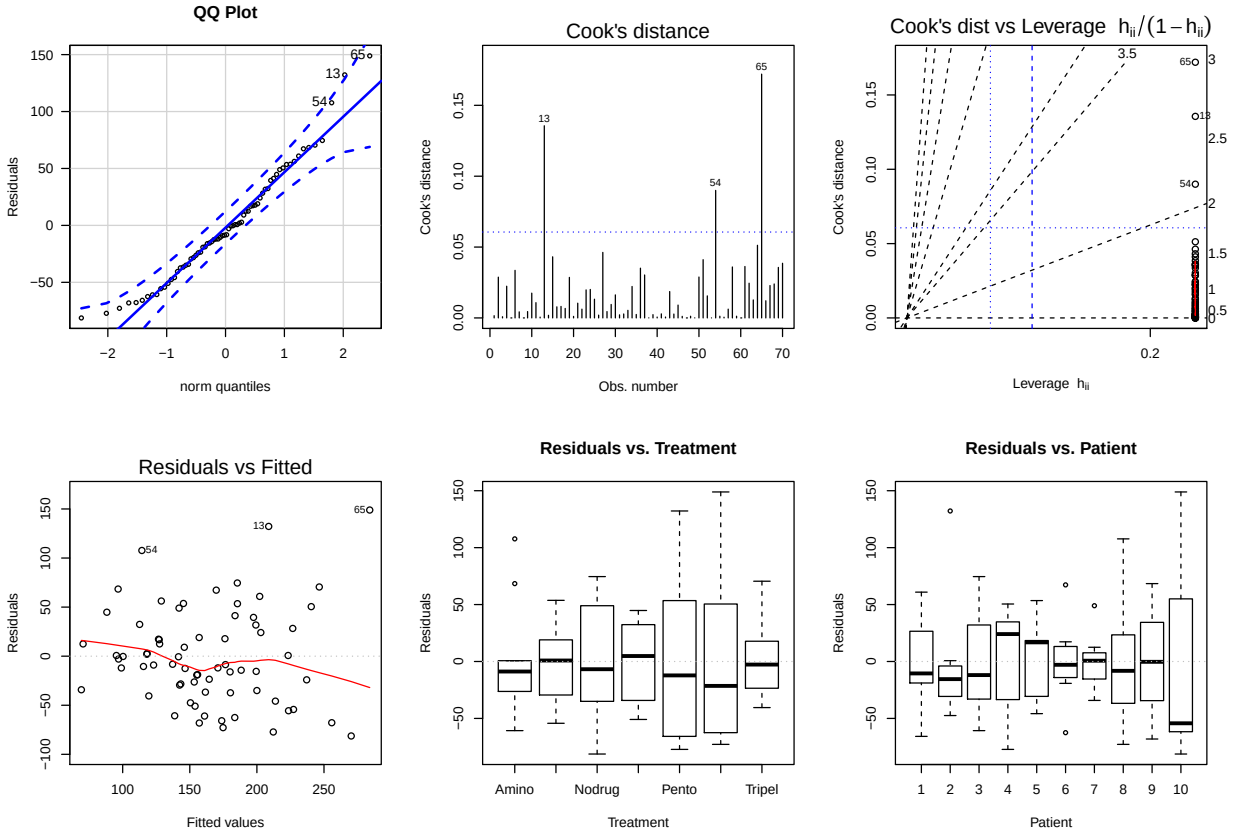
```
# Plot means and contrasts
p <- plot(cont_s, comparisons = TRUE, adjust = "bonf")  # adjust = "tukey" is default
p <- p + labs(title = "Tukey-adjusted Treatment contrasts")
p <- p + theme_bw()
print(p)
```



The Bonferroni comparisons for Treatment suggest that Papaverine induces a lower mean itching time than Placebo. All the other comparisons of treatments are insignificant. The comparison of Patient blocks is of less interest.

Diagnostic Analysis for the RB Analysis A diagnostic analysis of ANOVA-type models, including the RB model, is easily performed using the `lm()` output. The normal quantile (or QQ-plot) shows the residual distribution is slightly skewed to the right, in part, due to three cases that are not fitted well by the model (the outliers in the boxplots). Except for these cases, which are also the most influential cases (Cook's distance), the plot of the studentized residuals against fitted values shows no gross abnormalities.

```
# plot diagnostics
lm_diag_plots(lm_s_t_p, sw_plot_set = "simple")
```



Although the F-test for comparing treatments is not overly sensitive to modest deviations from normality, I will present a non-parametric analysis as a backup, to see whether similar conclusions are reached about the treatments.

Non-parametric Analysis of a RB Experiment Milton Friedman developed a non-parametric test for comparing treatments in an unreplicated randomized block design where the normality assumption may be violated. An unreplicated complete block design has exactly one observation in y for each combination of levels of groups and blocks. The null hypothesis is that apart from an effect of blocks, the location parameter of y is the same in each of the groups.

The output suggests significant differences among treatments, which supports the earlier conclusion.

```
# Friedman test for differences between groups conditional on blocks.
#   The formula is of the form a ~ b | c,
#     where a, b and c give the data values (a)
#     and corresponding groups (b) and blocks (c), respectively.
friedman.test(Seconds ~ Treatment | Patient, data = dat_itch_long)
```

```
##
##  Friedman rank sum test
##
## data:  Seconds and Treatment and Patient
## Friedman chi-squared = 14.887, df = 6, p-value = 0.02115
```

```
# Quade test is very similar to the Friedman test (compare the help pages).
quade.test(Seconds ~ Treatment | Patient, data = dat_itch_long)
```

```
##
##  Quade test
##
## data:  Seconds and Treatment and Patient
## Quade F = 3.7321, num df = 6, denom df = 54, p-value =
## 0.003542
```

5.2 Extending the One-Factor Design to Multiple Factors

The CRD (completely randomized design) for comparing insecticides below varies the levels of one factor (insecticide), while controlling other factors that influence survival time. The inferences from the one-way ANOVA apply to beetles with a given age from the selected strain that might be given the selected concentration of the insecticides. Any generalization of the conclusions to other situations must be justified scientifically, typically through further experimentation.

There are several ways to broaden the scope of the study. For example, several strains of beetles or several concentrations of the insecticide might be used. For simplicity, consider a simple two-factor experiment where three concentrations (Low, Medium, and High) are applied with each of the four insecticides. This is a completely crossed **two-factor experiment** where each of the $4 \times 3 = 12$ combinations of the two factors (insecticide and dose) are included in the comparison of survival times. With this experiment, the scientist can compare insecticides, compare concentrations, and check for an interaction between dose and insecticide.

Assuming that 48 beetles are available, the scientist would randomly assign them to the 12 experimental groups, giving prespecified numbers of beetles to the 12 groups. For simplicity, assume that the experiment is **balanced**, that is, the same number of beetles (4) is assigned to each group ($12 \times 4 = 48$). This is a CRD with two factors.

5.2.1 Example: Beetle insecticide two-factor design

The data below were collected using the experimental design just described. The table gives survival times of groups of four beetles randomly allocated to twelve treatment groups obtained by crossing the levels of four insecticides (A, B, C, D) at each of three concentrations of the insecticides (1=Low, 2=Medium, 3=High). This is a balanced 4-by-3 **factorial** design (two-factor design) that is replicated four times. The unit of measure for the survival times is 10 hours, that is, 0.3 is a survival time of 3 hours.

```
#### Example: Beetles
dat_beetles <-
  read_table2("http://statacumen.com/teach/ADA2/notes/ADA2_notes_Ch05_beetles.dat") %>%
  mutate(
    # make dose a factor variable and label the levels
    dose = factor(dose
                  , levels = c(1, 2, 3)
                  , labels = c("low", "medium", "high")
                  , ordered = TRUE
                  )
  , insecticide = factor(insecticide)
  )
```

```
## Parsed with column specification:
## cols(
##   dose = col_double(),
##   insecticide = col_character(),
##   t1 = col_double(),
##   t2 = col_double(),
##   t3 = col_double(),
##   t4 = col_double()
## )
```

	dose	insecticide	t1	t2	t3	t4
1	low	A	0.31	0.45	0.46	0.43
2	low	B	0.82	1.10	0.88	0.72
3	low	C	0.43	0.45	0.63	0.76
4	low	D	0.45	0.71	0.66	0.62
5	medium	A	0.36	0.29	0.40	0.23
6	medium	B	0.92	0.61	0.49	1.24
7	medium	C	0.44	0.35	0.31	0.40
8	medium	D	0.56	1.02	0.71	0.38
9	high	A	0.22	0.21	0.18	0.23
10	high	B	0.30	0.37	0.38	0.29
11	high	C	0.23	0.25	0.24	0.22
12	high	D	0.30	0.36	0.31	0.33

First we reshape the data to long format so each observation is its own row in the `data.frame` and indexed by the `dose` and `insecticide` variables.

```
dat_beetles_long <-
  dat_beetles %>%
  pivot_longer(
    cols      = starts_with("t")
  , names_to  = "number"
  , values_to = "hours10"
  ) %>%
  mutate(
    number = factor(number)
  )

str(dat_beetles_long)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame': 48 obs. of  4 variables:
##  $ dose       : Ord.factor w/ 3 levels "low"<"medium"<..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ insecticide: Factor w/ 4 levels "A","B","C","D": 1 1 1 1 2 2 2 2 3 3 ...
##  $ number     : Factor w/ 4 levels "t1","t2","t3",..: 1 2 3 4 1 2 3 4 1 2 ...
##  $ hours10    : num  0.31 0.45 0.46 0.43 0.82 1.1 0.88 0.72 0.43 0.45 ...
```

```
head(dat_beetles_long)
```

```
## # A tibble: 6 x 4
##   dose  insecticide number hours10
##   <ord> <fct>       <fct>    <dbl>
## 1 low   A           t1        0.31
## 2 low   A           t2        0.45
## 3 low   A           t3        0.46
## 4 low   A           t4        0.43
## 5 low   B           t1        0.82
## 6 low   B           t2        1.1
```

The basic unit of analysis is the **cell means**, which are the averages of

the 4 observations in each of the 12 treatment combinations. For example, in the table below, the sample mean survival for the 4 beetles given a low dose (dose=1) of insecticide A is 0.413. From the cell means we obtain the dose and insecticide **marginal means** by averaging over the levels of the other factor. For example, the marginal mean for insecticide A is the average of the cell means for the 3 treatment combinations involving insecticide A: $0.314 = (0.413 + 0.320 + 0.210)/3$.

```
sum_beetles <-
  dat_beetles_long %>%
  group_by(dose, insecticide) %>%
  summarize(
    m = mean(hours10)
  ) %>%
  pivot_wider(
    id_cols = insecticide
  , names_from = dose
  , values_from = m
  )
sum_beetles
```

```
## # A tibble: 4 x 4
##   insecticide   low medium  high
##   <fct>       <dbl>  <dbl> <dbl>
## 1 A           0.412  0.32  0.21
## 2 B           0.88   0.815 0.335
## 3 C           0.568  0.375 0.235
## 4 D           0.61   0.668 0.325
```

```
sum_beetles_margin1 <-
  dat_beetles_long %>%
  group_by(dose) %>%
  summarize(
    margin = mean(hours10)
  ) %>%
  pivot_wider(
    id_cols = NULL
  , names_from = dose
  , values_from = margin
  )
sum_beetles_margin1
```

```
## # A tibble: 1 x 3
##     low medium  high
##   <dbl>  <dbl> <dbl>
## 1 0.618  0.544 0.276
```

```
sum_beetles_margin2 <-
  dat_beetles_long %>%
```

```
  group_by(insecticide) %>%
  summarize(
    margin = mean(hours10)
  )
sum_beetles_margin2
```

```
## # A tibble: 4 x 2
##   insecticide margin
##   <fct>        <dbl>
## 1 A            0.314
## 2 B            0.677
## 3 C            0.392
## 4 D            0.534
```

```
sum_beetles_margin0 <-
  dat_beetles_long %>%
  summarize(
    margin = mean(hours10)
  )
sum_beetles_margin0
```

```
## # A tibble: 1 x 1
##   margin
##    <dbl>
## 1  0.479
```

```
# It's possible to combine all into a single table,
#   but it's probably not worth the effort
sum_beetles_table <-
  sum_beetles %>%
  bind_cols(
    sum_beetles_margin2 %>% select(margin)
  ) %>%
  bind_rows(
    data.frame(
      insecticide = "margin"
    , sum_beetles_margin1
    , sum_beetles_margin0
    )
  )
```

```
## Warning in bind_rows_(x, .id): Unequal factor levels: coercing to character
## Warning in bind_rows_(x, .id): binding character and factor vector, coercing into character
vector
## Warning in bind_rows_(x, .id): binding character and factor vector, coercing into character
vector
```

```
#knitr::kable(sum_beetles_table)
```

	insecticide	low	medium	high	margin
1	A	0.412	0.320	0.210	0.314
2	B	0.880	0.815	0.335	0.677
3	C	0.568	0.375	0.235	0.393
4	D	0.610	0.667	0.325	0.534
5	margin	0.618	0.544	0.276	0.479

Because the experiment is balanced, a marginal mean is the average of all observations that receive a given treatment. For example, the marginal mean for insecticide A is the average survival time for the 16 beetles given insecticide A.

Looking at the table of means, the insecticides have noticeably different mean survival times averaged over doses, with insecticide A having the lowest mean survival time averaged over doses. Similarly, higher doses tend to produce lower survival times. A more formal approach to analyzing the table of means is given in the next section.

5.2.2 The Interaction Model for a Two-Factor Experiment

Assume that you designed a **balanced** two-factor experiment with K responses at each combination of the I levels of factor 1 (F1) with the J levels of factor 2 (F2). The total number of responses is KIJ, or K times the IJ treatment combinations.

Let y_{ijk} be the k^{th} response at the i^{th} level of F1 and the j^{th} level of F2. A generic model for the experiment expresses y_{ijk} as a mean response plus a residual:

$$y_{ijk} = \mu_{ij} + e_{ijk},$$

where μ_{ij} is the population mean response for the treatment defined by the i^{th} level of F1 combined with the j^{th} level of F2. As in a one-way ANOVA, the responses within and across treatment groups are assumed to be independent, normally distributed, and have constant variance.

The **interaction model** expresses the population means as

$$\mu_{ij} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij},$$

where μ is a grand mean, α_i is the effect for the i^{th} level of F1, β_j is the effect for the j^{th} level of F2, and $(\alpha\beta)_{ij}$ is the interaction between the i^{th} level of F1 and the j^{th} level of F2. (Note that $(\alpha\beta)$ is an individual term distinct from α and β, $(\alpha\beta)$ is not their product.) The model is often written

$$y_{ijk} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + e_{ijk},$$

meaning

Response = Grand Mean + F1 effect + F2 effect + F1-by-F2 interaction + Residual.

The **additive model** having only main effects, no interaction terms, is $y_{ijk} = \mu + \alpha_i + \beta_j + e_{ijk}$, meaning

Response = Grand Mean + F1 effect + F2 effect + Residual.

The effects of F1 and F2 on the mean are additive.

Defining effects from cell means

The effects that define the population means and the usual hypotheses of interest can be formulated from the table of population means, given here.

	Level of F2				
Level of F1	1	2	$\cdots$	J	F1 marg
1	μ_{11}	μ_{12}	$\cdots$	μ_{1J}	$\bar{\mu}_{1\cdot}$
2	μ_{21}	μ_{22}	$\cdots$	μ_{2J}	$\bar{\mu}_{2\cdot}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
I	μ_{I1}	μ_{I2}	$\cdots$	μ_{IJ}	$\bar{\mu}_{I\cdot}$
F2 marg	$\bar{\mu}_{\cdot 1}$	$\bar{\mu}_{\cdot 2}$	$\cdots$	$\bar{\mu}_{\cdot J}$	$\bar{\mu}_{\cdot\cdot}$

The F1 **marginal population means** are averages within rows (over columns) of the table:

$$\bar{\mu}_{i\cdot} = \frac{1}{J}\sum_{c} \mu_{ic}.$$

The F2 marginal population means are averages within columns (over rows):

$$\bar{\mu}_{\cdot j} = \frac{1}{I}\sum_{r} \mu_{rj}.$$

The overall or **grand population mean** is the average of the cell means

$$\bar{\mu}_{\cdot\cdot} = \frac{1}{IJ}\sum_{rc} \mu_{rc} = \frac{1}{I}\sum_{i} \bar{\mu}_{i\cdot} = \frac{1}{J}\sum_{j} \bar{\mu}_{\cdot j}.$$

Using this notation, the effects in the interaction model are $\mu = \bar{\mu}_{\cdot\cdot}$, $\alpha_i = \bar{\mu}_{i\cdot} - \bar{\mu}_{\cdot\cdot}$, $\beta_j = \bar{\mu}_{\cdot j} - \bar{\mu}_{\cdot\cdot}$, and $(\alpha\beta)_{ij} = \mu_{ij} - \bar{\mu}_{i\cdot} - \bar{\mu}_{\cdot j} + \bar{\mu}_{\cdot\cdot}$. The effects sum to zero:

$$\sum_{i} \alpha_i = \sum_{j} \beta_j = \sum_{ij} (\alpha\beta)_{ij} = 0,$$

and satisfy $\mu_{ij} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij}$ (i.e., cell mean is sum of effects) required under the model.

The F1 and F2 effects are analogous to treatment effects in a one-factor experiment, except that here the treatment means are averaged over the levels of the other factor. The interaction effect will be interpreted later.

Estimating effects from the data

Let

$$\bar{y}_{ij} = \frac{1}{K}\sum_k y_{ijk} \qquad \text{and} \qquad s_{ij}^2$$

be the sample mean and variance, respectively, for the K responses at the i^{th} level of F1 and the j^{th} level of F2. Inferences about the population means are based on the table of sample means:

	Level of F2				
Level of F1	1	2	$\cdots$	J	F1 marg
1	$\bar{y}_{11}$	$\bar{y}_{12}$	$\cdots$	$\bar{y}_{1J}$	$\bar{y}_{1\cdot}$
2	$\bar{y}_{21}$	$\bar{y}_{22}$	$\cdots$	$\bar{y}_{2J}$	$\bar{y}_{2\cdot}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
I	$\bar{y}_{I1}$	$\bar{y}_{I2}$	$\cdots$	$\bar{y}_{IJ}$	$\bar{y}_{I\cdot}$
F2 marg	$\bar{y}_{\cdot 1}$	$\bar{y}_{\cdot 2}$	$\cdots$	$\bar{y}_{\cdot J}$	$\bar{y}_{\cdot\cdot}$

The F1 **marginal sample means** are averages within rows of the table:

$$\bar{y}_{i\cdot} = \frac{1}{J}\sum_c \bar{y}_{ic\cdot}$$

The F2 marginal sample means are averages within columns:

$$\bar{y}_{\cdot j} = \frac{1}{I}\sum_r \bar{y}_{rj\cdot}$$

Finally, $\bar{y}_{\cdot\cdot}$ is the average of the cell sample means:

$$\bar{y}_{\cdot\cdot} = \frac{1}{IJ}\sum_{ij} \bar{y}_{ij} = \frac{1}{I}\sum_i \bar{y}_{i\cdot} = \frac{1}{J}\sum_j \bar{y}_{\cdot j}.$$

The sample sizes in each of the IJ treatment groups are equal (K), so $\bar{y}_{i\cdot}$ is the sample average of all responses at the i^{th} level of F1, $\bar{y}_{\cdot j}$ is the sample average of all responses at the j^{th} level of F2, and $\bar{y}_{\cdot\cdot}$ is the average response in the experiment.

Under the interaction model, the estimated population mean for the $(i,j)^{th}$ cell is the observed cell mean: $\hat{\mu}_{ij} = \bar{y}_{ij}$. This can be partitioned into estimated effects

$$\begin{aligned}
\hat{\mu} &= \bar{y}_{\cdot\cdot} \quad \text{the estimated grand mean} \\
\hat{\alpha}_i &= \bar{y}_{i\cdot} - \bar{y}_{\cdot\cdot} \quad \text{the estimated F1 effect } i = 1, 2, \ldots, I \\
\hat{\beta}_j &= \bar{y}_{\cdot j} - \bar{y}_{\cdot\cdot} \quad \text{the estimated F2 effect } j = 1, 2, \ldots, J \\
\widehat{(\alpha\beta)}_{ij} &= \bar{y}_{ij} - \bar{y}_{i\cdot} - \bar{y}_{\cdot j} + \bar{y}_{\cdot\cdot} \quad \text{the estimated cell interaction}
\end{aligned} \tag{5.1}$$

that satisfy

$$\hat{\mu}_{ij} = \hat{\mu} + \hat{\alpha}_i + \hat{\beta}_j + \widehat{(\alpha\beta)}_{ij}.$$

The ANOVA table

The ANOVA table for a balanced two-factor design decomposes the total variation in the data, as measured by the Total SS, into components that measure the variation of marginal sample means for F1 and F2 (the F1 SS and F2 SS), a component that measures the degree to which the factors interact (the F1-by-F2 Interaction SS), and a component that pools the sample variances across the IJ samples (the Error SS). Each SS has a df, given in the following ANOVA table. As usual, the MS for each source of variation is the corresponding SS divided by the df. The MS Error estimates the common population variance for the IJ treatments.

Source	*df*	SS	MS
F1	$I-1$	$KJ\sum_i(\bar{y}_{i\cdot} - \bar{y}_{\cdot\cdot})^2$	MS F1=SS/df
F2	$J-1$	$KI\sum_j(\bar{y}_{\cdot j} - \bar{y}_{\cdot\cdot})^2$	MS F2=SS/df
Interaction	$(I-1)(J-1)$	$K\sum_{ij}(y_{ij} - \bar{y}_{i\cdot} - \bar{y}_{\cdot j} + \bar{y}_{\cdot\cdot})^2$	MS Inter=SS/df
Error	$IJ(K-1)$	$(K-1)\sum_{ij} s_{ij}^2$	MSE=SS/df
Total	$IJK-1$	$\sum_{ijk}(y_{ijk} - \bar{y}_{\cdot\cdot})^2$.	

The standard tests in the two-factor analysis, and their interpretations are:

The test of no F1 effect: $H_0 : \alpha_1 = \cdots = \alpha_I = 0$ is equivalent to testing $H_0 : \bar{\mu}_{1\cdot} = \bar{\mu}_{2\cdot} = \cdots = \bar{\mu}_{I\cdot}$. The absence of an F1 effect implies that each level of F1 has the same population mean response **when the means are averaged over levels of** F2. The test for no F1 effect is based on MS F1/MS Error, which is compared to the upper tail of an F-distribution with numerator and denominator df of $I - 1$ and $IJ(K - 1)$, respectively. H_0 is rejected when the F1 marginal means $\bar{y}_{i\cdot}$ vary significantly relative to the within sample variation. Equivalently, H_0 is rejected when the sum of squared F1 effects (between sample variation) is large relative to the within sample variation.

The test of no F2 effect: $H_0 : \beta_1 = \cdots = \beta_J = 0$ is equivalent to testing $H_0 : \bar{\mu}_{\cdot 1} = \bar{\mu}_{\cdot 2} = \cdots = \bar{\mu}_{\cdot J}$. The absence of a F2 effect implies that each level of F2 has the same population mean response **when the means are averaged over levels of** F1. The test for no F2 effect is based on MS F2/MS Error, which is compared to an F-distribution with numerator and denominator df of $J - 1$ and $IJ(K - 1)$, respectively. H_0 is rejected when the F2 marginal means $\bar{y}_{\cdot j}$ vary significantly relative to the within sample variation. Equivalently, H_0 is rejected when the sum of squared F2 effects (between sample variation) is large relative to the within sample variation.

The test of no interaction: $H_0 : (\alpha\beta)_{ij} = 0$ for all i and j is based on MS Interact/MS Error, which is compared to an F-distribution with numerator and denominator df of $(I - 1)(J - 1)$ and $IJ(K - 1)$, respectively.

The interaction model places no restrictions on the population means μ_{ij}. Since the population means can be arbitrary, the interaction model can be viewed as a one factor model with IJ treatments. One connection between the two ways of viewing the two-factor analysis is that the F1, F2, and Interaction SS for the two-way interaction model sum to the Treatment or Model SS for comparing the IJ treatments. The Error SS for the two-way interaction model is identical to the Error SS for a one-way ANOVA of the IJ treatments. An overall test of no differences in the IJ population means is part of the two-way

analysis.

I always summarize the data using the cell and marginal means instead of the estimated effects, primarily because means are the basic building blocks for the analysis. My discussion of the model and tests emphasizes both approaches to help you make the connection with the two ways this material is often presented in texts.

Understanding interaction

To understand interaction, suppose you (conceptually) plot the means in each row of the population table, giving what is known as the **population mean profile** plot. The F1 marginal population means average the population means within the F1 profiles. At each F2 level, the F2 marginal mean averages the population cell means across F1 profiles.

No interaction is present if the plot has perfectly **parallel** F1 profiles, as in the plot below for a 3×5 experiment. The levels of F1 and F2 **do not interact**. That is,

parallel profiles $\Leftrightarrow$ $\mu_{ij} - \mu_{hj}$ is independent of j for each i and h
difference between levels of F1 does not depend on level of F2

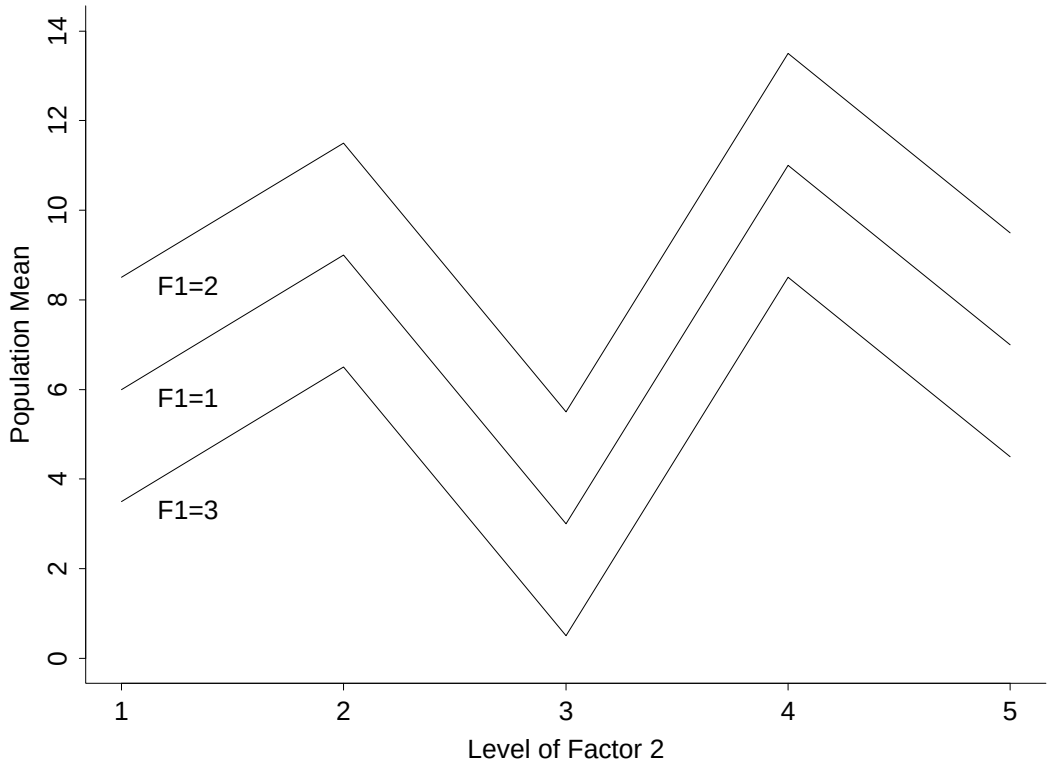
$\Leftrightarrow$ $\mu_{ij} - \bar{\mu}_{i\cdot} = \mu_{hj} - \bar{\mu}_{h\cdot}$ for all i, j, h
difference between level of F2 j and F2 mean does not depend on level o

$\Leftrightarrow$ $\mu_{ij} - \bar{\mu}_{i\cdot} = \bar{\mu}_{\cdot j} - \bar{\mu}_{\cdot\cdot}$ for all i, j
difference between level of F2 j and F2 mean is the same for all levels of

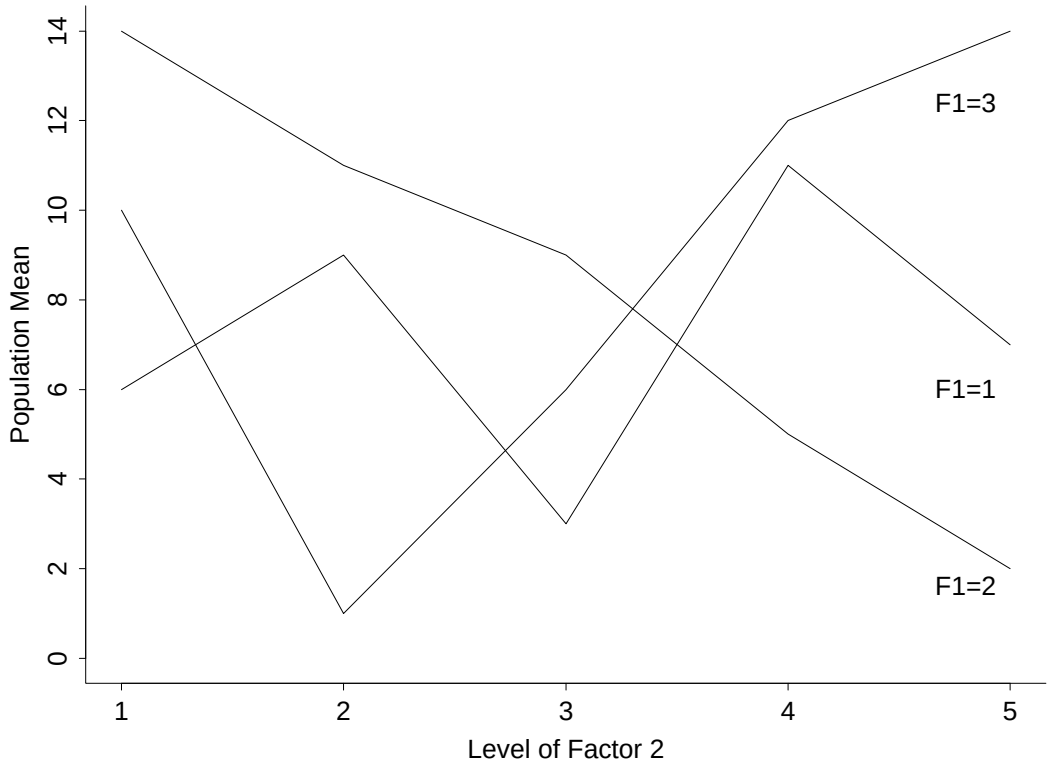
$\Leftrightarrow$ $\mu_{ij} - \bar{\mu}_{i\cdot} - \bar{\mu}_{\cdot j} + \bar{\mu}_{\cdot\cdot} = 0$ for all i, j
interaction effect is 0

$\Leftrightarrow$ $(\alpha\beta)_{ij} = 0$ for all i, j
interaction effect is 0

$\Leftrightarrow$ no interaction term in model.



Interaction is present if the profiles are not **perfectly parallel**. An example of a profile plot for two-factor experiment (3×5) with interaction is given below.



The roles of F1 and F2 can be reversed in these plots without changing the assessment of a presence or absence of interaction. It is often helpful to view the interaction plot from both perspectives.

A qualitative check for interaction can be based on the **sample means profile plot**, but keep in mind that profiles of sample means are never perfectly parallel even when the factors do not interact in the population. The Interaction SS measures the extent of non-parallelism in the sample mean profiles. In particular, the Interaction SS is zero when the sample mean profiles are perfectly parallel because $\widehat{(\alpha\beta)}_{ij} = 0$ for all i and j.

5.2.3 Example: Survival Times of Beetles

First we generate cell means and a sample means profile plot (interaction plot) for the beetle experiment. The 12 treatment cell means we calculated. Three variables were needed to represent each response in the data set: dose (1-3, categorical), insecticide (A-D, categorical), and time (the survival time).

As noted earlier, the insecticides have noticeably different mean survival times averaged over doses, with insecticide A having the lowest mean. Similarly, higher doses tend to produce lower survival times. The sample means profile plot shows some evidence of interaction.

```
# Calculate the cell means for each (dose, insecticide) combination
sum_beetles_mean <-
  dat_beetles_long %>%
  summarize(
    m = mean(hours10)
  )
sum_beetles_mean
```

```
## # A tibble: 1 x 1
##       m
##   <dbl>
## 1 0.479
```

```
sum_beetles_mean_d <-
  dat_beetles_long %>%
  group_by(dose) %>%
  summarize(
    m = mean(hours10)
  )
sum_beetles_mean_d
```

```
## # A tibble: 3 x 2
##   dose       m
##   <ord>  <dbl>
## 1 low    0.618
## 2 medium 0.544
## 3 high   0.276
```

```
sum_beetles_mean_i <-
  dat_beetles_long %>%
  group_by(insecticide) %>%
  summarize(
    m = mean(hours10)
  )
sum_beetles_mean_i
```

```
## # A tibble: 4 x 2
##   insecticide     m
```

```
##   <fct>       <dbl>
## 1 A           0.314
## 2 B           0.677
## 3 C           0.392
## 4 D           0.534
```

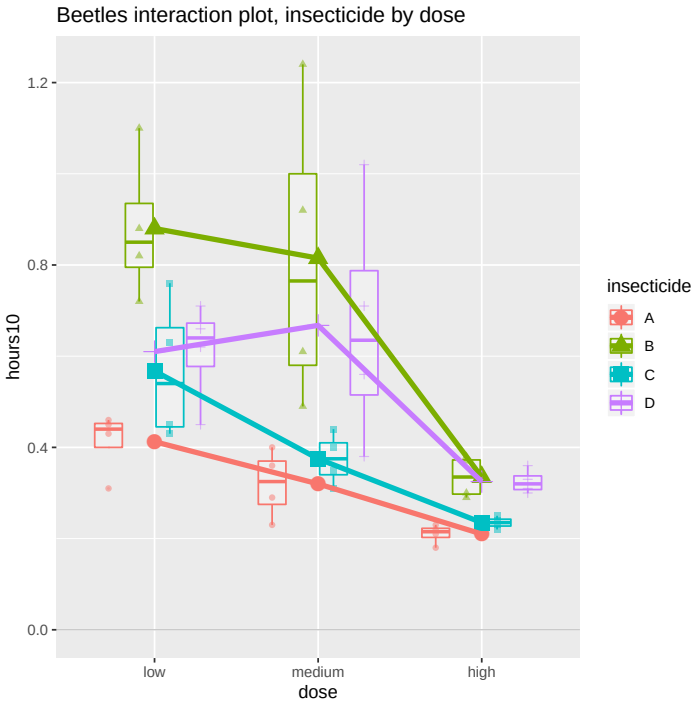
```
sum_beetles_mean_di <-
  dat_beetles_long %>%
  group_by(dose, insecticide) %>%
  summarize(
    m = mean(hours10)
  )
sum_beetles_mean_di
```

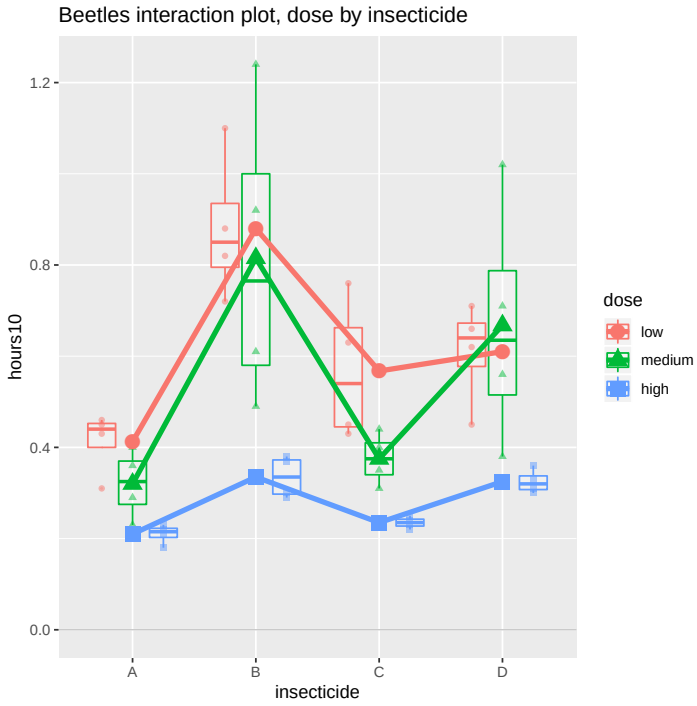
```
## # A tibble: 12 x 3
## # Groups:   dose [3]
##    dose   insecticide     m
##    <ord>  <fct>       <dbl>
##  1 low    A           0.412
##  2 low    B           0.88
##  3 low    C           0.568
##  4 low    D           0.61
##  5 medium A           0.32
##  6 medium B           0.815
##  7 medium C           0.375
##  8 medium D           0.668
##  9 high   A           0.21
## 10 high   B           0.335
## 11 high   C           0.235
## 12 high   D           0.325
```

```
# Interaction plots, ggplot

p <- ggplot(dat_beetles_long, aes(x = dose, y = hours10, colour = insecticide, shape = insecti
p <- p + geom_hline(aes(yintercept = 0), colour = "black"
                    , linetype = "solid", size = 0.2, alpha = 0.3)
p <- p + geom_boxplot(alpha = 0.25, outlier.size=0.1)
p <- p + geom_point(alpha = 0.5, position=position_dodge(width=0.75))
p <- p + geom_point(data = sum_beetles_mean_di, aes(y = m), size = 4)
p <- p + geom_line(data = sum_beetles_mean_di, aes(y = m, group = insecticide), size = 1.5)
p <- p + labs(title = "Beetles interaction plot, insecticide by dose")
print(p)

p <- ggplot(dat_beetles_long, aes(x = insecticide, y = hours10, colour = dose, shape = dose))
p <- p + geom_hline(aes(yintercept = 0), colour = "black"
                    , linetype = "solid", size = 0.2, alpha = 0.3)
p <- p + geom_boxplot(alpha = 0.25, outlier.size=0.1)
p <- p + geom_point(alpha = 0.5, position=position_dodge(width=0.75))
p <- p + geom_point(data = sum_beetles_mean_di, aes(y = m), size = 4)
p <- p + geom_line(data = sum_beetles_mean_di, aes(y = m, group = dose), size = 1.5)
p <- p + scale_colour_hue()  # ordered factor gives bad colors, reset color for dose
```

```
p <- p + labs(title = "Beetles interaction plot, dose by insecticide")
print(p)
```

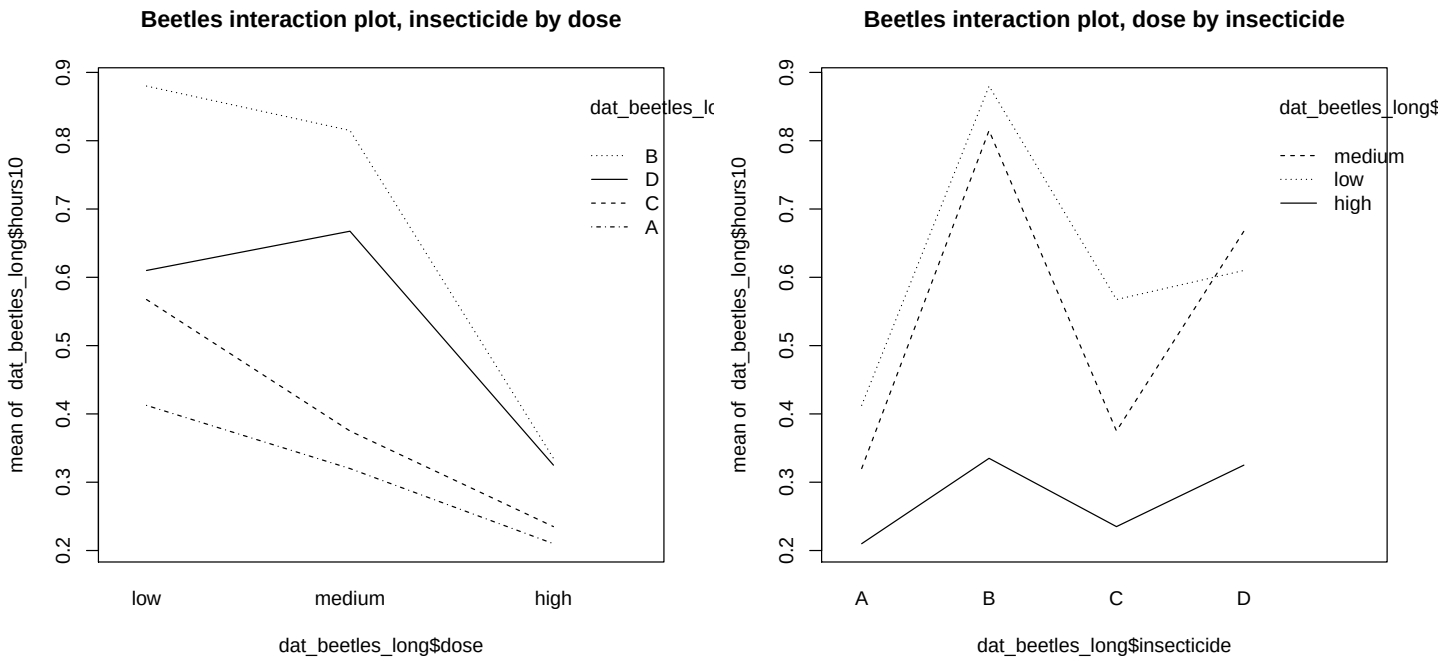
```
## Warning:  Using shapes for an ordinal variable is not advised
```





Base R has interaction plots, too.

```
# Interaction plots, base graphics
interaction.plot(dat_beetles_long$dose, dat_beetles_long$insecticide, dat_beetles_long$hours10
  , main = "Beetles interaction plot, insecticide by dose")
interaction.plot(dat_beetles_long$insecticide, dat_beetles_long$dose, dat_beetles_long$hours10
  , main = "Beetles interaction plot, dose by insecticide")
```



In the `lm()` function below we specify a first-order model with interactions, including the main effects and two-way interactions. The interaction between dose and insecticide is indicated with `dose:insecticide`. The shorthand `dose*insecticide` expands to "`dose + insecticide + dose:insecticide`" for this first-order model.

The F-test at the bottom of the `summary()` tests for no differences among the population mean survival times for the 12 dose and insecticide combinations. The p-value of < 0.0001 strongly suggests that the population mean survival times are not all equal.

The next summary at the top gives two partitionings of the one-way ANOVA Treatment SS into the SS for Dose, Insecticide, and the Dose by Insecticide interaction. The Mean Squares, F-statistics and p-values for testing these effects are given. The p-values for the F-statistics indicate that the dose and insecticide effects are significant at the 0.01 level. The F-test for no dose by insecticide interaction is not significant at the 0.10 level (p-value=0.112). Thus, the interaction seen in the profile plot of the sample means might be due solely to chance or sampling variability.

```
lm_h_d_i_di <-
  lm(
    hours10 ~ dose + insecticide + dose:insecticide
```

```
  , data = dat_beetles_long
  )
# lm_h_d_i_di <- lm(hours10 ~ dose*insecticide, data = dat_beetles_long) # equivalent
library(car)
Anova(lm_h_d_i_di, type=3)
```

```
## Anova Table (Type III tests)
##
## Response: hours10
##                   Sum Sq Df F value    Pr(>F)
## (Intercept)      1.18441  1 53.2501 1.343e-08 ***
## dose             0.08222  2  1.8482    0.1722
## insecticide      0.92121  3 13.8056 3.777e-06 ***
## dose:insecticide 0.25014  6  1.8743    0.1123
## Residuals        0.80073 36
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(lm_h_d_i_di)
```

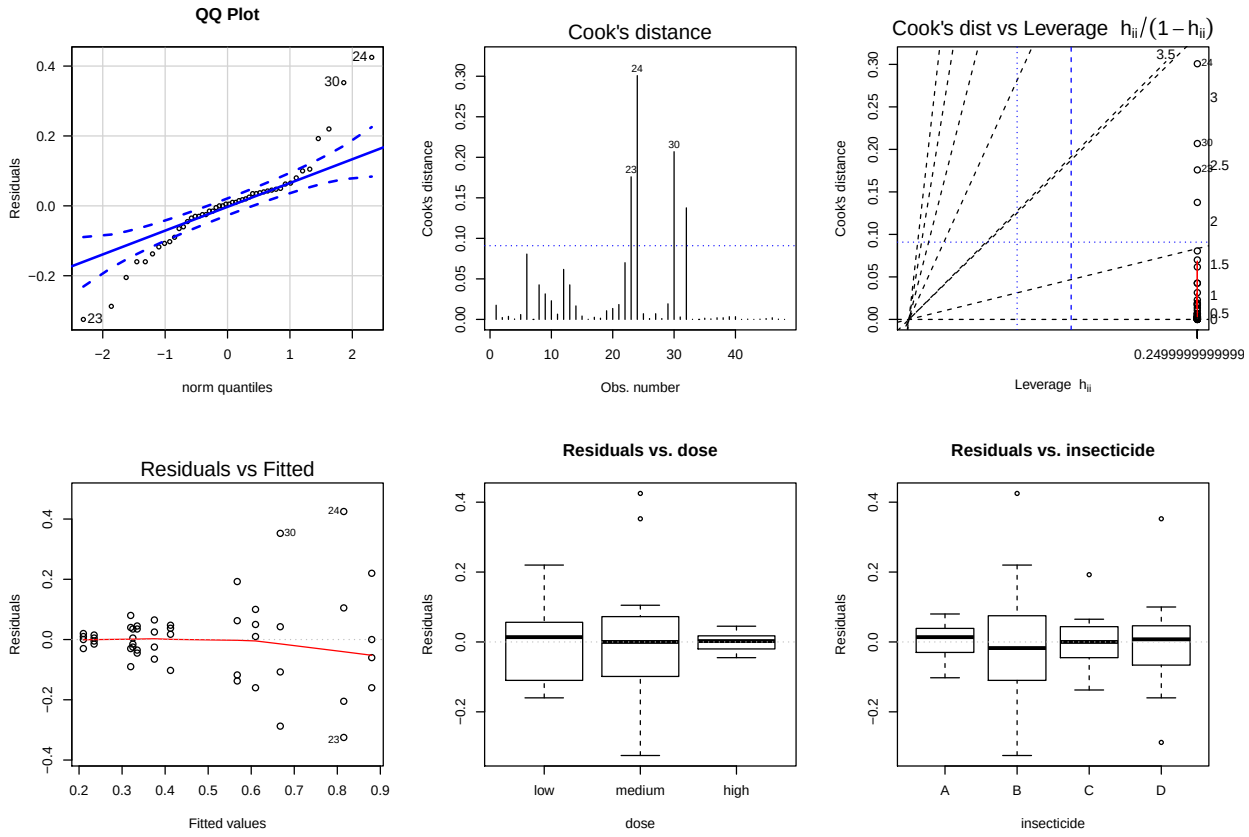
```
##
## Call:
## lm(formula = hours10 ~ dose + insecticide + dose:insecticide,
##     data = dat_beetles_long)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.32500 -0.04875  0.00500  0.04313  0.42500
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           0.314167   0.043053   7.297 1.34e-08 ***
## dose.L               -0.143189   0.074569  -1.920 0.062781 .
## dose.Q               -0.007144   0.074569  -0.096 0.924204
## insecticideB          0.362500   0.060886   5.954 8.01e-07 ***
## insecticideC          0.078333   0.060886   1.287 0.206458
## insecticideD          0.220000   0.060886   3.613 0.000916 ***
## dose.L:insecticideB -0.242184   0.105457  -2.297 0.027572 *
## dose.Q:insecticideB -0.162279   0.105457  -1.539 0.132594
## dose.L:insecticideC -0.091924   0.105457  -0.872 0.389164
## dose.Q:insecticideC  0.028577   0.105457   0.271 0.787950
## dose.L:insecticideD -0.058336   0.105457  -0.553 0.583562
## dose.Q:insecticideD -0.156155   0.105457  -1.481 0.147374
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1491 on 36 degrees of freedom
## Multiple R-squared:  0.7335,Adjusted R-squared:  0.6521
## F-statistic:  9.01 on 11 and 36 DF,  p-value: 1.986e-07
```

Diagnostic plots show the following features. The normal quantile plot

shows an "S" shape rather than a straight line, suggesting the residuals are not normal, but have higher kurtosis (more peaky) than a normal distribution. The residuals vs the fitted (predicted) values show that the higher the predicted value the more variability (horn shaped). The plot of the Cook's distances indicate a few influential observations. *For the sake of the example, we will ignore the model assumption deviations for now and address them later in Section 5.2.7. In practice, you would stop here and address the deviations prior to model selection, as we'll do in Section 5.2.7.*

```
# plot diagnistics
lm_diag_plots(lm_h_d_i_di, sw_plot_set = "simple")
```

```
## Warning in if ((class(fit$model[, var_names[i_plot]]) %in% c("numeric", :  the condition has length > 1 and only the first element
will be used
```



Since the interaction is not significant, I'll drop the interaction term and fit the additive model with main effects only. I update the model by removing the interaction term.

```
lm_h_d_i <- update(lm_h_d_i_di, ~ . - dose:insecticide )
library(car)
Anova(lm_h_d_i, type=3)
```

```
## Anova Table (Type III tests)
##
```

```
## Response: hours10
##              Sum Sq Df F value    Pr(>F)
## (Intercept) 1.18441  1  47.337 2.157e-08 ***
## dose        1.03301  2  20.643 5.704e-07 ***
## insecticide 0.92121  3  12.273 6.697e-06 ***
## Residuals   1.05086 42
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

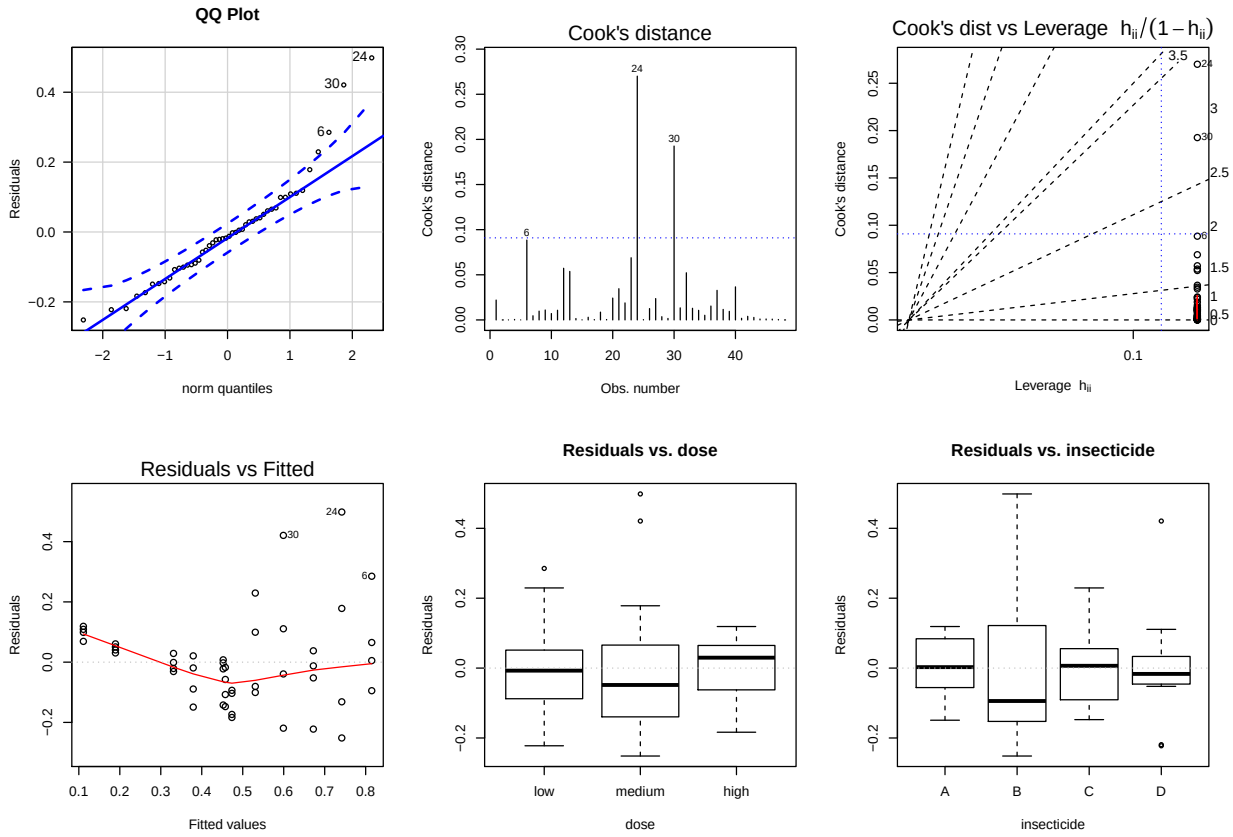
```
summary(lm_h_d_i)
```

```
##
## Call:
## lm(formula = hours10 ~ dose + insecticide, data = dat_beetles_long)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.25167 -0.09625 -0.01490  0.06177  0.49833
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.31417    0.04566   6.880 2.16e-08 ***
## dose.L       -0.24130    0.03954  -6.102 2.83e-07 ***
## dose.Q       -0.07961    0.03954  -2.013  0.05054 .
## insecticideB  0.36250    0.06458   5.614 1.43e-06 ***
## insecticideC  0.07833    0.06458   1.213  0.23189
## insecticideD  0.22000    0.06458   3.407  0.00146 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1582 on 42 degrees of freedom
## Multiple R-squared:  0.6503,Adjusted R-squared:  0.6087
## F-statistic: 15.62 on 5 and 42 DF,  p-value: 1.123e-08
```

As mentioned above, for the diagnostics below, we will ignore the model assumption deviations for now and address them later in Section 5.2.7.

```
# plot diagnostics
lm_diag_plots(lm_h_d_i, sw_plot_set = "simple")
```

```
## Warning in if ((class(fit$model[, var_names[i_plot]]) %in% c("numeric", :  the condition has length > 1 and only the first element
will be used
```



The Bonferroni multiple comparisons indicate which treatment effects are different.

```
# Contrasts to perform pairwise comparisons
cont_d <-
  emmeans::emmeans(
    lm_h_d_i
  , specs = "dose"
  )
cont_i <-
  emmeans::emmeans(
    lm_h_d_i
  , specs = "insecticide"
  )

# Means and CIs
#confint(cont_h, adjust = "bonferroni")

# Pairwise comparisons
cont_d %>% pairs(adjust = "bonf")   # adjust = "tukey" is default
```

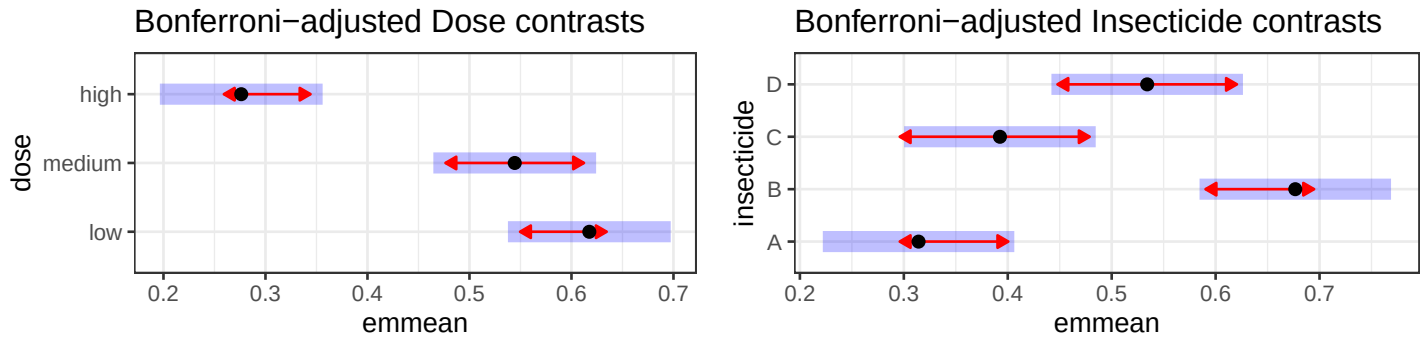
```
##  contrast      estimate     SE df t.ratio p.value
##  low - medium    0.0731 0.0559 42 1.308   0.5944
##  low - high      0.3412 0.0559 42 6.102   <.0001
##  medium - high   0.2681 0.0559 42 4.794   0.0001
##
```

```
## Results are averaged over the levels of: insecticide
## P value adjustment: bonferroni method for 3 tests
```

```
cont_i %>% pairs(adjust = "bonf")   # adjust = "tukey" is default
```

```
##  contrast estimate     SE df t.ratio p.value
##  A - B     -0.3625 0.0646 42 -5.614  <.0001
##  A - C     -0.0783 0.0646 42 -1.213  1.0000
##  A - D     -0.2200 0.0646 42 -3.407  0.0088
##  B - C      0.2842 0.0646 42  4.400  0.0004
##  B - D      0.1425 0.0646 42  2.207  0.1971
##  C - D     -0.1417 0.0646 42 -2.194  0.2030
##
## Results are averaged over the levels of: dose
## P value adjustment: bonferroni method for 6 tests
```

```
# Plot means and contrasts
p1 <- plot(cont_d, comparisons = TRUE)  #, adjust = "bonf")  # adjust = "tukey" is default
p1 <- p1 + labs(title = "Bonferroni-adjusted Dose contrasts")
p1 <- p1 + theme_bw()
print(p1)
p2 <- plot(cont_i, comparisons = TRUE)  #, adjust = "bonf")  # adjust = "tukey" is default
p2 <- p2 + labs(title = "Bonferroni-adjusted Insecticide contrasts")
p2 <- p2 + theme_bw()
print(p2)
```



Specifying Contrasts in `emmeans`

Note that testing multiple factors is of interest here. The code **above** corrects the p-values for all the tests done for each factor **separately**. However, we may want all of the tests over both factors are corrected **together**, in this case the Bonferroni-corrected significance level would be $(\alpha/(d+i))$, where d = number of dose comparisons and i = number of insecticide comparisons. We can specify a custom contrast using the code below. Custom contrasts will be a powerful tool that we will continue to use with multiple regression models.

```r
# Contrasts to perform pairwise comparisons
cont_di <-
  emmeans::emmeans(
    lm_h_d_i
  , specs = ~ dose + insecticide
  )

cont_di
```

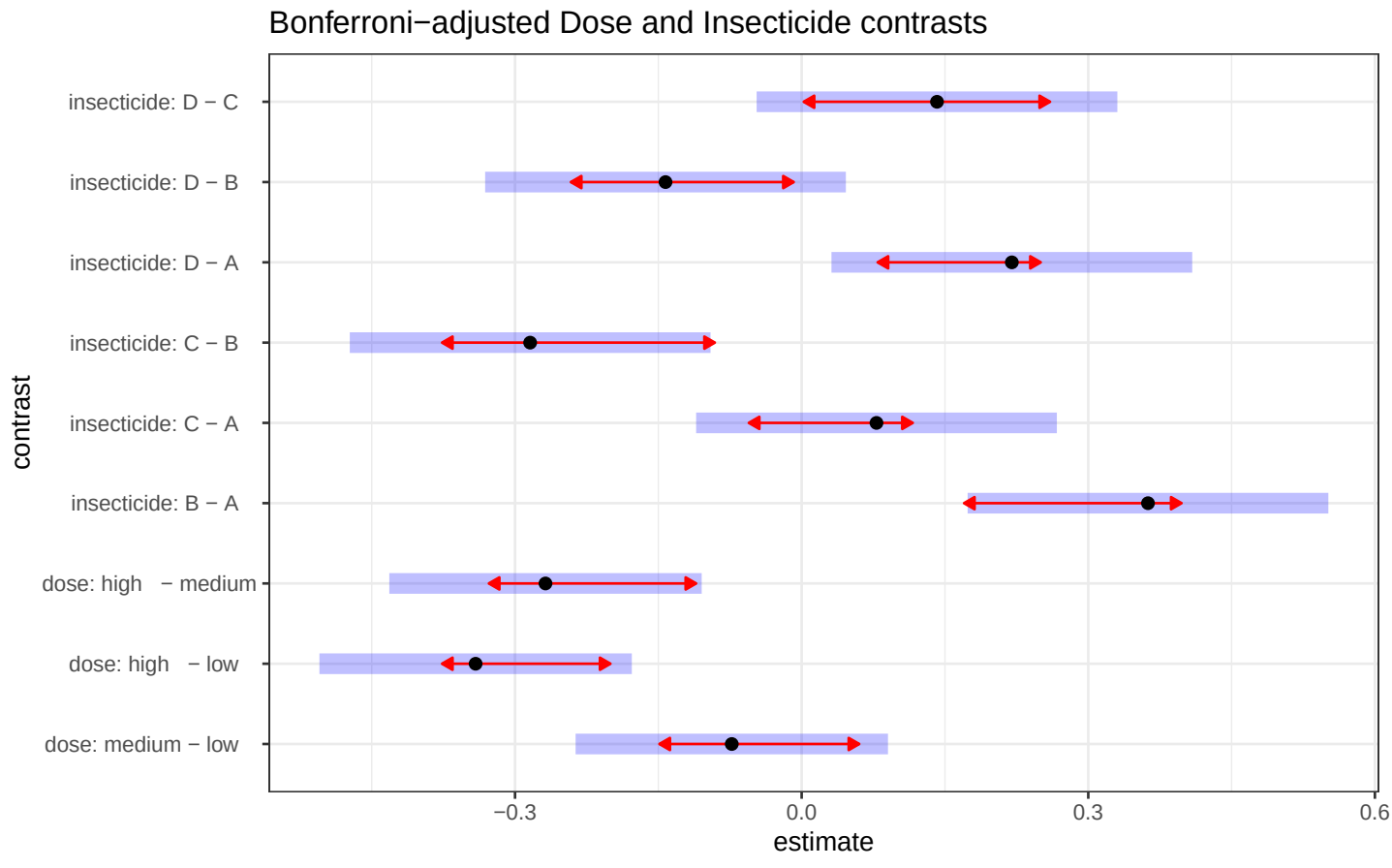
```
##  dose   insecticide emmean     SE df lower.CL upper.CL
##  low    A            0.452 0.0559 42  0.33943    0.565
##  medium A            0.379 0.0559 42  0.26631    0.492
##  high   A            0.111 0.0559 42 -0.00182    0.224
##  low    B            0.815 0.0559 42  0.70193    0.928
##  medium B            0.742 0.0559 42  0.62881    0.855
##  high   B            0.474 0.0559 42  0.36068    0.586
##  low    C            0.531 0.0559 42  0.41776    0.643
##  medium C            0.458 0.0559 42  0.34464    0.570
##  high   C            0.189 0.0559 42  0.07651    0.302
##  low    D            0.672 0.0559 42  0.55943    0.785
##  medium D            0.599 0.0559 42  0.48631    0.712
##  high   D            0.331 0.0559 42  0.21818    0.444
##
## Confidence level used: 0.95
```

```r
# define contrasts
#   indicate each position with a 1 and divide by the number of parameters included
c_dose_low      = c(1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0) / 4
c_dose_medium   = c(0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0) / 4
c_dose_high     = c(0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1) / 4
c_insecticide_A = c(1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0) / 3
c_insecticide_B = c(0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0) / 3
c_insecticide_C = c(0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0) / 3
c_insecticide_D = c(0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1) / 3

# Specified contrasts performed together
cont_di_c <-
  emmeans::contrast(
    cont_di
  , method = list(
      "dose: medium - low   " = c_dose_medium - c_dose_low
    , "dose: high   - low   " = c_dose_high   - c_dose_low
    , "dose: high   - medium" = c_dose_high   - c_dose_medium
    , "insecticide: B - A   " = c_insecticide_B - c_insecticide_A
    , "insecticide: C - A   " = c_insecticide_C - c_insecticide_A
    , "insecticide: C - B   " = c_insecticide_C - c_insecticide_B
    , "insecticide: D - A   " = c_insecticide_D - c_insecticide_A
    , "insecticide: D - B   " = c_insecticide_D - c_insecticide_B
    , "insecticide: D - C   " = c_insecticide_D - c_insecticide_C
    )
```

```
  , adjust = "bonf"
  )
cont_di_c
```

```
##  contrast               estimate     SE df t.ratio p.value
##  dose: medium - low      -0.0731 0.0559 42 -1.308  1.0000
##  dose: high   - low      -0.3412 0.0559 42 -6.102  <.0001
##  dose: high   - medium   -0.2681 0.0559 42 -4.794  0.0002
##  insecticide: B - A       0.3625 0.0646 42  5.614  <.0001
##  insecticide: C - A       0.0783 0.0646 42  1.213  1.0000
##  insecticide: C - B      -0.2842 0.0646 42 -4.400  0.0007
##  insecticide: D - A       0.2200 0.0646 42  3.407  0.0131
##  insecticide: D - B      -0.1425 0.0646 42 -2.207  0.2957
##  insecticide: D - C       0.1417 0.0646 42  2.194  0.3045
##
## P value adjustment: bonferroni method for 9 tests
```

```
# Plot means and contrasts
p <- plot(cont_di_c, comparisons = TRUE, adjust = "bonf")  # adjust = "tukey" is default
p <- p + labs(title = "Bonferroni-adjusted Dose and Insecticide contrasts")
p <- p + theme_bw()
print(p)
```



Interpretation of the Dose and Insecticide Effects

The interpretation of the dose and insecticide **main effects** depends on whether interaction is present. The distinction is important, so I will give both interpretations to emphasize the differences. Given the test for interaction, I would likely summarize the main effects assuming no interaction.

The average survival time decreases as the dose increases, with estimated mean survival times of 0.618, 0.544, and 0.276, respectively. A Bonferroni comparison shows that the population mean survival time for the high dose (averaged over insecticides) is significantly less than the population mean survival times for the low and medium doses (averaged over insecticides). The two lower doses are not significantly different from each other. This leads to two dose groups:

```
Dose:          1=Low  2=Med  3=Hig
Marg Mean:     0.618  0.544  0.276
Groups:        ------------  -----
```

If dose and insecticide **interact**, you can conclude that beetles given a high dose of the insecticide typically survive for shorter periods of time **averaged over insecticides**. You can not, in general, conclude that the highest dose yields the lowest survival time **regardless** of insecticide. For example, the difference in the medium and high dose marginal means (0.544 - 0.276 = 0.268) estimates the typical decrease in survival time achieved by using the high dose instead of the medium dose, averaged over insecticides. If the two factors interact, then the difference in mean times between the medium and high doses on a given insecticide may be significantly greater than 0.268, significantly less than 0.268, or even negative. In the latter case the medium dose would be **better** than the high dose for the given insecticide, even though the high dose gives better performance averaged over insecticides. An interaction forces you to use the cell means to decide which combination of dose and insecticide gives the best results (and the multiple comparisons as they were done above do not give multiple comparisons of cell means; a single factor variable combining both factors would need to be created). Of course, our profile plot tells us that this hypothetical situation is probably not tenable here, but it could be so when a significant interaction is present.

If dose and insecticide **do not interact**, then the difference in marginal dose means averaged over insecticides also estimates the difference in population mean survival times between two doses, **regardless of the insecticide**. This follows from the parallel profiles definition of no interaction. Thus, the difference in the medium and high dose marginal means (0.544 - 0.276 = 0.268) estimates the expected decrease in survival time anticipated from using the high dose instead of the medium dose, **regardless of the insecticide** (and hence also when averaged over insecticides). A practical implication of no interaction is that you can conclude that the high dose is best, regardless of the insecticide used. The difference in marginal means for two doses estimates the difference in average survival expected, regardless of the insecticide.

An ordering of the mean survival times on the four insecticides (averaged over the three doses) is given below. Three groups are obtained from the Bonferroni comparisons, with any two insecticides separated by one or more other insecticides in the ordered string having significantly different mean survival times averaged over doses.

If interaction is present, you can conclude that insecticide A is no better than C, but significantly better than B or D, when performance is **averaged over doses**. If the interaction is absent, then A is not significantly better than C, but is significantly better than B or D, **regardless of the dose**. Furthermore, for example, the difference in marginal means for insecticides B and A of 0.677 - 0.314 = 0.363 is the expected decrease in survival time from using A instead of B, regardless of dose. This is also the expected decrease in survival times when averaged over doses.

```
Insect:          B      D      C      A
Marg Mean:   0.677  0.534  0.393  0.314
Groups:      ------------
                    ------------
                           ------------
```

5.2.4 Example: Output voltage for batteries

The maximum output voltage for storage batteries is thought to be influenced by the temperature in the location at which the battery is operated and the material used in the plates. A scientist designed a two-factor study to examine this hypothesis, using three temperatures (50, 65, 80), and three materials for the plates (1, 2, 3). Four batteries were tested at each of the 9 combinations of temperature and material type. The maximum output voltage was recorded for each battery. This is a balanced 3-by-3 factorial experiment with four observations per treatment.

```
#### Example: Output voltage for batteries
dat_battery <-
  read_table2("http://statacumen.com/teach/ADA2/notes/ADA2_notes_Ch05_battery.dat") %>%
  mutate(
    material = factor(material)
  , temp     = factor(temp)
  )
```

```
## Parsed with column specification:
## cols(
##   material = col_double(),
##   temp = col_double(),
##   v1 = col_double(),
##   v2 = col_double(),
##   v3 = col_double(),
##   v4 = col_double()
## )
```

	material	temp	v1	v2	v3	v4
1	1	50	130	155	74	180
2	1	65	34	40	80	75
3	1	80	20	70	82	58
4	2	50	150	188	159	126
5	2	65	136	122	106	115
6	2	80	25	70	58	45
7	3	50	138	110	168	160
8	3	65	174	120	150	139
9	3	80	96	104	82	60

```
dat_battery_long <-
  dat_battery %>%
  pivot_longer(
    cols      = starts_with("v")
```

```
  , names_to  = "battery"
  , values_to = "maxvolt"
  ) %>%
  mutate(
    battery = factor(battery)
  )

str(dat_battery_long)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame': 36 obs. of  4 variables:
##  $ material: Factor w/ 3 levels "1","2","3": 1 1 1 1 1 1 1 1 1 1 ...
##  $ temp    : Factor w/ 3 levels "50","65","80": 1 1 1 1 2 2 2 2 3 3 ...
##  $ battery : Factor w/ 4 levels "v1","v2","v3",..: 1 2 3 4 1 2 3 4 1 2 ...
##  $ maxvolt : num  130 155 74 180 34 40 80 75 20 70 ...
```

The overall F-test at the bottom indicates at least one parameter in the model is significant. The two-way ANOVA table indicates that the main effect of temperature and the interaction are significant at the 0.05 level, the main effect of material is not.

```
lm_m_m_t_mt <- lm(maxvolt ~ material * temp, data = dat_battery_long)
library(car)
Anova(lm_m_m_t_mt, type=3)
```

```
## Anova Table (Type III tests)
##
## Response: maxvolt
##               Sum Sq Df  F value    Pr(>F)
## (Intercept)    72630  1 107.5664 6.456e-11 ***
## material         886  2   0.6562 0.5268904
## temp           15965  2  11.8223 0.0002052 ***
## material:temp   9614  4   3.5595 0.0186112 *
## Residuals      18231 27
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(lm_m_m_t_mt)
```

```
##
## Call:
## lm(formula = maxvolt ~ material * temp, data = dat_battery_long)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -60.750 -14.625   1.375  17.938  45.250
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       134.75      12.99  10.371 6.46e-11 ***
## material2          21.00      18.37   1.143 0.263107
```

```
## material3            9.25      18.37   0.503 0.618747
## temp65             -77.50      18.37  -4.218 0.000248 ***
## temp80             -77.25      18.37  -4.204 0.000257 ***
## material2:temp65    41.50      25.98   1.597 0.121886
## material3:temp65    79.25      25.98   3.050 0.005083 **
## material2:temp80   -29.00      25.98  -1.116 0.274242
## material3:temp80    18.75      25.98   0.722 0.476759
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 25.98 on 27 degrees of freedom
## Multiple R-squared:  0.7652,Adjusted R-squared:  0.6956
## F-statistic:    11 on 8 and 27 DF,  p-value: 9.426e-07
```

The cell means plots of the material profiles have different slopes, which is consistent with the presence of a temperature-by-material interaction.

```
# Calculate the cell means for each (temp, material) combination
sum_battery_mean    <-
  dat_battery_long %>%                                summarize(m = mean(maxvolt))
sum_battery_mean
```

```
## # A tibble: 1 x 1
##       m
##   <dbl>
## 1  106.
```

```
sum_battery_mean_m  <-
  dat_battery_long %>% group_by(material)       %>% summarize(m = mean(maxvolt))
sum_battery_mean_m
```

```
## # A tibble: 3 x 2
##   material     m
##   <fct>    <dbl>
## 1 1         83.2
## 2 2        108.
## 3 3        125.
```

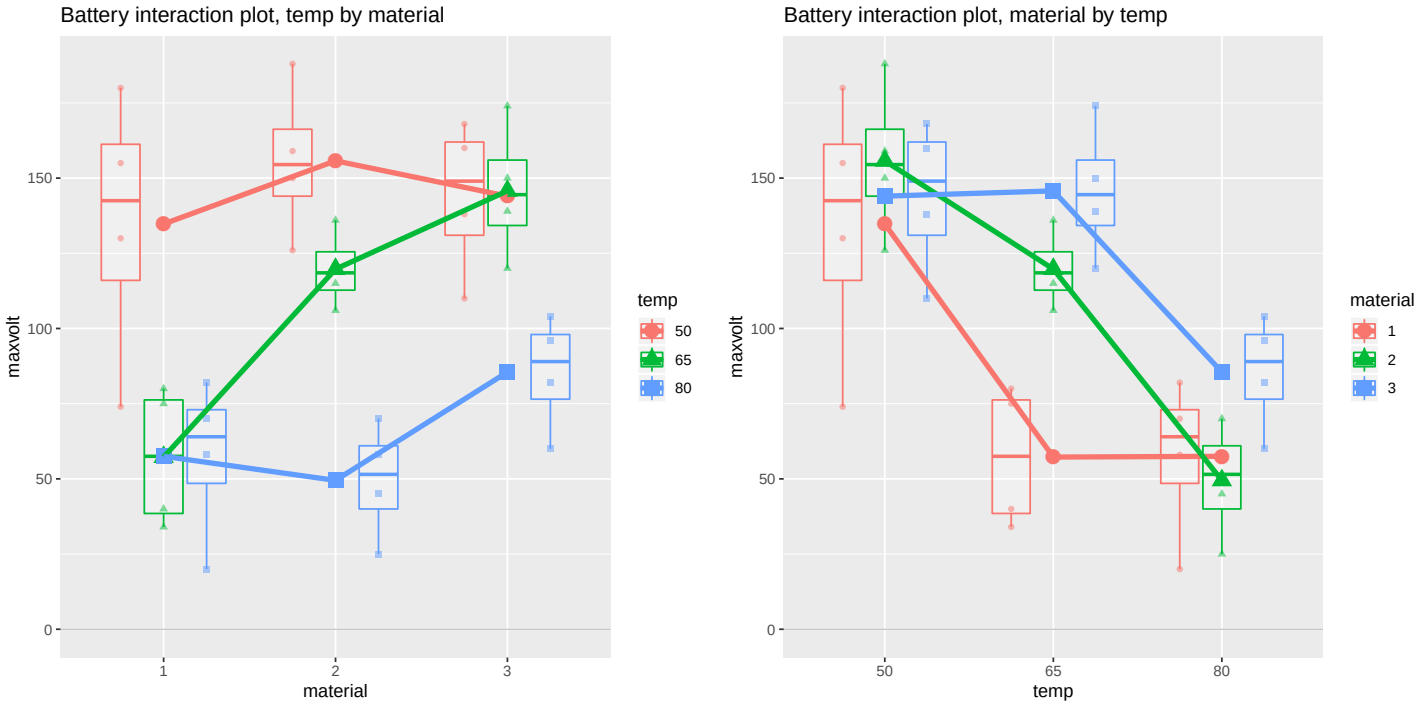
```
sum_battery_mean_t  <-
  dat_battery_long %>% group_by(temp)           %>% summarize(m = mean(maxvolt))
sum_battery_mean_t
```

```
## # A tibble: 3 x 2
##   temp      m
##   <fct> <dbl>
## 1 50    145.
## 2 65    108.
## 3 80     64.2
```

```
sum_battery_mean_mt <-
  dat_battery_long %>% group_by(material, temp) %>% summarize(m = mean(maxvolt))
sum_battery_mean_mt
```

```
## # A tibble: 9 x 3
## # Groups:   material [3]
##   material temp      m
##   <fct>    <fct> <dbl>
## 1 1        50    135.
## 2 1        65     57.2
## 3 1        80     57.5
## 4 2        50    156.
## 5 2        65    120.
## 6 2        80     49.5
## 7 3        50    144
## 8 3        65    146.
## 9 3        80     85.5
```

```
# Interaction plots, ggplot
p <- ggplot(dat_battery_long, aes(x = material, y = maxvolt, colour = temp, shape = temp))
p <- p + geom_hline(aes(yintercept = 0), colour = "black"
                    , linetype = "solid", size = 0.2, alpha = 0.3)
p <- p + geom_boxplot(alpha = 0.25, outlier.size=0.1)
p <- p + geom_point(alpha = 0.5, position=position_dodge(width=0.75))
p <- p + geom_point(data = sum_battery_mean_mt, aes(y = m), size = 4)
p <- p + geom_line(data = sum_battery_mean_mt, aes(y = m, group = temp), size = 1.5)
p <- p + labs(title = "Battery interaction plot, temp by material")
print(p)

p <- ggplot(dat_battery_long, aes(x = temp, y = maxvolt, colour = material, shape = material))
p <- p + geom_hline(aes(yintercept = 0), colour = "black"
                    , linetype = "solid", size = 0.2, alpha = 0.3)
p <- p + geom_boxplot(alpha = 0.25, outlier.size=0.1)
p <- p + geom_point(alpha = 0.5, position=position_dodge(width=0.75))
p <- p + geom_point(data = sum_battery_mean_mt, aes(y = m), size = 4)
p <- p + geom_line(data = sum_battery_mean_mt, aes(y = m, group = material), size = 1.5)
p <- p + labs(title = "Battery interaction plot, material by temp")
print(p)
```



The Bonferroni multiple comparisons may be inappropriate because of covariate interactions. That is, interactions make the main effects less meaningful (or their interpretation unclear) since the change in response when one factor is changed depends on what the second factor is.

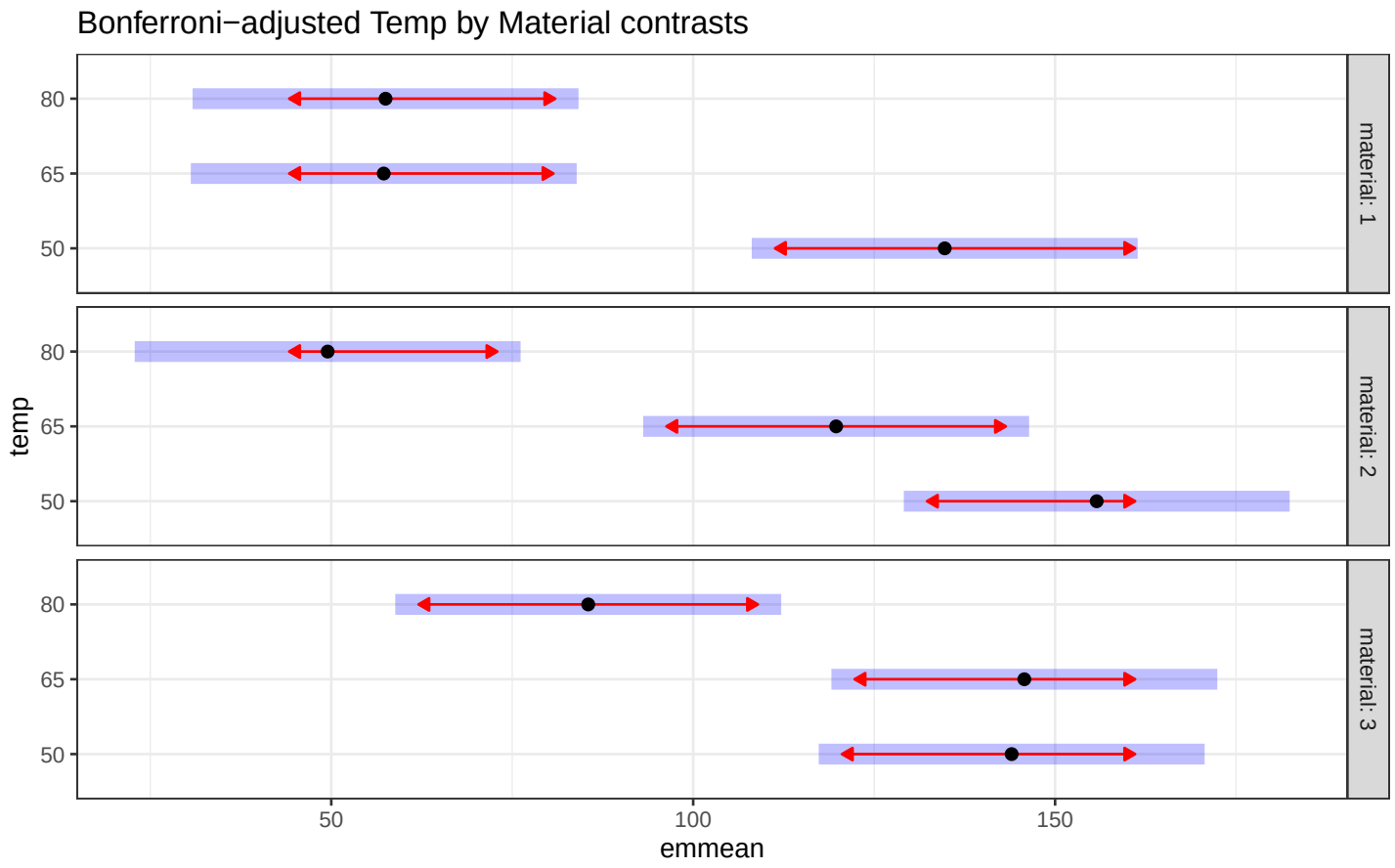
The significant interaction between temperature and material implies that you can not directly conclude that batteries stored at 50 degrees have the highest average output regardless of the material. Nor can you directly conclude that material 3 has a higher average output than material 1 regardless of the temperature. You can only conclude that the differences are significant when averaged over the levels of the other factor.

However, you can compare materials at each temperature, and you can compare temperatures for each material. Below, considering the situation where we want to know the sensitivity to temperature for each material, we condition on material and compare temperatures. Material 1 has a large reduction of maxvolt from temp 50 to 65. Materials 2 and 3 do not reduce their maxvolt untiltemp 80. There's a suggestion that Material 3 is the most robust to higher temperatures, but probably insufficent statistical evidence to claim this.

```
# Contrasts to perform pairwise comparisons
cont_t <-
  emmeans::emmeans(
```

```
    lm_m_m_t_mt
  , specs = "temp"
  , by = "material"
  )

# Means and CIs
confint(cont_t, adjust = "bonferroni")
## material = 1:
##  temp emmean SE df lower.CL upper.CL
##  50    134.8 13 27    101.6    167.9
##  65     57.2 13 27     24.1     90.4
##  80     57.5 13 27     24.3     90.7
##
## material = 2:
##  temp emmean SE df lower.CL upper.CL
##  50    155.8 13 27    122.6    188.9
##  65    119.8 13 27     86.6    152.9
##  80     49.5 13 27     16.3     82.7
##
## material = 3:
##  temp emmean SE df lower.CL upper.CL
##  50    144.0 13 27    110.8    177.2
##  65    145.8 13 27    112.6    178.9
##  80     85.5 13 27     52.3    118.7
##
## Confidence level used: 0.95
## Conf-level adjustment: bonferroni method for 3 estimates
# Pairwise comparisons
cont_t %>% pairs(adjust = "bonf")   # adjust = "tukey" is default
## material = 1:
##  contrast estimate   SE df t.ratio p.value
##  50 - 65     77.50 18.4 27  4.218  0.0007
##  50 - 80     77.25 18.4 27  4.204  0.0008
##  65 - 80     -0.25 18.4 27 -0.014  1.0000
##
## material = 2:
##  contrast estimate   SE df t.ratio p.value
##  50 - 65     36.00 18.4 27  1.959  0.1814
##  50 - 80    106.25 18.4 27  5.783  <.0001
##  65 - 80     70.25 18.4 27  3.823  0.0021
##
## material = 3:
##  contrast estimate   SE df t.ratio p.value
##  50 - 65     -1.75 18.4 27 -0.095  1.0000
##  50 - 80     58.50 18.4 27  3.184  0.0109
##  65 - 80     60.25 18.4 27  3.279  0.0086
##
## P value adjustment: bonferroni method for 3 tests
```

```
# Plot means and contrasts
p <- plot(cont_t, comparisons = TRUE, adjust = "bonf")  # adjust = "tukey" is default
p <- p + labs(title = "Bonferroni-adjusted Temp by Material contrasts")
p <- p + theme_bw()
print(p)
```



The Bonferroni comparisons indicate that the population mean max voltage for the three temperatures **by material types** decreases as the temperature increases, but differently for each material:

```
        Temp:    80       65       50

   Marg mean
material = 1:    57.5     57.2    134.8
       Group:   -------------    -----

material = 2:    49.5    119.8    155.8
       Group:   -----    -------------

material = 3:    85.5    145.8    144.0
       Group:   -----    -------------
```

5.2.5 emmeans and Bonferroni corrections for multi-way interaction models

Note that the p-value adjustment is not performed for all of the tests, but only among the variable levels named in the `specs=` option separately for each level of the `by=` variable. Thus, each set above is corrected for 3 tests, but not all 9 tests together. An easy solution to this has not been found in the `emmeans` package. However, an easy workaround (only for Bonferroni corrections) is to multiply the p-values by the number of levels of the `by=` variable. Thus, because there are 3 material levels, each of the p-values would be multiplied by 3.

Below we convert the contrast table to a tibble (a modern version of the data.frame) in order to multiply the p-values by 3. The resulting p-values have the correct Bonferroni adjustment.

```
# Original table
cont_t %>% pairs(adjust = "bonf") %>%
  as_tibble()
```

```
## # A tibble: 9 x 7
##   contrast material estimate    SE    df t.ratio   p.value
##   <fct>    <fct>       <dbl> <dbl> <dbl>   <dbl>     <dbl>
## 1 50 - 65  1           77.5   18.4    27  4.22   0.000744
## 2 50 - 80  1           77.3   18.4    27  4.20   0.000772
## 3 65 - 80  1           -0.25  18.4    27 -0.0136 1
## 4 50 - 65  2           36.0   18.4    27  1.96   0.181
## 5 50 - 80  2          106.    18.4    27  5.78   0.0000113
## 6 65 - 80  2           70.3   18.4    27  3.82   0.00211
## 7 50 - 65  3           -1.75  18.4    27 -0.0952 1
## 8 50 - 80  3           58.5   18.4    27  3.18   0.0109
## 9 65 - 80  3           60.2   18.4    27  3.28   0.00861
```

```
# Corrected table
cont_t %>% pairs(adjust = "bonf") %>%
  as_tibble() %>%
  mutate(
    p.value = 3 * p.value
    # Restrict p-values to have a maximum value of 1
  , p.value = ifelse(p.value > 1, 1, p.value)
  )
```

```
## # A tibble: 9 x 7
##   contrast material estimate    SE    df t.ratio   p.value
##   <fct>    <fct>       <dbl> <dbl> <dbl>   <dbl>     <dbl>
## 1 50 - 65  1           77.5   18.4    27  4.22   0.00223
## 2 50 - 80  1           77.3   18.4    27  4.20   0.00232
```

```
## 3 65 - 80  1          -0.25  18.4    27 -0.0136 1
## 4 50 - 65  2          36.0   18.4    27  1.96   0.544
## 5 50 - 80  2         106.    18.4    27  5.78   0.0000338
## 6 65 - 80  2          70.3   18.4    27  3.82   0.00634
## 7 50 - 65  3          -1.75  18.4    27 -0.0952 1
## 8 50 - 80  3          58.5   18.4    27  3.18   0.0328
## 9 65 - 80  3          60.2   18.4    27  3.28   0.0258
```

This only works for Bonferroni adjustments because of the simple multiplicative method of correction based on the number of comparisons. The alternative for other methods is to specify the full set of contrasts as in Section 5.2.3. As was demonstrated, this is not too painful and will give you the appropriate multiple comparisons correction.

Note that the plots will not reflect the correction applied in this way, but you will be able to make correct decisions for your hypothesis tests.

5.2.6 Checking assumptions in a two-factor experiment

The normality and constant variance assumptions for a two-factor design can be visually checked using side-by-side boxplots (as was produced in the `ggplot()` interaction plots) and residual plots. Another useful tool for checking constant variances is to plot the sample deviations for each group against the group means.

Let us check the distributional assumptions for the insecticide experiment. The sampling design suggests that the independence assumptions are reasonable. The group sample sizes are small, so the residual plots are likely to be more informative than the side-by-side boxplots and the plot of the standard deviations.

The code below generates plots and summary statistics for the survival times. The means $\bar{y}_{ij}$ and standard deviations s_{ij} for the 12 treatment combinations were calculated. The diagnostic plots we've been using for `lm()` displays residual plots. Only the relevant output is presented.

The set of box plots (each representing 4 points) for each insecticide/dose combination indicates both that means and standard deviations of treatments seem different. Also, there appears to be less variability for dose=3 (high) than for doses 1 and 2 in the table; the model assumes that variability is the same and does not depend on treatment. The plot of the standard deviation vs mean shows an increasing trend.

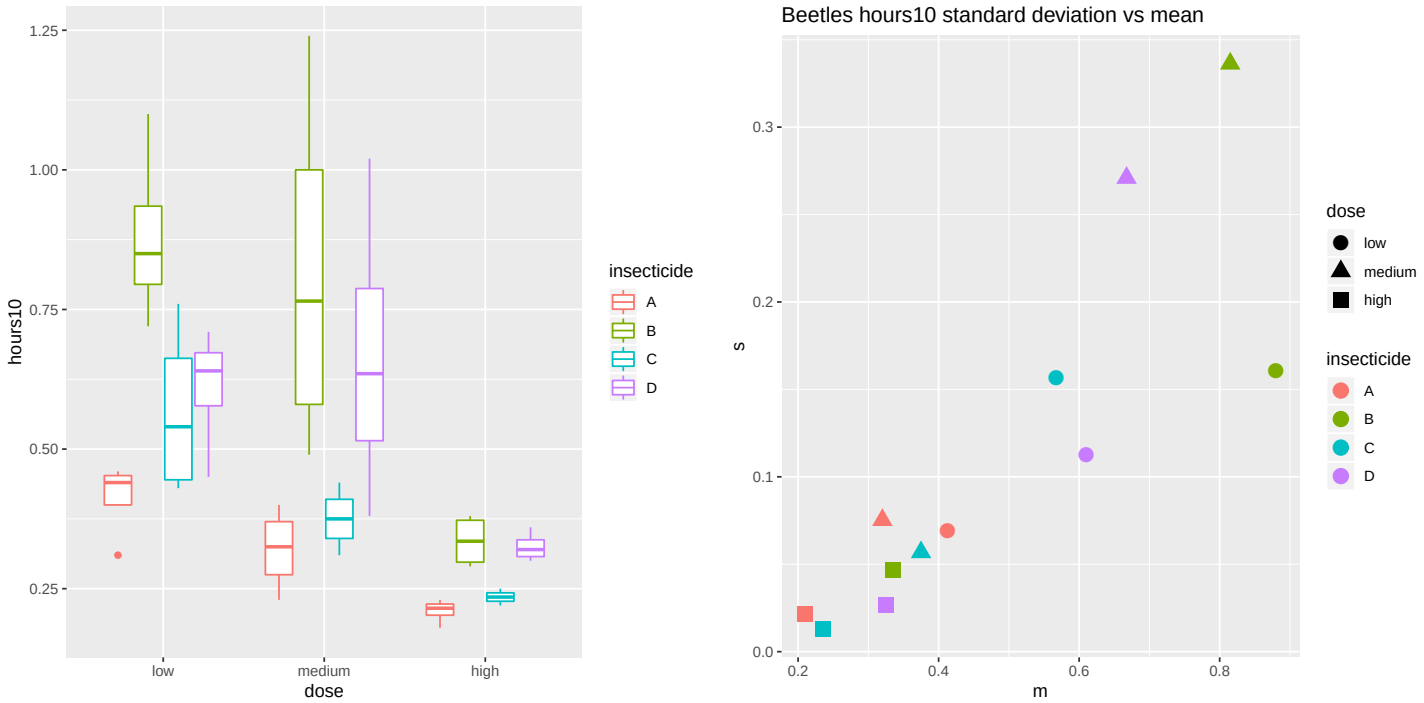
```
#### Example: Beetles, checking assumptions
# boxplots, ggplot
p <- ggplot(dat_beetles_long, aes(x = dose, y = hours10, colour = insecticide))
p <- p + geom_boxplot()
print(p)

# means and standard deviations for each dose/interaction cell
sum_beetles_di <-
  dat_beetles_long %>%
  group_by(dose, insecticide) %>%
  summarize(
    m = mean(hours10)
  , s = sd(hours10)
```

```
  )

sum_beetles_di
```

```
## # A tibble: 12 x 4
## # Groups:   dose [3]
##    dose   insecticide     m      s
##    <ord>  <fct>       <dbl>  <dbl>
##  1 low    A           0.412 0.0695
##  2 low    B           0.88  0.161
##  3 low    C           0.568 0.157
##  4 low    D           0.61  0.113
##  5 medium A           0.32  0.0753
##  6 medium B           0.815 0.336
##  7 medium C           0.375 0.0569
##  8 medium D           0.668 0.271
##  9 high   A           0.21  0.0216
## 10 high   B           0.335 0.0465
## 11 high   C           0.235 0.0129
## 12 high   D           0.325 0.0265
```

```
# mean vs sd plot
p <- ggplot(sum_beetles_di, aes(x = m, y = s, shape = dose, colour = insecticide))
p <- p + geom_point(size=4)
p <- p + labs(title = "Beetles hours10 standard deviation vs mean")
print(p)
```

```
## Warning:  Using shapes for an ordinal variable is not advised
```

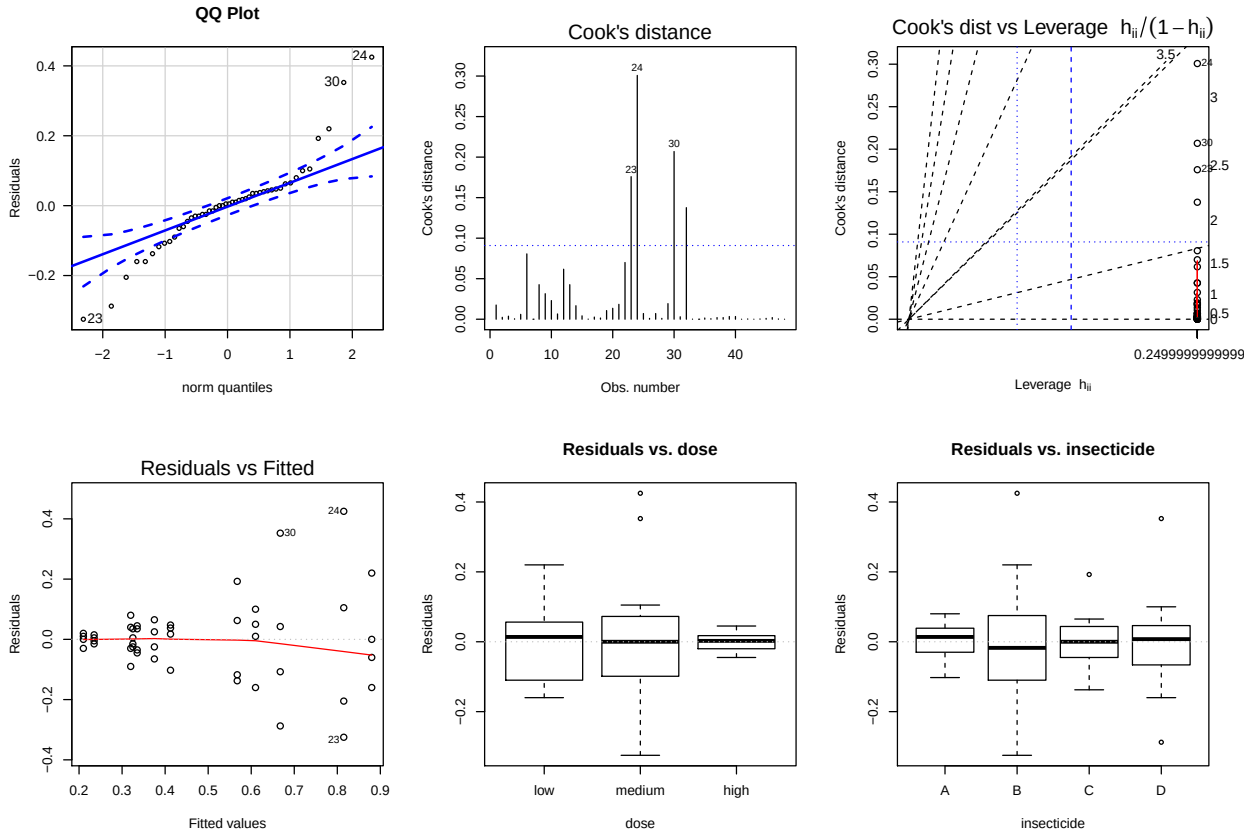


Recall the model assumption diagnostics deviations in Section 5.2.3. Diagnostic plots show the following features. The normal quantile plot

shows an "S" shape rather than a straight line, suggesting the residuals are not normal, but have higher kurtosis (more peaky) than a normal distribution. The residuals vs the fitted (predicted) values show that the higher the predicted value the more variability (horn shaped). The plot of the Cook's distances indicate a few influential observations.

```
# plot diagnistics
lm_diag_plots(lm_h_d_i_di, sw_plot_set = "simple")
```

```
## Warning in if ((class(fit$model[, var_names[i_plot]]) %in% c("numeric", :  the condition has length > 1 and only the first element
will be used
```



Survival times are usually right skewed, with the spread or variability in the distribution increasing as the mean or median increases. Ideally, the distributions should be symmetric, normal, and the standard deviation should be fairly constant across groups.

The boxplots (note the ordering) and the plot of the s_{ij} against $\bar{y}_{ij}$ show the tendency for the spread to increase with the mean. This is reinforced by the residual plot, where the variability increases as the predicted values (the cell means under the two-factor interaction model) increase.

As noted earlier, the QQ-plot of the studentized residuals is better suited to

examine normality here than the boxplots which are constructed from 4 observations. Not surprisingly, the boxplots do not suggest non-normality. Looking at the QQ-plot we clearly see evidence of non-normality.

5.2.7 A Remedy for Non-Constant Variance

A plot of cell standard deviations against the cell means is sometimes used as a diagnostic tool for suggesting transformations of the data. Here are some suggestions for transforming non-negative measurements to make the variability independent of the mean (i.e., stabilize the variance). The transformations also tend to reduce skewness, if present (and may induce skewness if absent!). As an aside, some statisticians prefer to plot the IQR against the median to get a more robust view of the dependence of spread on typical level because s_{ij} and $\bar{y}_{ij}$ are sensitive to outliers.

1. If s_{ij} increases linearly with $\bar{y}_{ij}$, use a **log** transformation of the response.
2. If s_{ij} increases as a quadratic function of $\bar{y}_{ij}$, use a reciprocal (**inverse**) transformation of the response.
3. If s_{ij} increases as a square root function of $\bar{y}_{ij}$, use a **square root** transformation of the response.
4. If s_{ij} is roughly independent of $\bar{y}_{ij}$, do not transform the response. This idea does not require the response to be non-negative!

A logarithmic transformation or a reciprocal (inverse) transformation of the survival times might help to stabilize the variance. The survival time distributions are fairly symmetric, so these nonlinear transformations may destroy the symmetry. As a first pass, I will consider the reciprocal transformation because the inverse survival time has a natural interpretation as the dying rate. For example, if you survive 2 hours, then 1/2 is the proportion of your remaining lifetime expired in the next hour. The unit of time is actually 10 hours, so 0.1/time is the actual rate. The 0.1 scaling factor has no effect on the analysis provided you appropriately rescale the results on the mean responses.

Create the **rate** variable.

```
#### Example: Beetles, non-constant variance
# create the rate variable (1/hours10)
dat_beetles_long <-
  dat_beetles_long %>%
  mutate(
    rate = 1 / hours10
  )
```

Redo the analysis replacing `hours10` by `rate`.

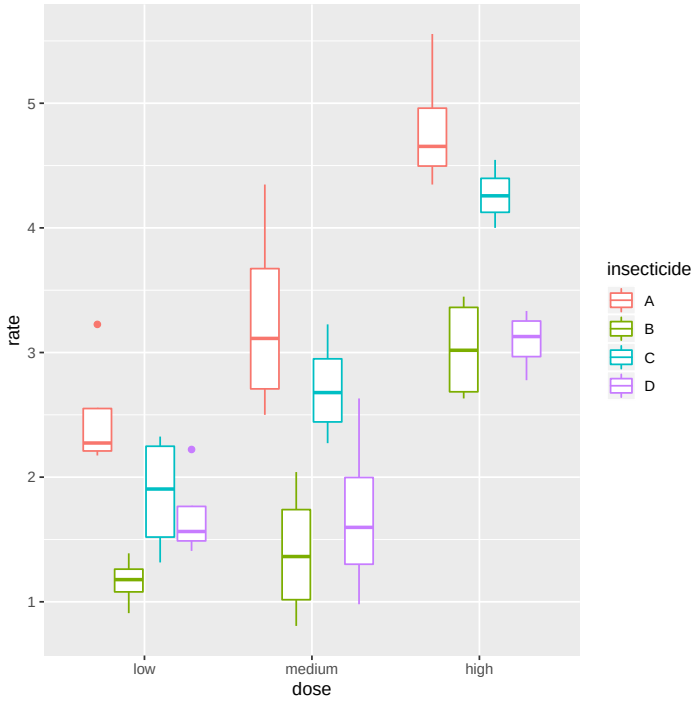
The standard deviations of rate appear much more similar than those of time did.

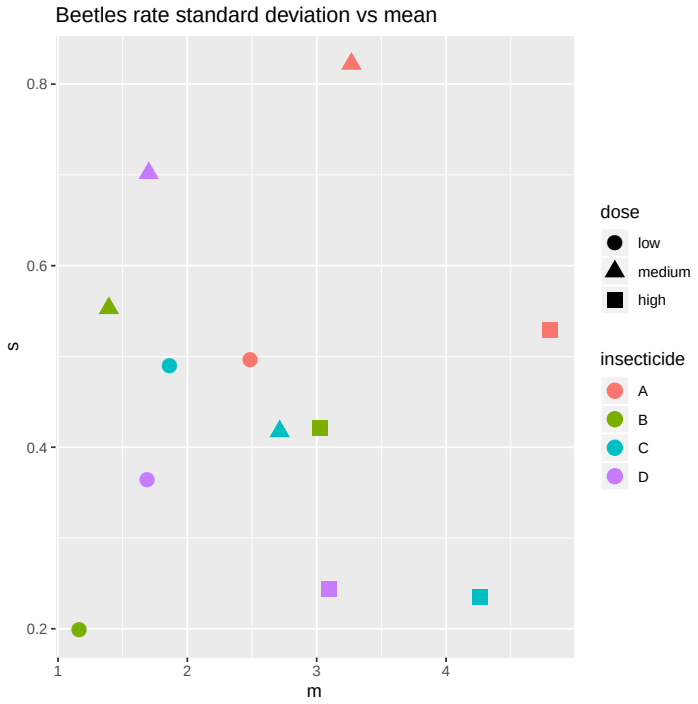
```
# boxplots, ggplot
p <- ggplot(dat_beetles_long, aes(x = dose, y = rate, colour = insecticide))
p <- p + geom_boxplot()
print(p)

# means and standard deviations for each dose/interaction cell
sum_beetles_di_rate <-
  dat_beetles_long %>%
  group_by(dose, insecticide) %>%
  summarize(
    m = mean(rate)
  , s = sd(rate)
  )

sum_beetles_di_rate
```

```
## # A tibble: 12 x 4
## # Groups:   dose [3]
##    dose   insecticide     m     s
##    <ord>  <fct>       <dbl> <dbl>
##  1 low    A            2.49 0.497
##  2 low    B            1.16 0.199
##  3 low    C            1.86 0.489
##  4 low    D            1.69 0.365
##  5 medium A            3.27 0.822
##  6 medium B            1.39 0.553
##  7 medium C            2.71 0.418
##  8 medium D            1.70 0.702
##  9 high   A            4.80 0.530
## 10 high   B            3.03 0.421
## 11 high   C            4.26 0.235
## 12 high   D            3.09 0.244
```

```
# mean vs sd plot
p <- ggplot(sum_beetles_di_rate, aes(x = m, y = s, shape = dose
          , colour = insecticide))
p <- p + geom_point(size=4)
p <- p + labs(title = "Beetles rate standard deviation vs mean")
print(p)
```

```
## Warning:  Using shapes for an ordinal variable is not advised
```





The profile plots and ANOVA table indicate that the main effects are significant but the interaction is not.

```
# Calculate the cell means for each (dose, insecticide) combination
sum_beetles_mean    <-
  dat_beetles_long %>%                                summarize(m = mean(rate))
sum_beetles_mean
```

```
## # A tibble: 1 x 1
##       m
##   <dbl>
## 1  2.62
```

```
sum_beetles_mean_i  <-
  dat_beetles_long %>% group_by(insecticide)       %>% summarize(m = mean(rate))
sum_beetles_mean_i
```

```
## # A tibble: 4 x 2
##   insecticide     m
##   <fct>       <dbl>
## 1 A            3.52
## 2 B            1.86
## 3 C            2.95
## 4 D            2.16
```

```
sum_beetles_mean_d  <-
  dat_beetles_long %>% group_by(dose)              %>% summarize(m = mean(rate))
sum_beetles_mean_d
```

```
## # A tibble: 3 x 2
##   dose       m
##   <ord>  <dbl>
## 1 low     1.80
## 2 medium  2.27
## 3 high    3.80
```

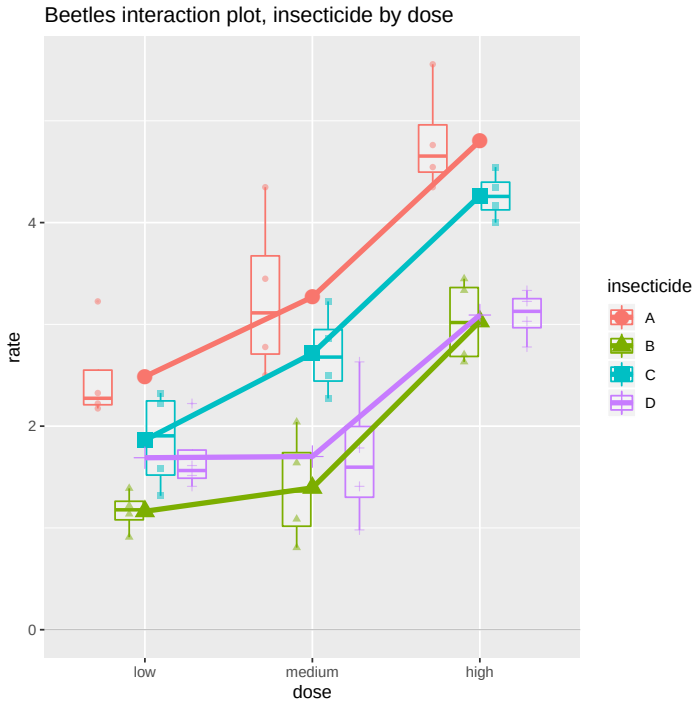
```
sum_beetles_mean_di <-
  dat_beetles_long %>% group_by(dose, insecticide) %>% summarize(m = mean(rate))
sum_beetles_mean_di
```

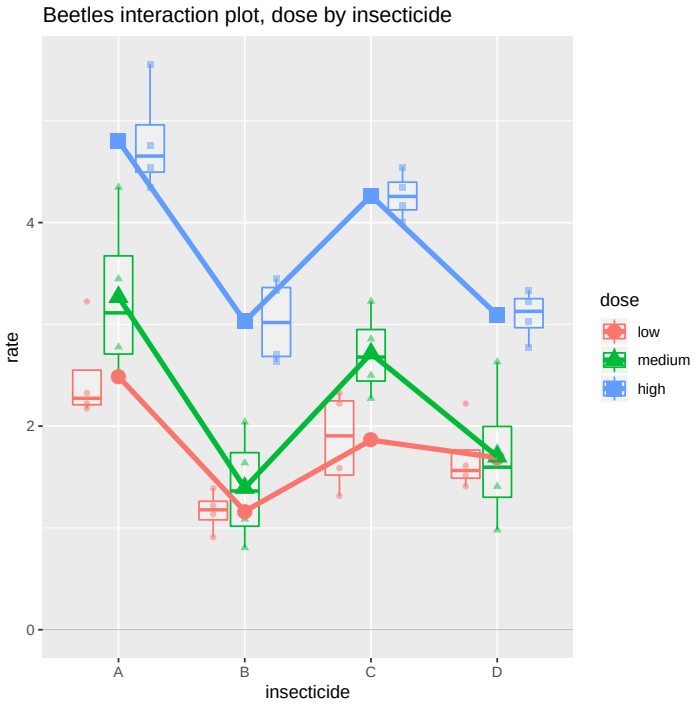
```
## # A tibble: 12 x 3
## # Groups:   dose [3]
##    dose   insecticide     m
##    <ord>  <fct>       <dbl>
##  1 low    A            2.49
##  2 low    B            1.16
##  3 low    C            1.86
##  4 low    D            1.69
```

```
##  5 medium A            3.27
##  6 medium B            1.39
##  7 medium C            2.71
##  8 medium D            1.70
##  9 high   A            4.80
## 10 high   B            3.03
## 11 high   C            4.26
## 12 high   D            3.09
```

```
# Interaction plots, ggplot
p <- ggplot(dat_beetles_long, aes(x = dose, y = rate, colour = insecticide, shape = insecticide))
p <- p + geom_hline(aes(yintercept = 0), colour = "black"
                    , linetype = "solid", size = 0.2, alpha = 0.3)
p <- p + geom_boxplot(alpha = 0.25, outlier.size=0.1)
p <- p + geom_point(alpha = 0.5, position=position_dodge(width=0.75))
p <- p + geom_point(data = sum_beetles_mean_di, aes(y = m), size = 4)
p <- p + geom_line(data = sum_beetles_mean_di, aes(y = m, group = insecticide), size = 1.5)
p <- p + scale_colour_hue()  # ordered factor gives bad colors, reset color for dose
p <- p + labs(title = "Beetles interaction plot, insecticide by dose")
print(p)

p <- ggplot(dat_beetles_long, aes(x = insecticide, y = rate, colour = dose, shape = dose))
p <- p + geom_hline(aes(yintercept = 0), colour = "black"
                    , linetype = "solid", size = 0.2, alpha = 0.3)
p <- p + geom_boxplot(alpha = 0.25, outlier.size=0.1)
p <- p + geom_point(alpha = 0.5, position=position_dodge(width=0.75))
p <- p + geom_point(data = sum_beetles_mean_di, aes(y = m), size = 4)
p <- p + geom_line(data = sum_beetles_mean_di, aes(y = m, group = dose), size = 1.5)
p <- p + scale_colour_hue()  # ordered factor gives bad colors, reset color for dose
p <- p + labs(title = "Beetles interaction plot, dose by insecticide")
print(p)
```

```
## Warning:  Using shapes for an ordinal variable is not advised
```






```
lm_r_d_i_di <- lm(rate ~ dose*insecticide, data = dat_beetles_long) # equivalent
library(car)
Anova(lm_r_d_i_di, type=3)
```

```
## Anova Table (Type III tests)
##
## Response: rate
##                  Sum Sq Df  F value    Pr(>F)
```

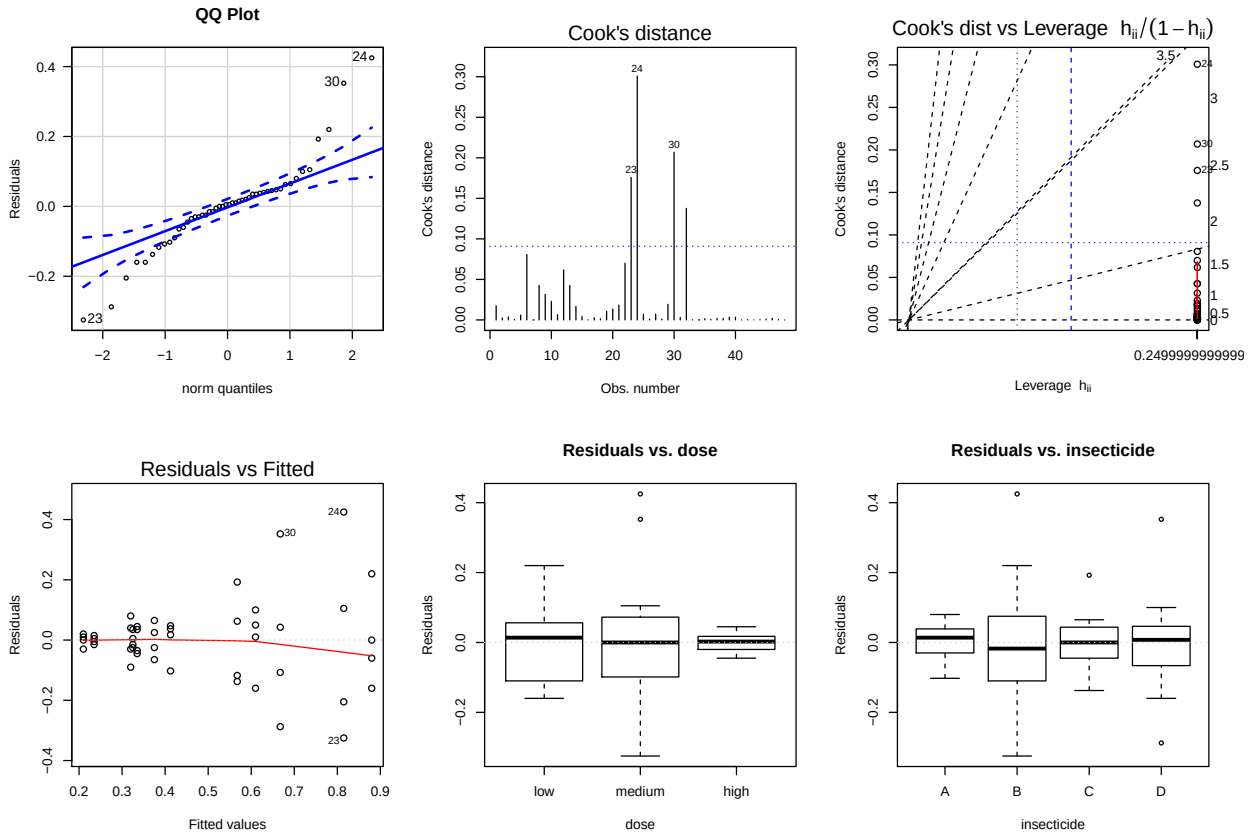
```
## (Intercept)      148.629  1 619.0687 < 2.2e-16 ***
## dose              11.104  2  23.1241 3.477e-07 ***
## insecticide       20.414  3  28.3431 1.376e-09 ***
## dose:insecticide   1.571  6   1.0904    0.3867
## Residuals          8.643 36
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(lm_r_d_i_di)
```

```
##
## Call:
## lm(formula = rate ~ dose * insecticide, data = dat_beetles_long)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.76847 -0.29642 -0.06914  0.25458  1.07936
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          3.51935    0.14145  24.881  < 2e-16 ***
## dose.L               1.63752    0.24499   6.684 8.56e-08 ***
## dose.Q               0.30726    0.24499   1.254  0.21787
## insecticideB        -1.65740    0.20004  -8.286 7.32e-10 ***
## insecticideC        -0.57214    0.20004  -2.860  0.00701 **
## insecticideD        -1.35834    0.20004  -6.790 6.19e-08 ***
## dose.L:insecticideB -0.31841    0.34647  -0.919  0.36421
## dose.Q:insecticideB  0.26660    0.34647   0.769  0.44664
## dose.L:insecticideC  0.06114    0.34647   0.176  0.86093
## dose.Q:insecticideC -0.02154    0.34647  -0.062  0.95078
## dose.L:insecticideD -0.64607    0.34647  -1.865  0.07039 .
## dose.Q:insecticideD  0.25548    0.34647   0.737  0.46568
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.49 on 36 degrees of freedom
## Multiple R-squared:  0.8681,Adjusted R-squared:  0.8277
## F-statistic: 21.53 on 11 and 36 DF,  p-value: 1.289e-12
```

```
# plot diagnistics
lm_diag_plots(lm_h_d_i_di, sw_plot_set = "simple")
```

```
## Warning in if ((class(fit$model[, var_names[i_plot]]) %in% c("numeric", :  the condition has length > 1 and only the first element
will be used
```



Drop the nonsignificant interaction term.

```
lm_r_d_i <- update(lm_r_d_i_di, ~ . - dose:insecticide)
library(car)
Anova(lm_r_d_i, type=3)
```

```
## Anova Table (Type III tests)
##
## Response: rate
##              Sum Sq Df F value    Pr(>F)
## (Intercept) 148.629  1 611.174 < 2.2e-16 ***
## dose         34.877  2  71.708 2.865e-14 ***
## insecticide  20.414  3  27.982 4.192e-10 ***
## Residuals    10.214 42
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(lm_r_d_i)
```

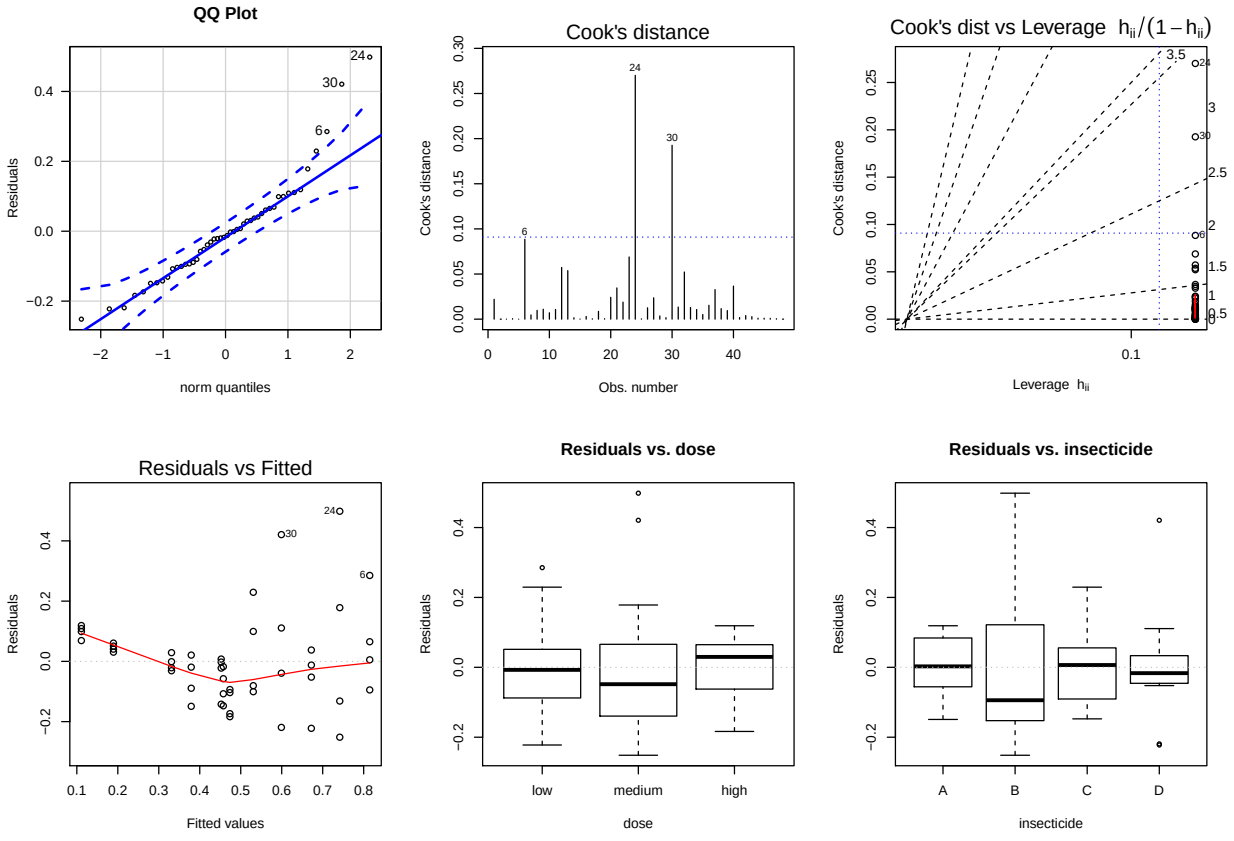
```
##
## Call:
## lm(formula = rate ~ dose + insecticide, data = dat_beetles_long)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.82757 -0.37619  0.02116  0.27568  1.18153
##
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.5193     0.1424  24.722  < 2e-16 ***
## dose.L         1.4117     0.1233  11.451 1.69e-14 ***
## dose.Q         0.4324     0.1233   3.507  0.00109 **
## insecticideB  -1.6574     0.2013  -8.233 2.66e-10 ***
## insecticideC  -0.5721     0.2013  -2.842  0.00689 **
## insecticideD  -1.3583     0.2013  -6.747 3.35e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4931 on 42 degrees of freedom
## Multiple R-squared:  0.8441,Adjusted R-squared:  0.8255
## F-statistic: 45.47 on 5 and 42 DF,  p-value: 6.974e-16
```

Unlike the original analysis, the residual plots do not show any gross deviations from assumptions. Also, no case seems relatively influential.

```
# plot diagnostics
lm_diag_plots(lm_h_d_i, sw_plot_set = "simple")
```

```
## Warning in if ((class(fit$model[, var_names[i_plot]]) %in% c("numeric", :  the condition has length > 1 and only the first element
will be used
```



Bonferroni multiple comparisons imply differences about the mean rates.

```
# Contrasts to perform pairwise comparisons
cont_d <-
  emmeans::emmeans(
```

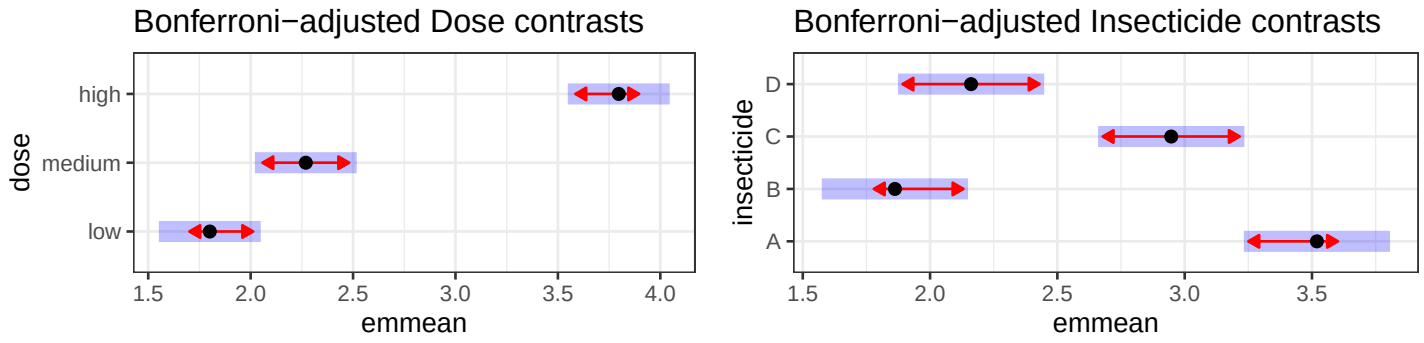
```
    lm_r_d_i
  , specs = "dose"
  )
cont_i <-
  emmeans::emmeans(
    lm_r_d_i
  , specs = "insecticide"
  )

# Means and CIs
#confint(cont_h, adjust = "bonferroni")

# Pairwise comparisons
cont_d %>% pairs(adjust = "bonf")   # adjust = "tukey" is default
```

```
##  contrast      estimate    SE df t.ratio p.value
##  low - medium    -0.469 0.174 42  -2.688 0.0308
##  low - high      -1.996 0.174 42 -11.451 <.0001
##  medium - high   -1.528 0.174 42  -8.763 <.0001
##
## Results are averaged over the levels of: insecticide
## P value adjustment: bonferroni method for 3 tests
```

```
cont_i %>% pairs(adjust = "bonf")   # adjust = "tukey" is default
```

```
##  contrast estimate    SE df t.ratio p.value
##  A - B       1.657 0.201 42  8.233  <.0001
##  A - C       0.572 0.201 42  2.842  0.0414
##  A - D       1.358 0.201 42  6.747  <.0001
##  B - C      -1.085 0.201 42 -5.391  <.0001
##  B - D      -0.299 0.201 42 -1.485  0.8693
##  C - D       0.786 0.201 42  3.905  0.0020
##
## Results are averaged over the levels of: dose
## P value adjustment: bonferroni method for 6 tests
```

```
# Plot means and contrasts
p1 <- plot(cont_d, comparisons = TRUE)  #, adjust = "bonf")  # adjust = "tukey" is default
p1 <- p1 + labs(title = "Bonferroni-adjusted Dose contrasts")
p1 <- p1 + theme_bw()
print(p1)
p2 <- plot(cont_i, comparisons = TRUE)  #, adjust = "bonf")  # adjust = "tukey" is default
p2 <- p2 + labs(title = "Bonferroni-adjusted Insecticide contrasts")
p2 <- p2 + theme_bw()
print(p2)
```



Comments on the Two Analyses of Survival Times

The effects of the transformation are noticeable. For example, the comparisons among doses and insecticides are less sensitive (differences harder to distinguish) on the original scale (look at the Bonferroni groupings). A comparison of the interaction p-values and profile plots for the two analyses suggests that the transformation eliminates much of the observed interaction between the main effects. Although the interaction in the original analysis was not significant at the 10% level (p-value=0.112), the small sample sizes suggest that power for detecting interaction might be low. To be on the safe side, one might interpret the main effects in the original analysis as if an interaction were present. This need appears to be less pressing with the rates.

The statistical assumptions are reasonable for an analysis of the rates. I think that the simplicity of the main effects interpretation is a strong motivating factor for preferring the analysis of the transformed data to the original analysis. You might disagree, especially if you believe that the original time scale is most relevant for analysis.

Given the suitability of the inverse transformation, I did not consider the logarithmic transformation.

5.3 Multiple comparisons: balanced (means) vs unbalanced (emmeans)

The **emmeans** provides a way to compare cell means (combinations of factors), something that is not possible directly with other strategies, such as `glht()`, which compares marginal means.

Using the battery example, we compare the multiple comparison methods using means (`glht()`) and emmeans[2] (`emmeans()`). When there are only main effects, the two methods agree. When there are model interactions, the comparisons of the main effects are inappropriate, and the results differ depending on the method of comparison. When there are model interactions and you want to compare cell means, levels of one factor at each level of another factor separately, then you must use `emmeans()`.

Finally, an important point demonstrated in the next section is that the cell and marginal averages given by the **means** and **emmeans** methods agree here for the main effects model because the design is balanced. For unbalanced designs with two or more factors, **emmeans** and **means** compute different averages. I will argue that **emmeans** are the appropriate averages for unbalanced analyses. You should use the **means** statement with caution — it is OK for balanced or unbalanced one-factor designs, and for the balanced two-factor designs (including the RB) that we have discussed.

[2] `emmeans` is a package written by Russell V. Lenth, PhD of UNM 1975, and well-known for his online power calculators.

5.4 Unbalanced Two-Factor Designs and Analysis

Sample sizes are usually unequal, or **unbalanced**, for the different treatment groups in an experiment. Although this has no consequence on the specification of a model, you have to be more careful with the **analysis of unbalanced experiments** that have two or more factors. With unbalanced designs, the Type I and Type III SS differ, as do the main effect averages given by **means** and **emmeans**. Inferences might critically depend on which summaries are used. I will use the following example to emphasize the differences between the types of SS and averages.

5.4.1 Example: Rat insulin

The experiment consists of measuring insulin levels in rats a certain length of time after a fixed dose of insulin was injected into their jugular or portal veins. This is a two-factor study with two vein types (jugular, portal) and three time levels (0, 30, and 60). An unusual feature of this experiment is that the rats used in the six vein and time combinations are distinct. I will fit a two-factor interaction model, which assumes that the responses are independent within and across treatments (other models may be preferred). The design is **unbalanced**, with sample sizes varying from 3 to 12.

An alternative experimental design might randomly assign rats to the two vein groups, and then measure the insulin levels of each rat at the three time points. Depending on the questions of interest, you could compare veins using a one-way MANOVA, or use other multivariate techniques that allow correlated responses within rats.

```
#### Example: Rat insulin
dat_rat <-
  read_table2("http://statacumen.com/teach/ADA2/notes/ADA2_notes_Ch05_ratinsulin.dat") %>%
  mutate(
    vein = factor(vein)
  , time = factor(time)
  )
```

```
## Parsed with column specification:
## cols(
##   vein = col_character(),
##   time = col_double(),
##   insulin = col_double()
## )
str(dat_rat)
```

```
## Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame': 48 obs. of  3 variables:
##  $ vein   : Factor w/ 2 levels "j","p": 1 1 1 1 1 1 1 1 1 1 ...
##  $ time   : Factor w/ 3 levels "0","30","60": 1 1 1 1 1 2 2 2 2 2 ...
##  $ insulin: num  18 36 12 24 43 61 116 63 132 68 ...
```

```
head(dat_rat, 3)
```

```
## # A tibble: 3 x 3
##   vein  time  insulin
##   <fct> <fct>   <dbl>
## 1 j     0          18
## 2 j     0          36
## 3 j     0          12
```

```
tail(dat_rat, 3)
```

```
## # A tibble: 3 x 3
##   vein  time  insulin
##   <fct> <fct>   <dbl>
## 1 p     60        105
## 2 p     60         71
## 3 p     60         83
```

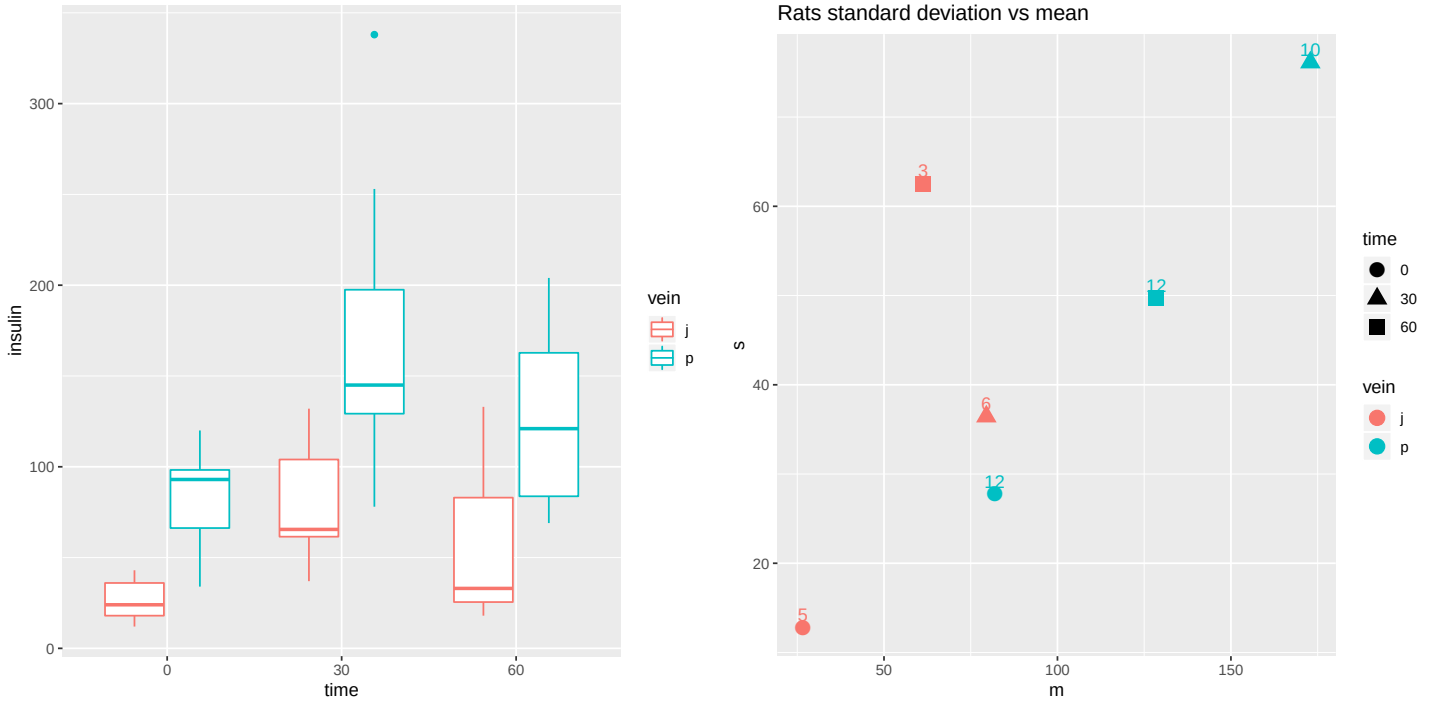
It appears the standard deviation increases with the mean.

```
# boxplots, ggplot
p <- ggplot(dat_rat, aes(x = time, y = insulin, colour = vein))
p <- p + geom_boxplot()
print(p)

# means and standard deviations for each time/interaction cell
sum_rat_tv <-
  dat_rat %>%
  group_by(time, vein) %>%
  summarize(
    m = mean(insulin)
  , s = sd(insulin)
  , n = length(insulin)
  )

sum_rat_tv
```

```
## # A tibble: 6 x 5
## # Groups:   time [3]
##   time  vein      m     s     n
```

```
##   <fct> <fct> <dbl> <dbl> <int>
## 1 0     j      26.6  12.8     5
## 2 0     p      81.9  27.7    12
## 3 30    j      79.5  36.4     6
## 4 30    p     173.   76.1    10
## 5 60    j      61.3  62.5     3
## 6 60    p     128.   49.7    12
```

```
p <- ggplot(sum_rat_tv, aes(x = m, y = s, shape = time, colour = vein, label=n))
p <- p + geom_point(size=4)
# labels are sample sizes
p <- p + geom_text(hjust = 0.5, vjust = -0.5)
p <- p + labs(title = "Rats standard deviation vs mean")
print(p)
```



We take the log of insulin to correct the problem. The variances are more constant now, except for one sample with only 3 observations which has a larger standard deviation than the others, but because this is based on such a small sample size, it's not of much concern.

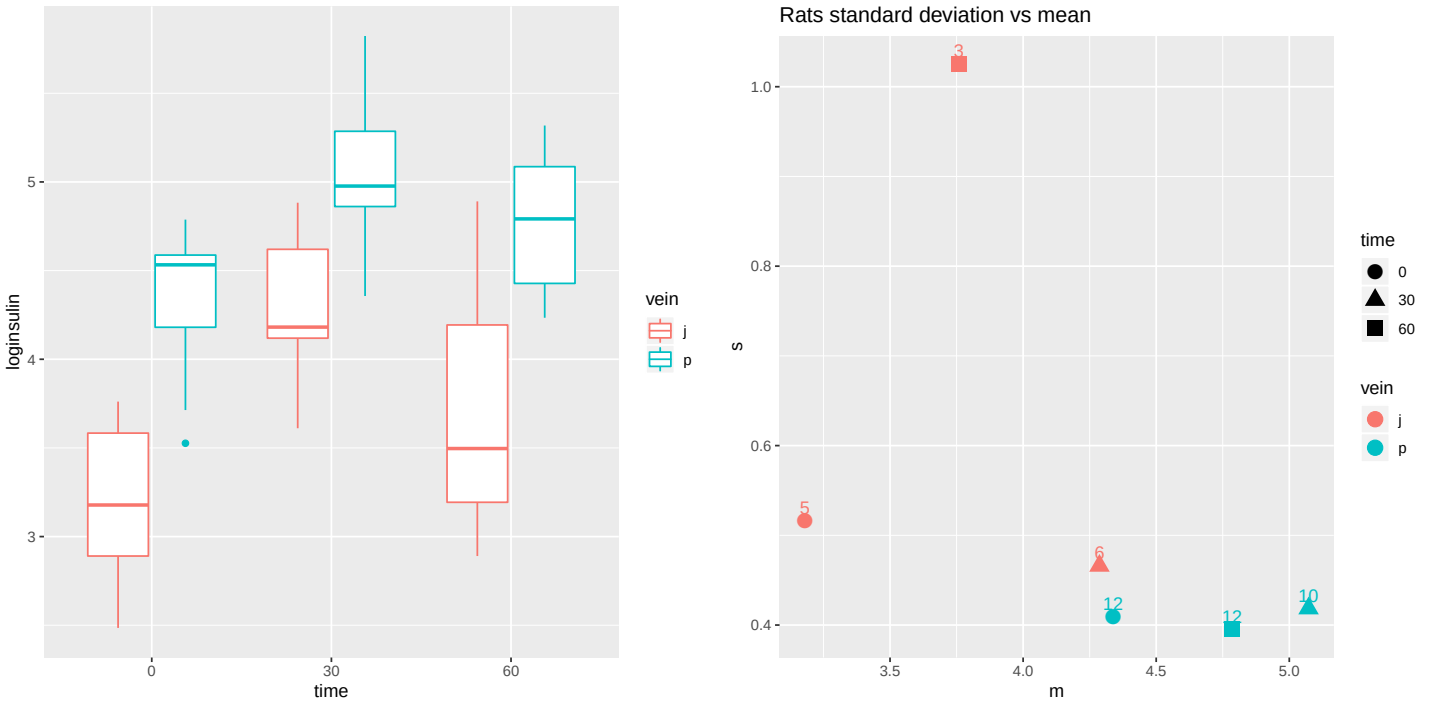
```
dat_rat <-
  dat_rat %>%
  mutate(
    loginsulin = log(insulin)
  )
```

```
# boxplots, ggplot
p <- ggplot(dat_rat, aes(x = time, y = loginsulin, colour = vein))
p <- p + geom_boxplot()
```

```
print(p)

# means and standard deviations for each time/interaction cell
sum_rat_tv <-
  dat_rat %>%
  group_by(time, vein) %>%
  summarize(
    m = mean(loginsulin)
  , s = sd(loginsulin)
  , n = length(loginsulin)
  )

sum_rat_tv
```

```
## # A tibble: 6 x 5
## # Groups:   time [3]
##   time  vein      m     s     n
##   <fct> <fct> <dbl> <dbl> <int>
## 1 0     j      3.18 0.517     5
## 2 0     p      4.34 0.410    12
## 3 30    j      4.29 0.466     6
## 4 30    p      5.07 0.419    10
## 5 60    j      3.76 1.03      3
## 6 60    p      4.79 0.395    12
```

```
# mean vs sd plot
p <- ggplot(sum_rat_tv, aes(x = m, y = s, shape = time, colour = vein, label=n))
p <- p + geom_point(size=4)
# labels are sample sizes
p <- p + geom_text(hjust = 0.5, vjust = -0.5)
p <- p + labs(title = "Rats standard deviation vs mean")
print(p)
```



Type I and Type III SS

We can request ANOVA tables including Type I or Type III SS[3] for each effect in a model. The Type I SS is the **sequential reduction** in Error SS achieved when an effect is added to a model that includes only the prior effects listed in the **model** statement. The Type III SS are more difficult to define explicitly, but they roughly correspond to the reduction in Error SS achieved when an effect is added **last** to the model (or conditional on all other effects being in the model).

Type I SS and Type III SS are equal for balanced designs and for one-way ANOVA, but are typically different for unbalanced designs, where there is no unique way to define the SS for an effect. The problem here is similar to multiple regression, where the SS for a predictor X is the decrease in Residual SS when X is added to a model. This SS is not unique because the change in the Residual SS depends on which predictors are included in the model prior to X. In a regression analysis, the standard tests for effects in a model are based on Type III SS and not on the Type I SS.

[3]For the ugly details, see `http://goanna.cs.rmit.edu.au/~fscholer/anova.php`.

For the insulin analysis, the Type I and Type III interaction SS are identical because this effect was added last to the **model** statement. The Type I and III SS for the main effects are not equal. Also note that the Type I SS for the main effects and interaction add to the model SS, but the Type III SS do not.

Looking at the output, we see the Type I and Type III SS are different, except for the interaction term.

```
lm_i_t_v_tv <-
  lm(
    loginsulin ~ time*vein
  , data = dat_rat
  , contrasts = list(time = contr.sum, vein = contr.sum)
  )
## CRITICAL!!!  Unbalanced design warning.
##   The contrast statement above must be included identifying
##   each main effect with "contr.sum" in order for the correct
##   Type III SS to be computed.
##   See http://goanna.cs.rmit.edu.au/~fscholer/anova.php
library(car)
# Type I   SS  (intercept SS not shown)
summary(aov(lm_i_t_v_tv))
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## time         2  5.450   2.725   12.18 6.74e-05 ***
## vein         1  9.321   9.321   41.66 8.82e-08 ***
## time:vein    2  0.259   0.130    0.58    0.565
## Residuals   42  9.399   0.224
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

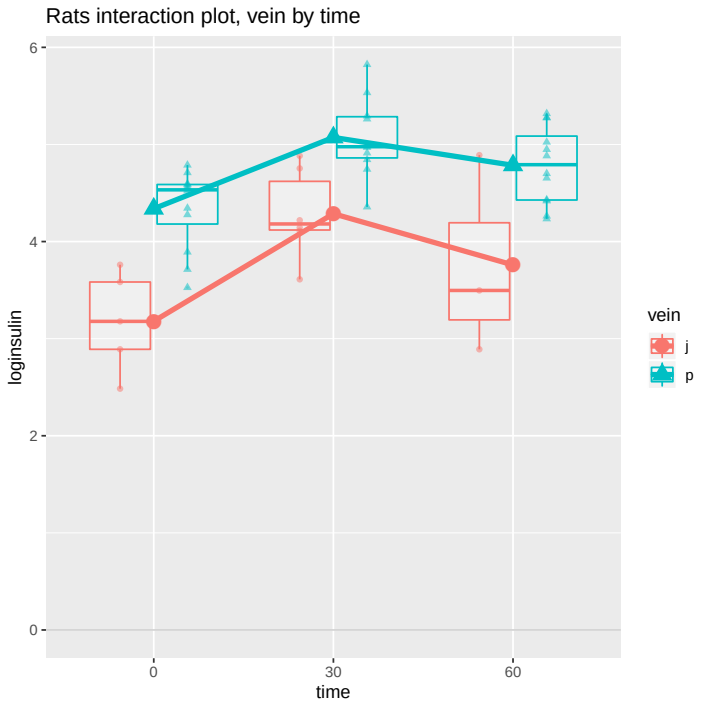
```
# Type III SS
Anova(lm_i_t_v_tv, type=3)
```

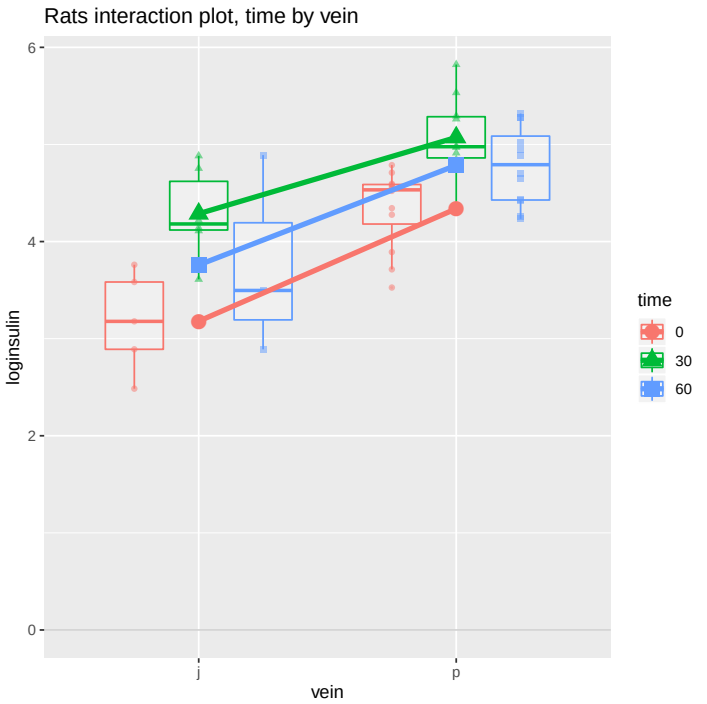
```
## Anova Table (Type III tests)
##
## Response: loginsulin
##             Sum Sq Df   F value    Pr(>F)
## (Intercept) 668.54  1 2987.5842 < 2.2e-16 ***
## time          6.18  2   13.7996 2.475e-05 ***
## vein          9.13  1   40.7955 1.101e-07 ***
## time:vein     0.26  2    0.5797    0.5645
## Residuals     9.40 42
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Because the profile plot lines all seem parallel, and because of the interaction Type III SS p-value above, it appears there is not sufficient evidence for a vein-by-time interaction. For now we'll keep the interaction in the model for the

purpose of discussing differences between means and emmeans and Type I and Type III SS.

```
# Interaction plots, ggplot
p <- ggplot(dat_rat, aes(x = time, y = loginsulin, colour = vein, shape = vein))
p <- p + geom_hline(aes(yintercept = 0), colour = "black"
                    , linetype = "solid", size = 0.2, alpha = 0.3)
p <- p + geom_boxplot(alpha = 0.25, outlier.size=0.1)
p <- p + geom_point(alpha = 0.5, position=position_dodge(width=0.75))
p <- p + geom_point(data = sum_rat_tv, aes(y = m), size = 4)
p <- p + geom_line(data = sum_rat_tv, aes(y = m, group = vein), size = 1.5)
p <- p + labs(title = "Rats interaction plot, vein by time")
print(p)

p <- ggplot(dat_rat, aes(x = vein, y = loginsulin, colour = time, shape = time))
p <- p + geom_hline(aes(yintercept = 0), colour = "black"
                    , linetype = "solid", size = 0.2, alpha = 0.3)
p <- p + geom_boxplot(alpha = 0.25, outlier.size=0.1)
p <- p + geom_point(alpha = 0.5, position=position_dodge(width=0.75))
p <- p + geom_point(data = sum_rat_tv, aes(y = m), size = 4)
p <- p + geom_line(data = sum_rat_tv, aes(y = m, group = time), size = 1.5)
p <- p + labs(title = "Rats interaction plot, time by vein")
print(p)
```





Means versus emmeans

The **emmeans** (**estimated marginal means**, sometimes called **adjusted means**) for a single factor is an arithmetic average of cell means. For example, the mean responses in the jugular vein at times 0, 30, and 60 are 3.18, 4.29, and 3.76, respectively. The **emmeans** for the jugular vein is thus $3.74 = (3.18 + 4.29 + 3.76)/3$. This average gives equal weight to the 3 times even though the sample sizes at these times differ (5, 6, and 3). The **means** of 3.78 for the jugular is the average of the 14 jugular responses, ignoring time. If the cell sample sizes were equal, the **emmeans** and **means** averages would agree.

The **means** and **emmeans** for individual cells (i.e., for the 6 **vein*time** combinations) are identical, and equal to cell means.

```
# unbalanced, group means vs emmeans don't match
sum_rat_t <-
  dat_rat %>%
  group_by(time) %>%
  summarize(
    m = mean(loginsulin)
  )
sum_rat_t
```

```
## # A tibble: 3 x 2
##   time      m
##   <fct> <dbl>
## 1 0      4.00
## 2 30     4.78
## 3 60     4.58
```

```
cont_t <-
  emmeans::emmeans(
    lm_i_t_v_tv
  , specs = "time"
  )
```

```
## NOTE: Results may be misleading due to involvement in interactions
cont_t
```

```
##  time emmean    SE df lower.CL upper.CL
##  0      3.76 0.126 42     3.50     4.01
##  30     4.68 0.122 42     4.43     4.93
##  60     4.27 0.153 42     3.96     4.58
##
## Results are averaged over the levels of: vein
## Confidence level used: 0.95
```

```r
# unbalanced, group means vs emmeans don't match
sum_rat_v <-
  dat_rat %>%
  group_by(vein) %>%
  summarize(
    m = mean(loginsulin)
  )
sum_rat_v
```

```
## # A tibble: 2 x 2
##   vein      m
##   <fct> <dbl>
## 1 j      3.78
## 2 p      4.71
```

```r
cont_v <-
  emmeans::emmeans(
    lm_i_t_v_tv
  , specs = "vein"
  )
## NOTE: Results may be misleading due to involvement in interactions
cont_v
```

```
##  vein emmean     SE df lower.CL upper.CL
##  j      3.74 0.1319 42     3.48     4.01
##  p      4.73 0.0814 42     4.57     4.90
##
## Results are averaged over the levels of: time
## Confidence level used: 0.95
```

```r
  # compare jugular mean above (3.778) with the emmeans average below (3.742)
(3.179610 + 4.286804 + 3.759076)/3
```

```
## [1] 3.74183
```

```r
# unbalanced, but highest-order interaction cell means will match
sum_rat_tv <-
  dat_rat %>%
  group_by(vein, time) %>%
  summarize(
    m = mean(loginsulin)
  )
sum_rat_tv
```

```
## # A tibble: 6 x 3
## # Groups:   vein [2]
##   vein  time      m
##   <fct> <fct> <dbl>
## 1 j     0      3.18
## 2 j     30     4.29
## 3 j     60     3.76
## 4 p     0      4.34
```

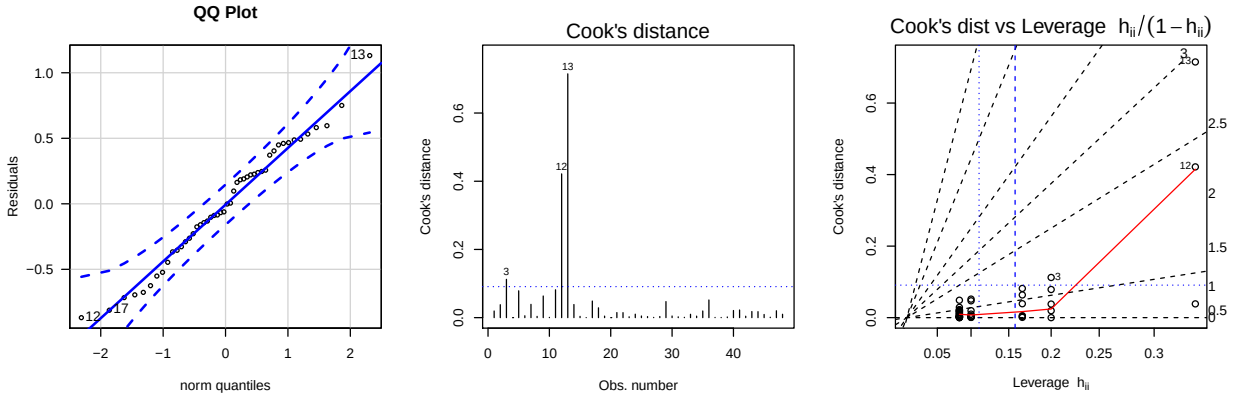
```
## 5 p      30     5.07
## 6 p      60     4.79
```

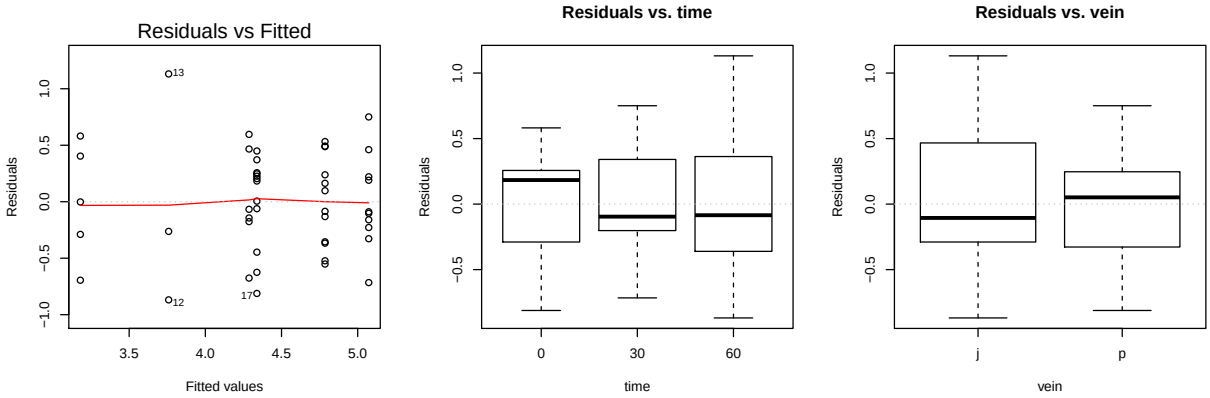
```
cont_tv <-
  emmeans::emmeans(
    lm_i_t_v_tv
  , specs = "time"
  , by = "vein"
  )
cont_tv
```

```
## vein = j:
##  time emmean    SE df lower.CL upper.CL
##  0      3.18 0.212 42     2.75     3.61
##  30     4.29 0.193 42     3.90     4.68
##  60     3.76 0.273 42     3.21     4.31
##
## vein = p:
##  time emmean    SE df lower.CL upper.CL
##  0      4.34 0.137 42     4.06     4.61
##  30     5.07 0.150 42     4.77     5.37
##  60     4.79 0.137 42     4.51     5.06
##
## Confidence level used: 0.95
```

For completeness, these diagnostic plots are mostly fine, though the plot of the Cook's distances indicate a couple influential observations.

```
# interaction model
lm_i_t_v_tv <-
  lm(
    loginsulin ~ time*vein
  , data = dat_rat
  , contrasts = list(time = contr.sum, vein = contr.sum)
  )

# plot diagnostics
lm_diag_plots(lm_i_t_v_tv, sw_plot_set = "simple")
```




Residuals vs Fitted
Residuals
Fitted values
13
12
17
Residuals vs. time
Residuals
time
0
30
60
Residuals vs. vein
Residuals
vein
j
p

Should I use means or emmeans, Type I or Type III SS?

Use ***emmeans*** *and* ***Type III*** *SS.*

Regardless of whether the design is balanced, the basic building blocks for a two-factor analysis are cell means, and the marginal means, defined as the average of the cell means over the levels of the other factor.

The F-statistics based on Type III SSs are appropriate for unbalanced two-factor designs because they test the same hypotheses that were considered in balanced designs. That is, the Type III F-tests on the main effects check for equality in population means averaged over levels of the other factor. The Type III F-test for no interaction checks for parallel profiles. Given that the Type III F-tests for the main effects check for equal population cell means averaged over the levels of the other factor, multiple comparisons for main effects should be based on **emmeans**.

The Type I SS and F-tests and the multiple comparisons based on **means should be ignored** because they do not, in general, test meaningful hypotheses. The problem with using the **means** output is that the experimenter has fixed the sample sizes for a two-factor experiment, so comparisons of **means**, which ignore the second factor, introduces a potential bias due to choice of sample sizes. Put another way, any differences seen in the **means** in the jugular and portal could be solely due to the sample sizes used in the experiment and not due to differences in the veins.

Focusing on the Type III SS, the F-tests indicate that the vein and time effects are significant, but that the interaction is not significant. The jugular and portal profiles are reasonably parallel, which is consistent with a lack of interaction. What can you conclude from the **emmeans** comparisons of veins and times?

Answer: significant differences between veins, and between times 0 and 30.

5.5 Writing factor model equations and interpretting coefficients

This section is an exercise in writing statistical factor models, plotting predicted values, and interpretting model coefficients. You'll need pen (preferrably multi-colored) and paper.

In class I'll discuss indicator variables and writing these models. Together, we'll plot the model predicted values, write the model for each factor combination, and indicate the β coefficients on the plot (βs are vertical differences)[4]. From the exercise, I hope that coefficient interpretation will become clear. Assume a balanced design with n_i observations for each treatment combination.

5.5.1 One-way ANOVA, 1 factor with 3 levels

1. Write ANOVA factor model (general and indicator-variables)
2. Write model for each factor level with βs and predicted values
3. Plot the predicted values on axes
4. Label the βs on the plot
5. Calculate marginal and grand means in table and label in plot

Level	1	2	3	
$\hat{y}$	5	4	6	

5.5.2 Two-way ANOVA, 2 factors with 3 and 2 levels, additive model

1. Write two-way ANOVA factor model (general and indicator-variables)
2. Write model for each factor level with βs and predicted values
3. Plot the predicted values on axes
4. Label the βs on the plot
5. Calculate marginal and grand means in table and label in plot

[4]Please attempt by hand before looking at the solutions at `http://statacumen.com/teach/ADA2/notes/ADA2_notes_05_PairedAndBlockDesigns_CoefScan.pdf`.

$\hat{y}$	Factor 1		
Factor 2	1	2	3
1	5	4	6
2	8	7	9

5.5.3 Two-way ANOVA, 2 factors with 3 and 2 levels, interaction model

1. Write two-way ANOVA factor model (general and indicator-variables)
2. Write model for each factor level with βs and predicted values
3. Plot the predicted values on axes
4. Label the βs on the plot
5. Calculate marginal and grand means in table and label in plot

$\hat{y}$	Factor 1		
Factor 2	1	2	3
1	5	4	6
2	8	10	3