Chapter 4: Worked Example

Jules Lanari-Collard

2024-08-09

This chapter uses data from a cross-over study by Dai, Lov, Martin-Arrowsmith et al. [**?**]. The trial measured various outcomes after ingesting a protein-based beverage, comparing between beef- or insect-derived proteins. 20 subjects were randomised into either the cricket-beef (C-B) sequence or the beef-cricket (B-C) sequence. For the following examples, we focus on data collected on insulin levels 300 minutes after ingestion.

We will apply the methods described in the preceding chapters to the data, in order to determine whether there is a difference between the beef- or insect-derived proteins (in terms of resulting insulin levels).

Standard Cross-over Design

Data Structure

Most data manipulation and visualisation is performed using the `tidyverse` package (which includes `ggplot`); other packages used are indicated where applicable. It is important that the columns for subject, sequence, period and treatment are treated as factors (instead of continuous).

The data were collected in 'long' format, where we have a column for period and a column for the measurement (see table 1). For some plots it is easier for the data to be in 'wide' format (see table 2); this can easily be achieved with `tidyr::pivot_wider()`.

```
data.wide <- data %>%
  pivot_wider(id_cols = c(Subject, Sequence),
              names_from = Period, values_from = Insulin,
              names_prefix = "Period")
```

Summarising the Data

Now we can move on to the summary tables and plots, to gain a better understanding of the data. First, the summary table; initial summaries by sequence and period can be easily calculated using

Table 1: Subsample of Data in 'Long' Format

Subject	Sequence	Period	Treatment	Insulin
1	C-B	1	CRICKET	18.3
1	C-B	2	BEEF	14.1
2	C-B	1	CRICKET	14.7
2	C-B	2	BEEF	24.7
4	C-B	1	CRICKET	45.9

Table 2: Subsample of Data in 'Wide' Format

Subject	Sequence	Period1	Period2
1	C-B	18.3	14.1
2	C-B	14.7	24.7
4	C-B	45.9	35.8
5	C-B	46.1	38.6
7	C-B	17.6	17.7

`dplyr::summarise()`, but the aggregated total summaries (total, by period and by sequence) must be calculated separately and joined to compile the full summary table.

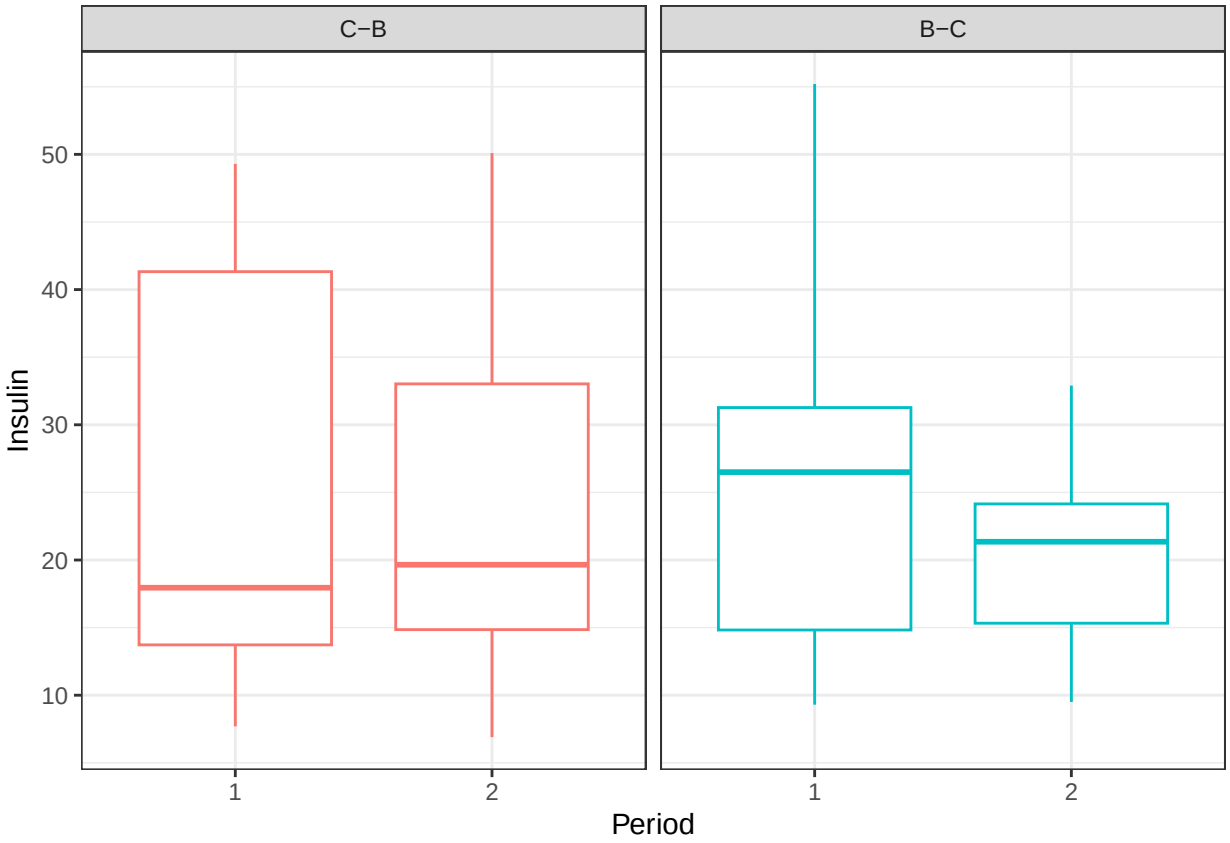
```
# Summary by period & sequence
summary.period.sequence <- data.wide %>%
  group_by(Sequence) %>%
  summarise(Subjects = n(),
            meanPeriod1 = mean(Period1),
            sdPeriod1 = sd(Period1),
            meanPeriod2 = mean(Period2),
            sdPeriod2 = sd(Period2))

# Summary by sequence
summary.sequence <- data %>%
  group_by(Sequence) %>%
  summarise(meanOverall = mean(Insulin),
            sdOverall = sd(Insulin))

# Summary by period
summary.period <- data %>%
  group_by(Period) %>%
  summarise(meanPeriod = mean(Insulin),
            sdPeriod = sd(Insulin)) %>%
  pivot_wider(names_from = Period, values_from = c(meanPeriod, sdPeriod),
              names_sep = "") %>%
  mutate(Sequence = "Total", Subjects = n_distinct(data$Subject),
         meanOverall = mean(data$Insulin),
         sdOverall = sd(data$Insulin),
         .before = meanPeriod1)

# Combine tables together
table1 <- inner_join(summary.sequence, summary.period.sequence,
                     by = "Sequence") %>%
  bind_rows(summary.period) %>%
  relocate(Subjects, .after = Sequence)
```

Visualisations

We start with a boxplot (see figure **??**) and groups-by-periods plot (see figure **??**), summarising the results by sequence and period. The subject-profile plot (see figure **??**) provides a subject-level visualisation.

Table 3: Summary Table

		Overall		Period 1		Period 2	
Sequence	Subjects	Mean	SD	Mean	SD	Mean	SD
C-B	10	24.55	14.43	25.13	16.07	23.97	13.45
B-C	10	23.12	11.11	25.84	13.83	20.40	7.27
Total	20	23.84	12.74	25.48	14.60	22.18	10.68



Now we examine potential treatment differences with figures **??** and **??**. Due to the relatively small sample size, it is difficult to extract any potential trends from the data at this stage. Notice, however, that there are a number of potential outliers in each sequence, which appear to be influencing the centroids (which are calculated using means).

Reconstructing the plot, using median centroids instead (see figure **??**), confirms this suspicion and provides more insight. We see that the median centroids appear on opposite sides of the line, with some vertical separation, indicating a potential difference between the treatments.

Modelling for Treatment Differences

First we construct an ANOVA table, to verify that the other effects do not overly influence the data. This can be easily done using the built-in `aov()` function on the data in 'long' format, specifying the linear model equation as an argument. The ANOVA table itself can be obtained using the `summary()` function on the

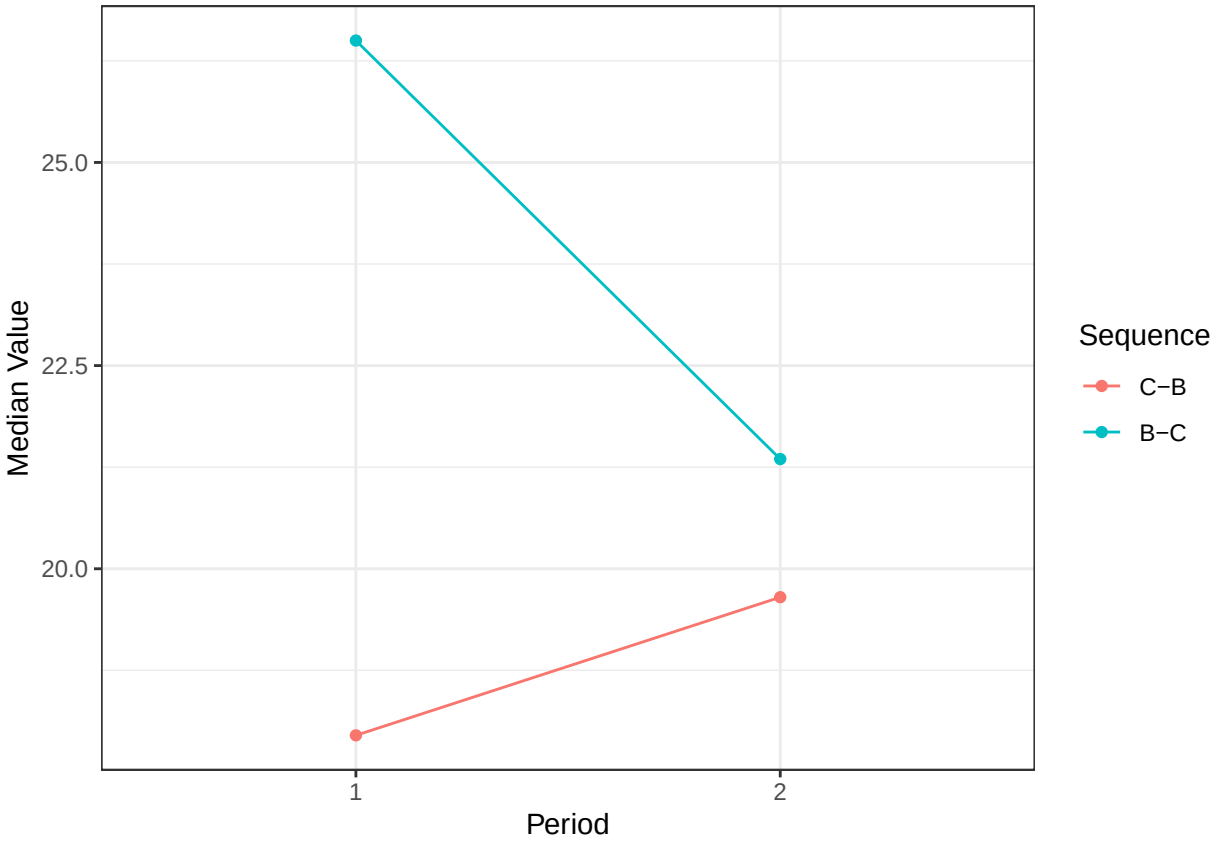


Figure 1: Groups-by-Periods Plot

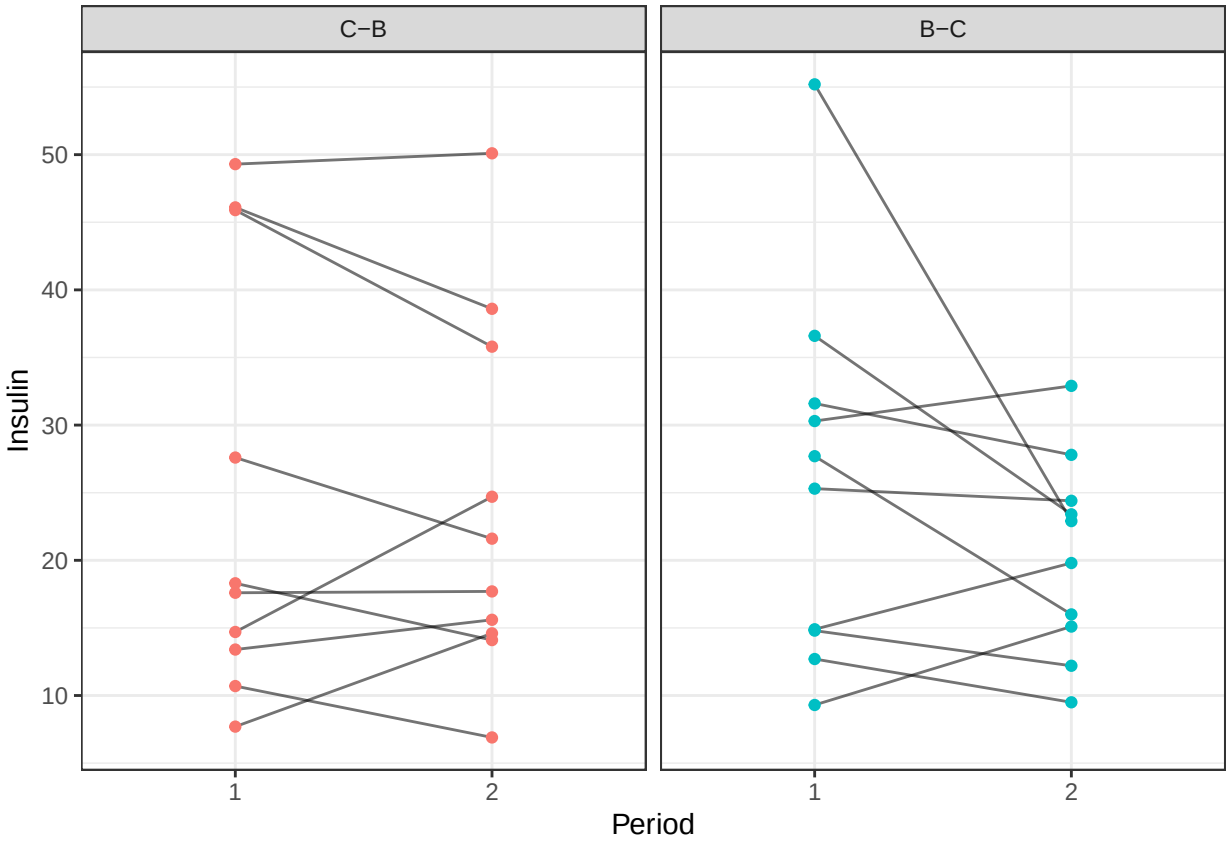


Figure 2: Subject-Profile Plot

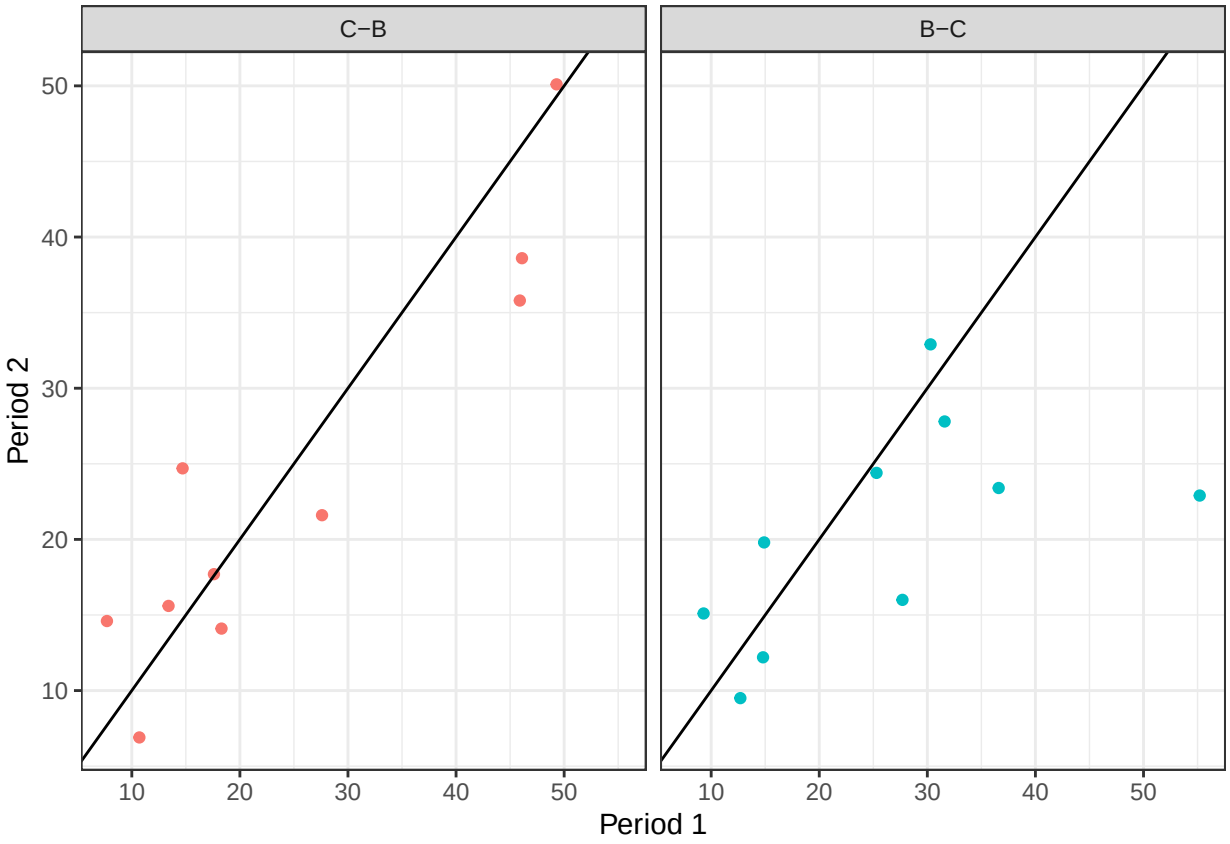


Figure 3: Period-by-Period Plot

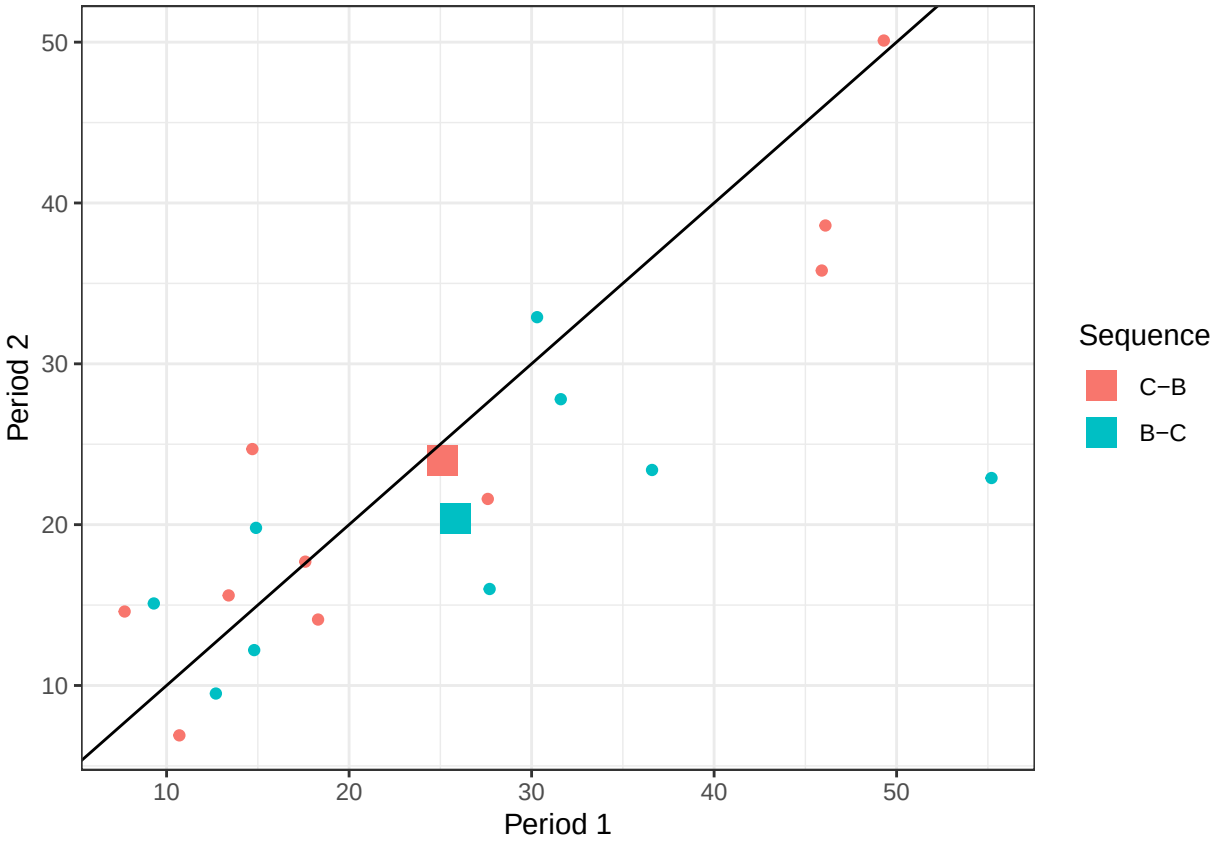


Figure 4: Period-by-Period Plot with Centroids

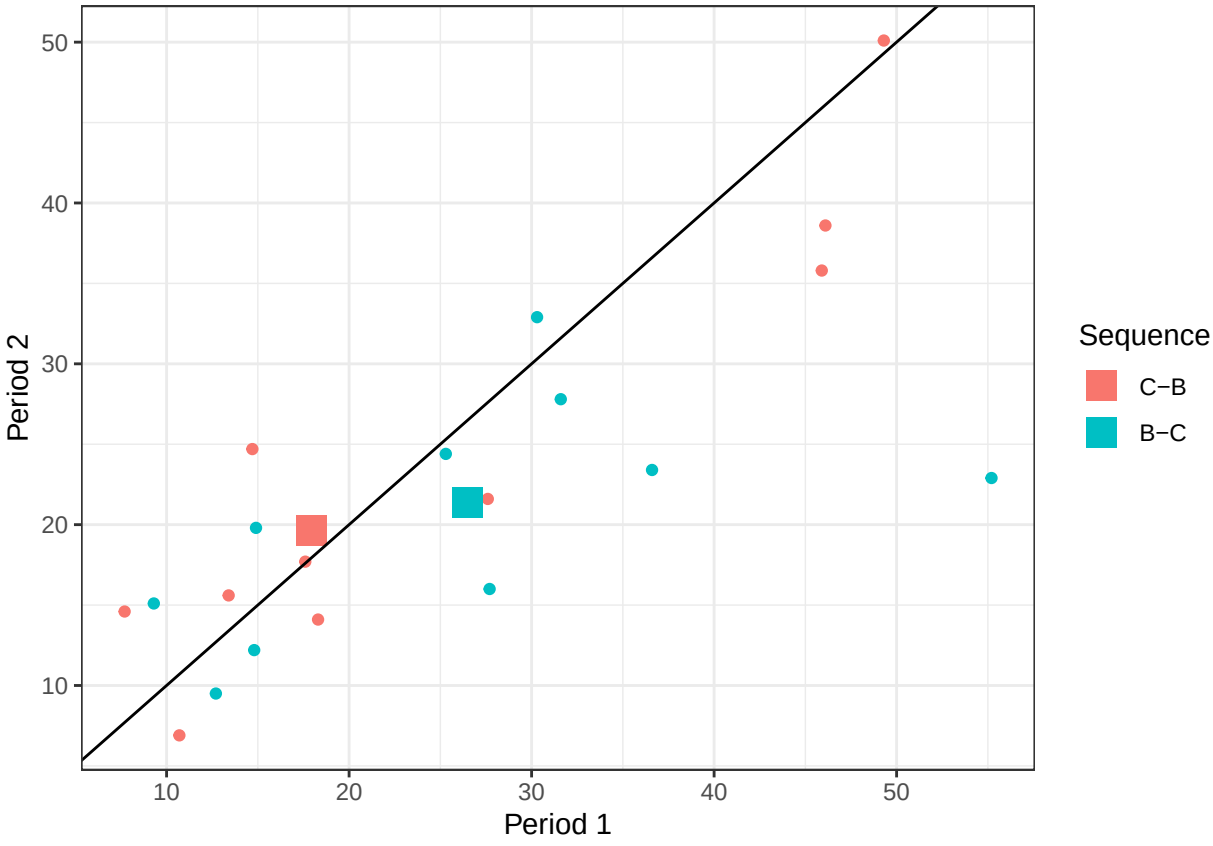


Figure 5: Period-by-Period Plot with Median Centroids

aov object. The resulting table is shown in table **??**.

```
anova.table <- aov(Insulin ~ Treatment + Period + Sequence + Subject, data = data) %>%
  summary()
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Treatment	1	45.80	45.80	1.08	0.3116
Period	1	108.90	108.90	2.58	0.1258
Sequence	1	20.45	20.45	0.48	0.4955
Subject	18	5390.02	299.45	7.09	0.0001
Residuals	18	760.56	42.25		

Table 4: ANOVA on Linear Model

We see that none of the other effects are significant, so we can safely move on to test for treatment differences. The mixed model can be implemented using the `lme4::lmer()` function, in conjunction with the `lmerTest` package (for p-values). We specify the linear model as normal, with the `(1|Subject)` term specifying that Subject is a random effect.

```
library(lme4)
library(lmerTest)
mixed.model <- lmer(Insulin ~ Treatment + Period + Sequence + (1 | Subject),
                    data = data)
```

Table 5: Estimates for Mixed Model

	Estimate	Std. Error	df	t value	Pr(>|t|)
(Intercept)	25.13	4.13	22.98	6.08	0.00
TreatmentBEEF	2.14	2.06	18.00	1.04	0.31
Period2	-3.30	2.06	18.00	-1.61	0.13
SequenceB-C	-1.43	5.47	18.00	-0.26	0.80

As shown in the model results table **??**, the treatment term is not significant, so at this stage we cannot conclude that there is a difference between the treatments.

Verifying Assumptions

At this stage it is important to verify the assumptions of the model. The `broom.mixed::augment()` function adds columns to the original dataframe with the fitted values, residuals and more. This information can be used to verify the model assumptions, by constructing a plot of residuals against fitted values (figure **??**), and a Q-Q plot (figure **??**).

```
library(broom.mixed)
mixed.model.metrics <- mixed.model %>% augment()
```

Excluding a few outliers, figures **??** and **??** show nothing to suggest that the assumptions of homoscedasticity and normality were violated.

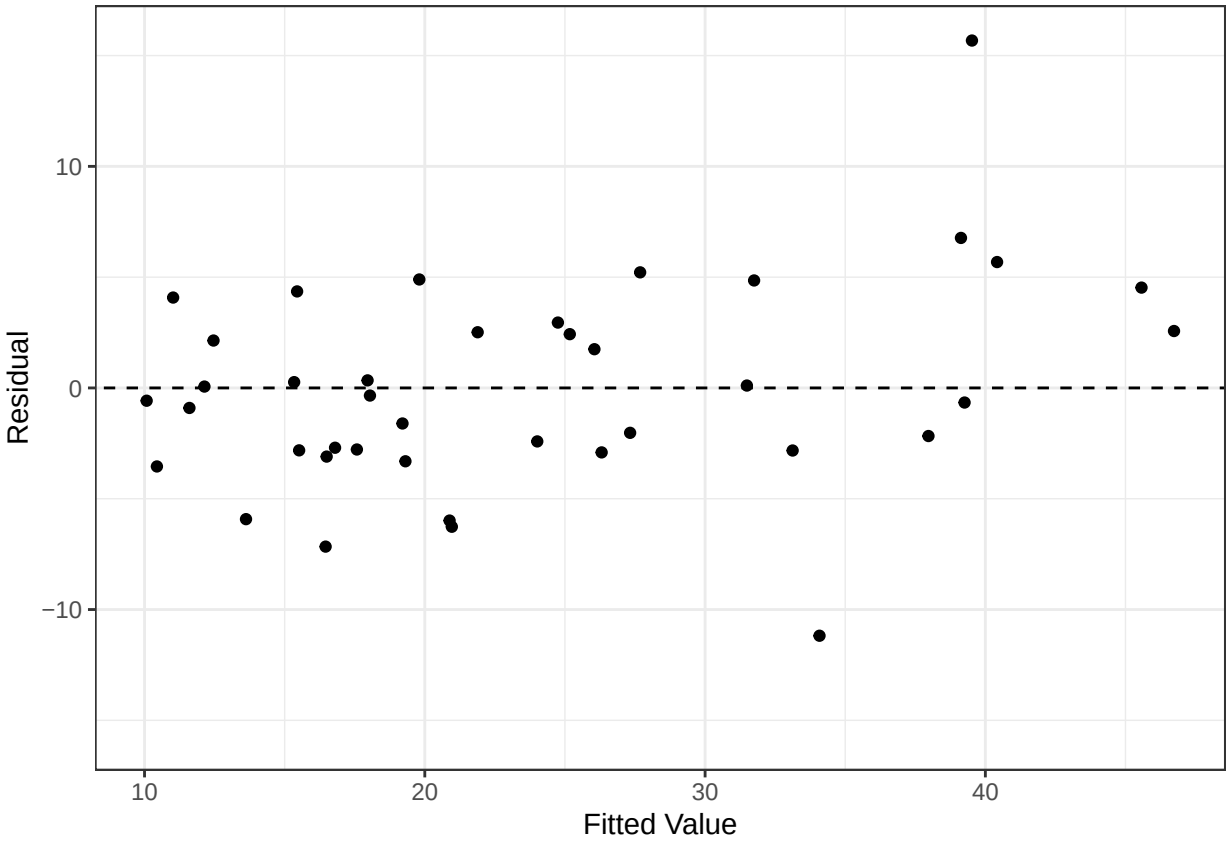


Figure 6: Verifying Homoscedasticity

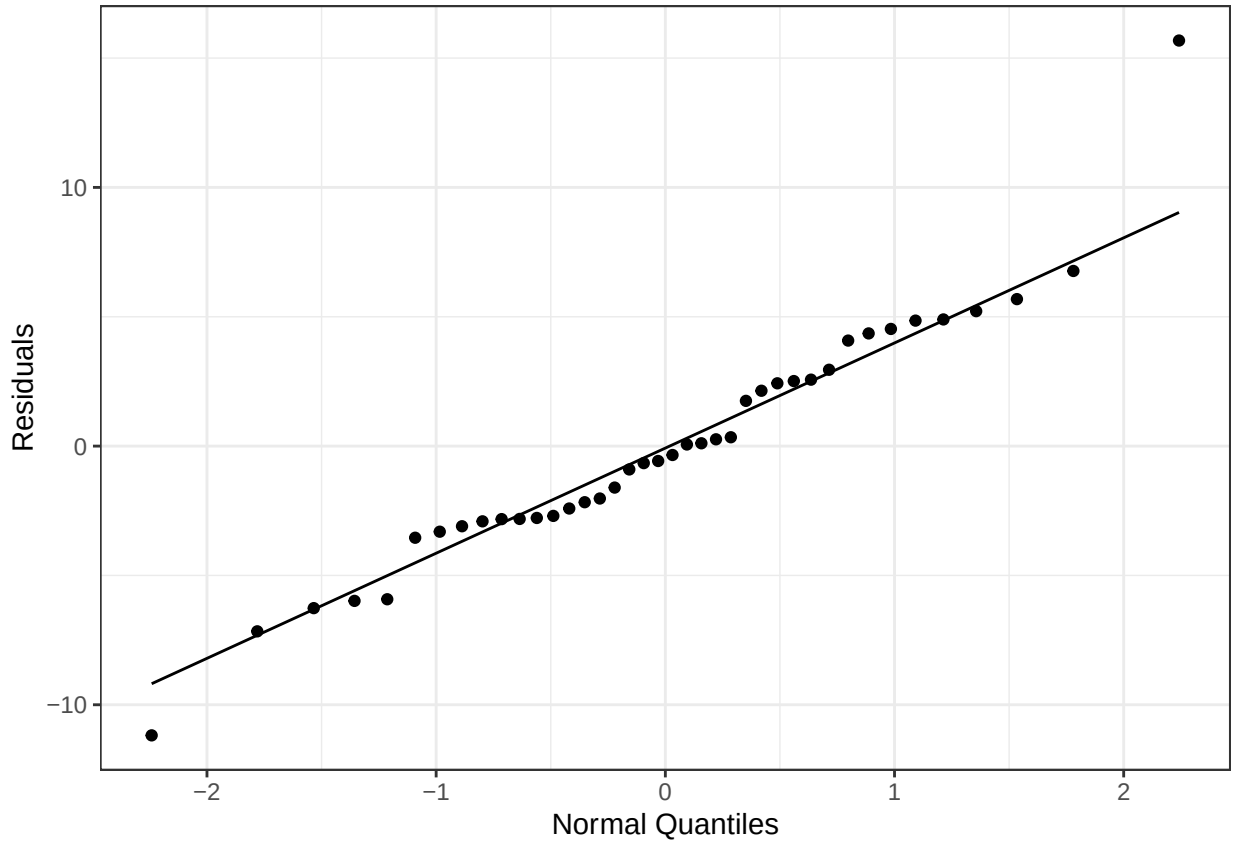


Figure 7: Q-Q Plot

Table 6: LS Means

Sequence	emmean	SE	df	lower.CL	upper.CL
C-B	24.55	3.87	18	16.42	32.68
B-C	23.12	3.87	18	14.99	31.25

Table 7: Subsample of Data with Baselines

Subject	Sequence	Period	Treatment	Pre	Post
1	C-B	1	CRICKET	20.3	18.3
1	C-B	2	BEEF	14.6	14.1
2	C-B	1	CRICKET	11.1	14.7
2	C-B	2	BEEF	11.1	24.7
4	C-B	1	CRICKET	38.3	45.9

Adjusted Means

Finally, we can report the adjusted means. The `emmeans::emmeans()` function provides adjusted means, taking as input the model and variable by which to separate the means. The results are shown in table 6.

```
library(emmeans)
emm <- emmeans(mixed.model, ~ Sequence)
```

Using Baseline Measurements

Data Structure

The study also measured insulin levels before ingestion, so we can incorporate these baseline measurements into the analysis. The format of the data with baseline (Pre) and post-ingestion insulin levels (Post) is shown in table 7. For summaries and modelling, we need the data also in 'longer' and 'wider' formats, shown in tables 8 and 9.

```
data.long <- data %>%
  pivot_longer(cols = c(Pre,Post),
               names_to = "Measurement",
               values_to = "Insulin",
               names_transform = as_factor)

data.wide <- data %>%
  pivot_wider(id_cols = c(Subject, Sequence),
              names_from = Period, values_from = c(Pre, Post)) %>%
  relocate(Post_1, .after = "Pre_1")
```

For modelling, we also need to calculate the difference in baselines for each subject. Using a host of functions from the `dplyr` package, we first calculate the difference in the 'wider' format, before using `pivot_longer()` to transform the data into the format needed for implementing the model. The resulting data is shown in table **??**

Table 8: Subsample of Data with Baselines in 'Wider' Format

Subject	Sequence	Pre_1	Post_1	Pre_2	Post_2
1	C-B	20.3	18.3	14.6	14.1
2	C-B	11.1	14.7	11.1	24.7
4	C-B	38.3	45.9	51.6	35.8
5	C-B	40.3	46.1	51.8	38.6
7	C-B	24.6	17.6	31.3	17.7

Table 9: Subsample of Data with Baselines in 'Longer' Format

Subject	Sequence	Period	Treatment	Measurement	Insulin
1	C-B	1	CRICKET	Pre	20.3
1	C-B	1	CRICKET	Post	18.3
1	C-B	2	BEEF	Pre	14.6
1	C-B	2	BEEF	Post	14.1

```
data.baselines <- data.wide %>%
  mutate(baseline_diff = Pre_1 - Pre_2) %>%
  rename(`1` = "Post_1", `2` = "Post_2") %>%
  pivot_longer(cols = c(`1`,`2`), names_to = "Period", values_to = "Post",
               names_transform = as_factor) %>%
  select(Subject, Period, baseline_diff) %>%
  inner_join(data, join_by(Subject == Subject, Period == Period)) %>%
  relocate(Sequence, .after = "Subject") %>%
  relocate(baseline_diff, .after = "Post")
```

Summarising with Baselines

The summary table can be expanded to include the baseline measurements, as shown in table 11.

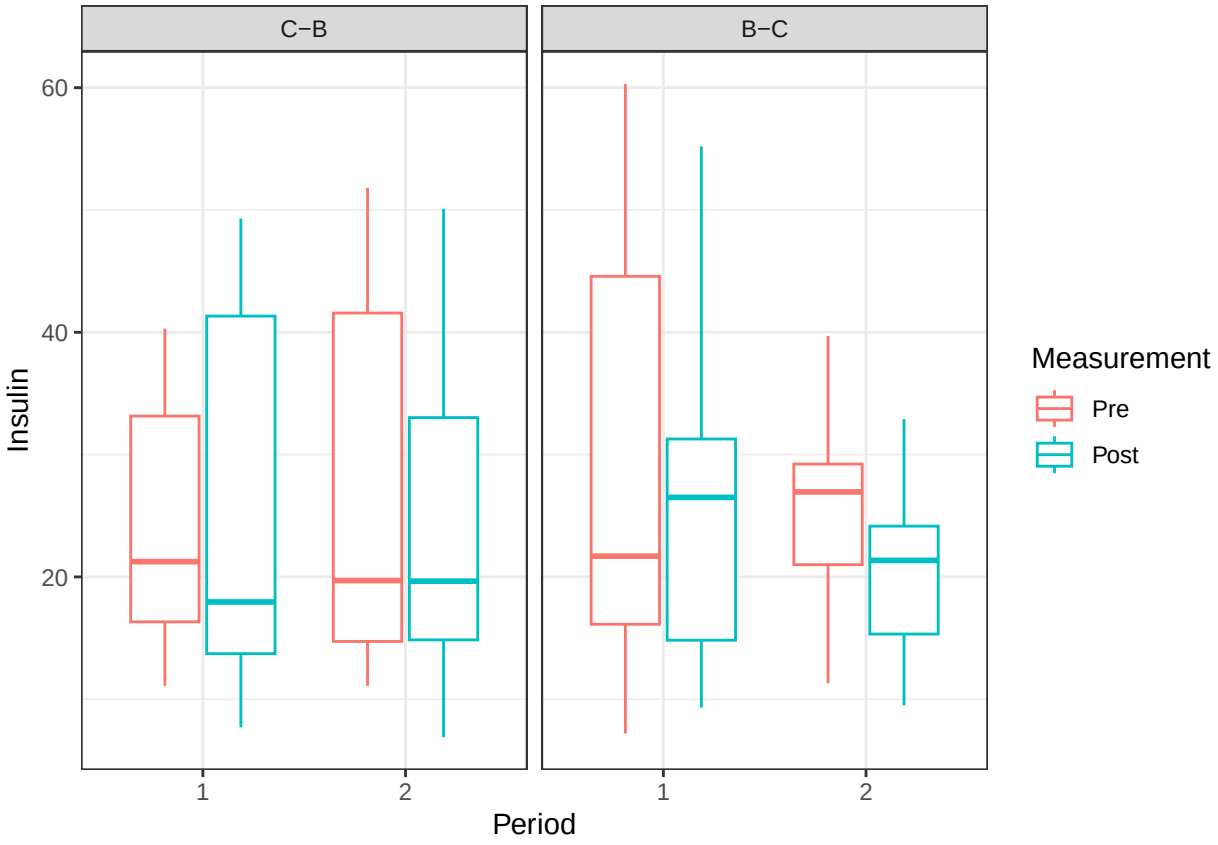
We can also extend the boxplot to incorporate the baseline measurements (see figure **??**.

Table 10: Subsample of Data with Baseline Differences

Subject	Sequence	Period	Treatment	Pre	Post	baseline_diff
1	C-B	1	CRICKET	20.3	18.3	5.7
1	C-B	2	BEEF	14.6	14.1	5.7
2	C-B	1	CRICKET	11.1	14.7	0.0
2	C-B	2	BEEF	11.1	24.7	0.0
4	C-B	1	CRICKET	38.3	45.9	-13.3

Table 11: Summary Table with Baseline Values (Pre)

		Overall				Period 1				Period 2			
		Pre		Post		Pre		Post		Pre		Post	
Sequence	Subject	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
C-B	10	25.71	13.53	24.55	14.43	24.03	10.70	25.13	16.07	27.39	16.31	23.97	13.45
B-C	10	27.91	14.39	23.12	11.11	29.73	18.94	25.84	13.83	26.10	8.44	20.40	7.27
Total	20	26.81	13.83	23.84	12.74	26.88	15.25	25.48	14.60	26.75	12.66	22.18	10.68



Modelling with Baselines

First we construct the ANOVA table, this time including the baseline by period interaction, as shown in table **??**.

```
anova.table.baseline <- aov(Post ~ Treatment + Period * baseline_diff + Sequence + Subject,
                            data = data.baselines) %>%
  summary()
```

We update the mixed model to include the period-by-baseline difference interaction. Estimates are shown in table **??**, and the updated LS means in table 12.

Table 12: LS Means

Sequence	emmean	SE	df	lower.CL	upper.CL
C-B	24.01	4.03	17	15.49	32.52
B-C	23.66	4.03	17	15.15	32.18

```
mixed.model.baselines <- lmer(Post ~ Treatment + Period * baseline_diff + Sequence + (1|Subject),
              data = data.baselines)
```