Contrasts and ANOVA

Neil J. Hatfield

Last Updated: March 20, 2023

In this tutorial, we are going to explore setting up and testing Contrasts in the context of ANOVA models **using parametric shortcuts *only***.

1 Setting Up R and Loading Data

For this particular guide/tutorial, we will load (most) of our usual packages: `tidyverse`, `knitr`, and `kableExtra`. We will also load `parameters`, `emmeans`, `DescTools`, and the `effectsize` packages.

```
# Demo code for setting up R for contrasts ----
packages <- c("tidyverse", "knitr", "kableExtra",
              "parameters", "emmeans", "DescTools",
              "effectsize")
lapply(packages, library, character.only = TRUE, quietly = TRUE)

options(knitr.kable.NA = "")
options(contrasts = c("contr.sum", "contr.poly"))

source("https://raw.github.com/neilhatfield/STAT461/master/rScripts/ANOVATools.R")
```

Don't forget that we also need to set options including the all important constraint that our factor effects must add to zero (i.e. `c("contr.sum", "contr.poly")`).

1.1 Loading Data

We will make use of the **Free Amino Acids in Cheese** (i.e., the Cheese Study; pg. 91 of Oehlert) for the examples in this guide. You will also want to load the Song Knowledge study data for the final example. For this guide, I'm not going to show the loading code here. However, if you get stuck in writing the code, check out the Code Appendix at the end of the guide/tutorial.

2 Key Decisions for Post Hoc Analysis

Just as with pairwise approaches to Post Hoc analysis, we have the same sets of steps to take for contrasts.

2.1 Study Design Decisions

Our first steps for contrasts is to make sure that contrasts are actually warranted by our research questions. Notice that I wrote the plural form of question. In the pairwise case of Post Hoc analysis, the pairwise SRQs are implied by our main SRQ. The same is not true for contrasts: we actually need to articulate what contrasts we're interested in investigating.

> In addition to investigating whether the strain of starting bacteria has an impact on the amount of free amino acids in the cheese, we have two additional research questions. First, we want to know whether having any of the two starter strains leads to a difference in free amino acids as compared to not having either starter strains. Second, we want to also investigate if getting both strains is statistically different from getting one of the two strains.

The above paragraph lays out the statistical research "questions" (in statement form) for two contrasts. The first contrast proposes combining three of the levels of our factor (i.e., A, B, and both–AB) together and comparing them to the fourth level (i.e., none). The second question has us compare the level of both strains (i.e., AB) against the combination of individual strains (i.e., A and B). (Yes, the wording for this second question is rather close to what we might anticipate for pairwise comparisons.)

As an example for an algebraic form of the hypotheses for contrasts, consider the following for the first Cheese study contrasts:

$$H_0: \quad \frac{\mu_A + \mu_B + \mu_{AB}}{3} = \mu_N$$
$$H_1: \quad \frac{\mu_A + \mu_B + \mu_{AB}}{3} \neq \mu_N$$

2.1.1 Testing Families

Just as with pairwise comparison based Post Hoc analysis, we still need to conceptualize a testing family for contrasts. One approach you can take is to simply imagine all of your contrasts belonging to one testing family. This can give you two testing families for Post Hoc analysis: one for (all) the pairwise comparisons and one for contrasts. Once you conceptualize your testing family of contrasts, be sure you record the number of comparisons, m. For the above example, $m = 2$.

2.1.2 Choose Your Type I Error Rate and Method

Just as with pairwise Post Hoc analysis, we still have to select a Type I Error Rate and method. The table from the Post Hoc guide/tutorial appears below. I've placed the Scheffé method in bold face as this method is built especially for contrasts and guards against any data snooping that might have occurred.

Selected Type I Error Rate	Possible Parametric Methods
Simultaneous Confidence Intervals	Bonferroni, Šidák, Tukey HSD, Tukey-Kramer HSD, **Scheffé**
Strong Familywise/Maximum Experimentwise	Hochberg, Holm, REGWR, Gabriel, Dunnett, DSCF
False Discovery Rate	Benjamini-Hochberg, Student-Newman-Keuls
Experimentwise Error Rate	[ANOVA Omnibus Tests,] Protected LSD
Comparisonwise Error Rate	Most Two Sample Tests, Unprotected LSD

After picking which Error Rate you want to control, you choose an appropriate method. There are some pieces of guidance you can follow for choosing the particular method. However, a lot comes down to personal preference.

2.2 Checking Appropriateness and Assumptions

Again, we'll bank upon the appropriateness checks and your assessment of assumptions for the omnibus test (i.e., the ANOVA F test) for our contrasts.

As a refresher for the Cheese study, here is a reproduction of the modern ANOVA table for this study:

Table 2: ANOVA Table for Free Amino Acids in Cheese Study

Source	SS	df	MS	F	p-value	Eta Sq.	Omega Sq.	Epsilon Sq.
strain	5.6279	3	1.8760	11.9318	0.0183	0.8995	0.8039	0.8241
Residuals	0.6289	4	0.1572					

3 Setting Up Your Contrasts

There are two parts to setting up your contrast in R: making sure you know the correct order of your factor levels and then saving the weights.

3.1 Check the Order

Checking the order of your factor levels is a quick application of the `levels` function:

```
# Demo code to check the order of factor levels ----
levels(cheeseData$strain)
```

```
## [1] "A"    "AB"   "B"    "None"
```

The output tells us that our contrasts will need to be in the order (A, AB, B, None).

You must always check the ordering that the software is using. Suppose that we used "(A & B)" instead of "AB" for both strains together. The ordering would then be ((A & B), A, B, None). Any changes you make to the names of the levels will impact the ordering.

3.2 Save the Weights

Now that we know the order of the factor levels, we can create contrasts by saving the weights for each level as a vector. For example,

```
# Demo code to save the contrasts weights as vectors ----
c1 <- c(1/3, 1/3, 1/3, -1)
c2 <- c(-1/2, 1, -1/2, 0)
```

The first contrast, `c1`, pools the cheeses that received either Strain A, Strain B, or Both together and compares them against those which received the base set of cultures ("None"). The second contrast, `c2`, compares the pooling of Strain A and Strain B against the combination of both strains. By saving the contrasts, they are now available for our use.

4 Checking Your Contrasts

There are a couple of conditions that we need to check with our contrasts before we actually test them.

4.1 Weights Sum to Zero

The first check we need to do is make sure that contrasts weights sum to zero. You can do this visually/mentally, or you can ask `R` to do so. One argument to have `R` do this check is to make sure you didn't make a typo when programming the contrasts. You can perform this with the `sum` function:

```
# Demo code to check that the weights add to zero ----
sum(c1)
```

```
## [1] -5.551115e-17
```

```
sum(c2)
```

```
## [1] 0
```

Notice that `sum(c1)` is a non-zero value. This is a byproduct of computer arithmetic. Given that this value is -5.55×10^{-17}, we'll go ahead and say this is zero.

On the other hand, if you were to see a sum such as 0.072, that would be sign that something is wrong with our weights.

4.2 Checking Orthogonality

When we have multiple contrasts, we often like to have "orthogonal contrasts". We like orthogonal contrasts for two reasons. First, this means that they are independent from each other. Second, orthogonal contrasts will perfectly partition the sums of squares for the factor they are applied to. This fact is especially useful in situations where we want to compare a set of new treatments to a standard care or null treatment.

Two contrasts, w_i and $w_i^\star$, are said to be orthogonal if the weighted sum of their i-th components add to zero (the weighted dot product of the two contrast vectors). That is,

$$\sum_{i=1}^{g} \frac{w_i \cdot w_i^\star}{n_i}$$

In this formula, the weighting of the weights comes from the sample size for each component, n_i. For our two Cheese study contrasts, we would have the following:

$$\begin{aligned}\sum_{i=1}^{4} \frac{w_i \cdot w_i^\star}{n_i} \Rightarrow \frac{1/3 * 1}{2} + \frac{1/3 * -1/2}{2} + \frac{1/3 * -1/2}{2} + \frac{-1 * 0}{2} \\ = \frac{1}{6} + \frac{-1}{12} + \frac{-1}{12} + 0 \\ = 0\end{aligned}$$

Thus our two contrasts, `c1` and `c2`, are orthogonal to each other.

We can use `R` to help us check orthogonality of our contrasts:

```
# Demo code for checking orthogonality of two contrasts ----
## Create a vector of sample sizes in the same order as the contrast vectors
n <- c(2,2,2,2)

## Use the definition (returns a number)
sum((c1*c2)/n)
```

```
## [1] 0
```

```
## Using weighted dot product (returns a 1x1 matrix)
(c1/n) %*% (c2 / n)
```

```
##      [,1]
## [1,]    0
```

5 Testing Your Contrasts in `R`

Once you have set up your contrasts, you have three routes to take for testing them: use the `emmeans` package, use base `R`, or use the `DescTools` package. Regardless of which approach you use, the general steps are still the same:

1) Check the order of your factor levels.
2) Construct your contrast weight vectors
3) Apply the contrasts (difference exist here)
4) Report the results

5.1 The `emmeans` Approach

The `emmeans` package's approach to contrasts comes with several benefits:

1) We may apply any of the MC/SI Problem adjustment methods that we would use for pairwise post hoc analysis.

2) The approach is extendable to k-way ANOVA (more complicated designs) as well as doing contrasts on interaction terms or with conditions.
3) We can add on measures of effect size to each contrast.

```
# Demo Code for using the emmeans package for contrasts ----
## Get the appropriate means from the model
cheeseMeans <- emmeans::emmeans(
  object = cheeseModel,
  specs = ~ strain # Nothing goes on the left of ~; list what term you want
)

# cheeseMeans # Look at the output object to double check the order of levels
## Should match the order of levels(cheeseData$strain)

## Apply the contrasts
cheeseContrasts1 <- emmeans::contrast(
  object = cheeseMeans, # Notice that this is the means object
  method = list(
    "Starter Bacteria vs. None" = c(1/3, 1/3, 1/3, -1), # Alt. c1
    "Both Strains vs. Only One" = c(-1/2, 1, -1/2, 0) # Alt. c2
  ),
  adjust = "fdr" # MC/SI Adjustment method
)

## Add effect sizes and make a nice looking table
as.data.frame(cheeseContrasts1) %>%
  dplyr::mutate(
    cohen = effectsize::t_to_d(t = t.ratio, df_error = df)$d, # Effect Sizes
    ps = probSup(cohen) # Effect sizes; this comes from Neil's ANOVA toolkit
  ) %>%
  kable(
    digits = 3,
    caption = "Cheese Study Contrasts Using emmeans Package",
    col.names = c("Contrast", "Difference", "SE", "DF", "t Statistic", "p-value*",
                  "Cohen's d", "Prob. of Superiority"),
    align = "lccccccc",
    booktabs = TRUE
  ) %>%
  kableExtra::kable_styling(
    bootstrap_options = c("striped", "condensed"),
    font_size = 12,
    latex_options = c("HOLD_position", "scale_down")
  ) %>%
  kableExtra::footnote(
    general = "Adjusted via Benjamini-Hochberg method for controlling the False Discovery Rate.",
    general_title = "* ",
    footnote_as_chunk = TRUE
  )
```

Table 3: Cheese Study Contrasts Using emmeans Package

Contrast	Difference	SE	DF	t Statistic	p-value*	Cohen's d	Prob. of Superiority
Starter Bacteria vs. None	1.167	0.324	4	3.605	0.023	3.605	0.995
Both Strains vs. Only One	1.454	0.343	4	4.234	0.023	4.234	0.999

* Adjusted via Benjamini-Hochberg method for controlling the False Discovery Rate.

The `method` argument of `emmeans::contrast` is critical. This is not only where you apply your contrasts but also where you can give them meaningful names.

In Table 3, we can see several columns worth note. The Difference column reports the difference between the two meta-groups (meta-treatments). For example, there were 1.167 more units of free amino acids per cheese when we used at least one of the starter bacteria than when we didn't. The next three columns provide information about the hypothesis test (*standard error* (of the difference), *degrees of freedom*, and the t statistic). We can then see the *adjusted* p-value. (Remember that the null hypothesis for of our linear contrasts is that there is no statistically significant difference between the meta-groups.)

The last two columns provide us with the effect sizes with Cohen's d and the Probability of Superiority. Keep in mind that we set the order of comparison with the positive/negative signs in our contrasts vectors. Since we used `(1/3, 1/3, 1/3, -1)`, then 99.5% of the time we repeatedly sample a cheese that got a starter bacteria and cheese that did not get any starter bacteria, the cheese with the starter will have more free amino acids.

5.2 The Base R Approach

Using Base R to do contrasts involves a few different steps than using the `emmeans` package. One of the benefits of using this approach is that you can place the contrasts into an ANOVA table, rather than a separate table.

After we create our contrasts, we need to connect them to our factor. There's nothing inherent in how we constructed the vectors that says "Hey, these are for the strain of bacteria!". We have to tell R that this is what we want. To do this, we'll used the `contrasts` function:

```
# Demo code for setting contrasts in the Cheese study ----
## Base R Approach
contrasts(cheeseData$strain) <- cbind(c1, c2) #cbind makes a contrast matrix

# Generic example
# contrasts(dataFrame$factor) <- matrixOfContrasts
```

If you had previously run your ANOVA model, you need to re-run the model after connecting the contrasts to the factor. This will apply the appropriate parametric shortcut for the contrasts.

```
# Demo code for testing contrasts in the Cheese study ----
## Base R approach
cheeseContrasts2 <- aov(
  formula = acids ~ strain,
  data = cheeseData
)
```

Now we get to a bit of a tricky point: we need to isolate the appropriate portion of the model results. For quick examinations we would use `summary` to do this. However, neither of these methods will display the tests for the contrasts. Thus, we need to make use of a new argument: `split`

```
# Demo Code for Getting Contrast Results ----
## Base R approach
contrastOut <- summary(
  object = cheeseContrasts2,
  split = list( # Apply meaningful labels to your contrasts here
    strain = list(
      "Starter Bacteria vs. None" = 1, # 1 means first position in the cbind above
      "Both Strains vs. Only One" = 2 # 2 means second position in the cbind above
    )
  )
)
```

The `split` argument allows us to call the contrasts we want to explore. The structure of the `split` argument is a list of lists. The outer list has elements which start with the name of the factor; `stain = list(`. The inner lists begin with the meaningful labels (e.g., `"Starter Bacteria vs. None"`) and then a number which corresponds to order in which you listed the weight vectors in the `cbind` call.

Now that we have stored the output that we need, `contrastOut`, we can impose our adjustments for the multiple comparison problem and create a professional looking table.

```r
# Demo code for adjusting p-values and making a professional table ----
## Base R Approach
adjustPValues(
  contrastObject = contrastOut,
  method = "fdr"
) %>%
  knitr::kable(
    digits = 3,
    col.names = c(
      "DF", "SS", "MS", "F", "Raw p-value", "Adj. p-value"),
    caption = "ANOVA Table for Free Amino Acids in Cheese Study",
    booktabs = TRUE,
    align = rep("c", 5)
  ) %>%
  kableExtra::kable_styling(
    font_size = 12,
    latex_options = c("HOLD_position")
  )
```

Table 4: ANOVA Table for Free Amino Acids in Cheese Study

	DF	SS	MS	F	Raw p-value	Adj. p-value
strain	3	5.628	1.876	11.932	0.018	
strain: Starter Bacteria vs. None	1	2.043	2.043	12.997	0.023	0.023
strain: Both Strains vs. Only One	1	2.819	2.819	17.929	0.013	0.023
Residuals	4	0.629	0.157			

There are few important notes with this approach. Since this approach places the contrasts inside a (classical) ANOVA table, `R` has used an F test while `emmeans` used a t test. If you take the square root of F, you'll get the equivalent value of the t statistic for the contrasts.

The `adjustPValues` function comes from my set of helper tools (`ANOVATools`). You need to pass this function the output of the `summary` call with the `split` argument and you need to state which method you want to use. The `method` argument supports all of the following methods:

Chosen Method	Set `method =`
Bonferroni	`"bonferroni"`
Holm	`"holm"`
Hochberg	`"hochberg"`
Benjamini & Hochberg	`"BH"` OR `"fdr"`

We can pipe the output of `adjustPValues` into a `kable` call to make our professional table. In Table 4, the two contrasts appear as the second and third rows of our table.

5.2.1 Important Limitation

Notice that Table 4 does not show any effect sizes. If you look at the generating code above the table, you'll notice that there was no call to the `parameters::model_parameters` function. When we created the `constrastOut` object, we ended up making an object that isn't compatible with the `parameters` package. Thus, if you want to add in appropriate effect sizes, you would want to create a new custom data frame.

5.3 `DescTools` Package

The `DescTools` package includes the `ScheffeTest` function that will allow you test your contrasts using the Scheffé method to control the SCI Type I Error Rate, or to account for data snooping.

Unlike the base package approach, you will not get an ANOVA table, but rather, you'll get a table like what you would see when doing pairwise post hoc analysis. The process here has three steps:

1) Fit your ANOVA model just as you typically would.
    - You do not need to bind your contrasts to your factor beforehand.
2) Save the output of the `DescTools::ScheffeTest` to an object.
3) Make a professional looking table.

Examine the following example code:

```
# Demo code for Scheffe Testing Contrasts ----
## DescTools Scheffe Test Approach
## Step 1 -- Fit the ANOVA model as usual
cheeseModel <- aov(
  formula = acids ~ strain,
  data = cheeseData,
  na.action = "na.omit"
)

## Step 2--Save output of Scheffe Test
scheffeCheese <- DescTools::ScheffeTest(
  x = cheeseModel,
  contrasts = cbind(c1, c2),
  conf.level = 0.9, # 1 -- Your Overall Type I Error Rate
)

## Step 3--Make a Professional looking table
knitr::kable(
  x = scheffeCheese[[1]], # Grab the output
  digits = 4,
  col.names = c(
    "Difference", "Lower Bound", "Upper Bound", "p-value"),
  caption = "Scheffe Test Results",
  booktabs = TRUE,
  align = rep("c", 4)
  ) %>%
  kableExtra::kable_styling(
    font_size = 12,
    latex_options = c("HOLD_position")
  )
```

Table 6: Scheffe Test Results

	Difference	Lower Bound	Upper Bound	p-value
A,AB,B-None	1.1672	0.0192	2.3151	0.0953
AB-A,B	1.4540	0.2364	2.6716	0.0585

Notice that the "names" for the contrasts were automatically appended to the left side of Table 6. The `ScheffeTest` function automatically used the contrast vectors to figure out the the appropriate "names".

A second thing to notice in Table 6 is the presence of confidence intervals for the difference. These are simultaneous 90% confidence intervals for the two differences between meta-groups.

5.3.1 Important Limitation

Just as with the base `R` approach, notice that there are no effect sizes listed here. If you want effect sizes for contrasts, you'd either need to make a custom object here or use the `emmeans` approach.

6 Reporting and Interpreting Results

For this example, I'm going to suppose that we choose to use the Scheffé method to control our Type I Error rate at no more than 10%.

Given the results in Table 6, we can see that for our first contrast, adding either Strain A, Strain B, or both strains of starting bacteria, appear to lead to higher levels of free amino acids in the cheese than the naturally occurring cultures (i.e., "none" for additional strains). We would anticipate getting a difference at least as large (magnitude) around 9.5% of the time when there is no difference between the three-way combination and no additional bacteria.

For our second question, we see a slight more unusual result. Comparing the joint treatment of both strains with the combination of the individual inoculations, we anticipate observing a difference at least as large (in magnitude) as 1.45 units/cheese around 6% of the time if there actually was no difference. This suggests that if we want more umami-rich and firmer cheese (the result of higher free amino acids), then we should use both Strain and Strain B in the cheese making process rather than just one or the other.

7 Putting Things Together–Your Turn

Use what you've seen here in the context of the Song Knowledge study to test the contrast of graduating students (seniors) vs. non-graduating students (juniors, others).

Note: since there is just one contrast in this family, you won't have to worry about adjustments to p*-values.*

8 Code Appendix

```
# Setting Document Options ----
knitr::opts_chunk$set(
  echo = FALSE,
  warning = FALSE,
  message = FALSE,
  fig.align = "center"
)

packages <- c("tidyverse", "knitr", "kableExtra",
              "parameters", "emmeans", "DescTools",
              "effectsize")
lapply(packages, library, character.only = TRUE, quietly = TRUE)

options(knitr.kable.NA = "")
options(contrasts = c("contr.sum", "contr.poly"))

source("https://raw.github.com/neilhatfield/STAT461/master/rScripts/ANOVATools.R")
# Demo code for setting up R for contrasts ----
packages <- c("tidyverse", "knitr", "kableExtra",
              "parameters", "emmeans", "DescTools",
              "effectsize")
lapply(packages, library, character.only = TRUE, quietly = TRUE)

options(knitr.kable.NA = "")
options(contrasts = c("contr.sum", "contr.poly"))

source("https://raw.github.com/neilhatfield/STAT461/master/rScripts/ANOVATools.R")

# Demo Code for loading data ----
## Create the Cheese data frame
cheeseData <- data.frame(
  strain = as.factor(
    c("None", "None", "A", "A",
      "B", "B", "AB", "AB")
  ),
  acids = c(
    4.195, 4.175, 4.125, 4.735,
    4.865, 5.745, 6.155, 6.488
  )
)

## Song Knowledge study
songData <- read.csv(
file = "https://raw.github.com/neilhatfield/STAT461/master/dataFiles/songKnowledge_Sp23.csv",
header = TRUE,
sep = ","
)
### Set year to an ordered factor
songData$Year <- factor(
  x = songData$Year,
  levels = c("Junior", "Senior", "Other")
)

# Demo Code for Fitting Oneway ANOVA Model ----
## Cheese study
cheeseModel <- aov(
  formula = acids ~ strain,
```

```
  data = cheeseData,
  na.action = "na.omit"
)

## ANOVA Table
parameters::model_parameters(
  model = cheeseModel,
  effectsize_type = c("eta", "omega", "epsilon")
) %>%
  knitr::kable(
  digits = 4,
  col.names = c(
    "Source", "SS", "df", "MS", "F", "p-value",
    "Eta Sq.", "Omega Sq.", "Epsilon Sq."),
  caption = "ANOVA Table for Free Amino Acids in Cheese Study",
  booktabs = TRUE,
  align = c("l", rep("c", 8))
  ) %>%
  kableExtra::kable_styling(
    font_size = 12,
    latex_options = c("HOLD_position")
  )

# Demo code to check the order of factor levels ----
levels(cheeseData$strain)

# Demo code to save the contrasts weights as vectors ----
c1 <- c(1/3, 1/3, 1/3, -1)
c2 <- c(-1/2, 1, -1/2, 0)

# Demo code to check that the weights add to zero ----
sum(c1)
sum(c2)

# Demo code for checking orthogonality of two contrasts ----
## Create a vector of sample sizes in the same order as the contrast vectors
n <- c(2,2,2,2)

## Use the definition (returns a number)
sum((c1*c2)/n)

## Using weighted dot product (returns a 1x1 matrix)
(c1/n) %*% (c2 / n)

# Demo Code for using the emmeans package for contrasts ----
## Get the appropriate means from the model
cheeseMeans <- emmeans::emmeans(
  object = cheeseModel,
  specs = ~ strain # Nothing goes on the left of ~; list what term you want
)

# cheeseMeans # Look at the output object to double check the order of levels
## Should match the order of levels(cheeseData$strain)

## Apply the contrasts
cheeseContrasts1 <- emmeans::contrast(
  object = cheeseMeans, # Notice that this is the means object
  method = list(
    "Starter Bacteria vs. None" = c(1/3, 1/3, 1/3, -1), # Alt. c1
```

```
    "Both Strains vs. Only One" = c(-1/2, 1, -1/2, 0) # Alt. c2
  ),
  adjust = "fdr" # MC/SI Adjustment method
)

## Add effect sizes and make a nice looking table
as.data.frame(cheeseContrasts1) %>%
  dplyr::mutate(
    cohen = effectsize::t_to_d(t = t.ratio, df_error = df)$d, # Effect Sizes
    ps = probSup(cohen) # Effect sizes; this comes from Neil's ANOVA toolkit
  ) %>%
  kable(
    digits = 3,
    caption = "Cheese Study Contrasts Using emmeans Package",
    col.names = c("Contrast", "Difference", "SE", "DF", "t Statistic", "p-value*",
                  "Cohen's d", "Prob. of Superiority"),
    align = "lccccccc",
    booktabs = TRUE
  ) %>%
  kableExtra::kable_styling(
    bootstrap_options = c("striped", "condensed"),
    font_size = 12,
    latex_options = c("HOLD_position", "scale_down")
  ) %>%
  kableExtra::footnote(
    general = "Adjusted via Benjamini-Hochberg method for controlling the False Discovery Rate.",
    general_title = "* ",
    footnote_as_chunk = TRUE
  )
# Demo code for setting contrasts in the Cheese study ----
## Base R Approach
contrasts(cheeseData$strain) <- cbind(c1, c2) #cbind makes a contrast matrix

# Generic example
# contrasts(dataFrame$factor) <- matrixOfContrasts

# Demo code for testing contrasts in the Cheese study ----
## Base R approach
cheeseContrasts2 <- aov(
  formula = acids ~ strain,
  data = cheeseData
)

# Demo Code for Getting Contrast Results ----
## Base R approach
contrastOut <- summary(
  object = cheeseContrasts2,
  split = list( # Apply meaningful labels to your contrasts here
    strain = list(
      "Starter Bacteria vs. None" = 1, # 1 means first position in the cbind above
      "Both Strains vs. Only One" = 2 # 2 means second position in the cbind above
    )
  )
)

# Demo code for adjusting p-values and making a professional table ----
## Base R Approach
adjustPValues(
  contrastObject = contrastOut,
```

```
  method = "fdr"
) %>%
  knitr::kable(
    digits = 3,
    col.names = c(
      "DF", "SS", "MS", "F", "Raw p-value", "Adj. p-value"),
    caption = "ANOVA Table for Free Amino Acids in Cheese Study",
    booktabs = TRUE,
    align = rep("c", 5)
  ) %>%
  kableExtra::kable_styling(
    font_size = 12,
    latex_options = c("HOLD_position")
  )

# Demo code for Scheffe Testing Contrasts ----
## DescTools Scheffe Test Approach
## Step 1 -- Fit the ANOVA model as usual
cheeseModel <- aov(
  formula = acids ~ strain,
  data = cheeseData,
  na.action = "na.omit"
)

## Step 2--Save output of Scheffe Test
scheffeCheese <- DescTools::ScheffeTest(
  x = cheeseModel,
  contrasts = cbind(c1, c2),
  conf.level = 0.9, # 1 -- Your Overall Type I Error Rate
)

## Step 3--Make a Professional looking table
knitr::kable(
  x = scheffeCheese[[1]], # Grab the output
  digits = 4,
  col.names = c(
    "Difference", "Lower Bound", "Upper Bound", "p-value"),
  caption = "Scheffe Test Results",
  booktabs = TRUE,
  align = rep("c", 4)
  ) %>%
  kableExtra::kable_styling(
    font_size = 12,
    latex_options = c("HOLD_position")
  )
```