Chapter 8 Comparing means

Adapted by Abigail Noyce from *Fundamentals of Quantitative Analysis*, James Bartlett and Wilhelmiina Toivo.[26]

Inferential statistics are tools that let us work from a limited data set to draw conclusions about the broader population from which those data were sampled. One of the most common families of tools are those that let us compare multiple sample means (such as those derived from different groups of subjects, or different experimental conditions). This reading walks through those common tools, the R code to apply them, and how to interpret the resulting output. For all of these cases, the predictive (independent) variable is assumed *categorical*: we are asking questions about two different groups.

All of these case-specific tools are applications of the *general linear model*, are a family of tools to predict one or more continuously-measured outcome variables from one or more predictors (which can be categorical or continuous).

8.1 Comparing two groups using *t*-tests

For practice, we're going to look at data from Lopez et al. (2023).[27] This paper replicated an (in)famous experiment that won the Ig-Nobel prize. Participants engaged in a intricate setting (seriously, go and look at the diagrams in the article) where they ate soup from bowls on a table. In the control group, participants could eat as much soup as they wanted and could ask for a top-up from the researchers. In the experimental group, the soup bowls automatically topped up through a series of hidden tubes under the table. The idea behind the control group is they get an accurate visual cue by the soup bowl reducing, and the experimental group get an inaccurate visual cue by the soup bowl seemingly never reducing. So, the inaccurate visual cue would interfere with natural signs of getting full and lead to people eating more.

In the original article, participants in the experimental group ate more soup than participants in the control group, but the main author was involved in a series of research misconduct cases.[28] Lopez et al. (2023) wanted to see if the result would replicate in an independent study, so they predicted they would find the same results.

Our research question is thus “Is there a difference in actual calories consumed between the control and experimental groups?” What are the null hypothesis ($H_0$) and alternative hypothesis ($H_1$) that correspond to this research question?

8.1.1 Initial exploration and descriptive statistics

Read the data file.

```
datafile <- here("datasets/lopez/Lopez_2023.csv")
lopez_data <- read.csv(datafile)
glimpse(lopez_data)
```

```
## Rows: 464
## Columns: 9
## $ ParticipantID      <int> 1002, 1004, 1007, 1016, 1018, 1021, 1022, 1024, 102…
## $ Sex                <int> 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, …
## $ Age                <int> 18, 19, 19, 21, 20, 20, 21, 21, 19, 20, 21, 20, 21,…
## $ Ethnicity          <int> 7, 3, 3, 4, 1, 3, 1, 6, 4, 7, 1, 3, 3, 4, 7, 2, 3, …
## $ OzEstimate         <dbl> 3.0, 2.0, 1.0, 3.0, 5.0, 1.0, 1.0, 3.0, 4.0, 1.0, 4…
## $ CalEstimate        <dbl> 65, 10, 20, 25, 50, 5, 20, 180, 470, 50, 130, 100, …
## $ M_postsoup         <dbl> 3.3, 3.1, 43.4, 5.5, 6.0, 0.8, 3.8, 4.5, 7.9, 8.1, …
## $ F_CaloriesConsumed <dbl> 73.19441, 68.75839, 962.61743, 121.99069, 133.08075…
## $ Condition          <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …
```

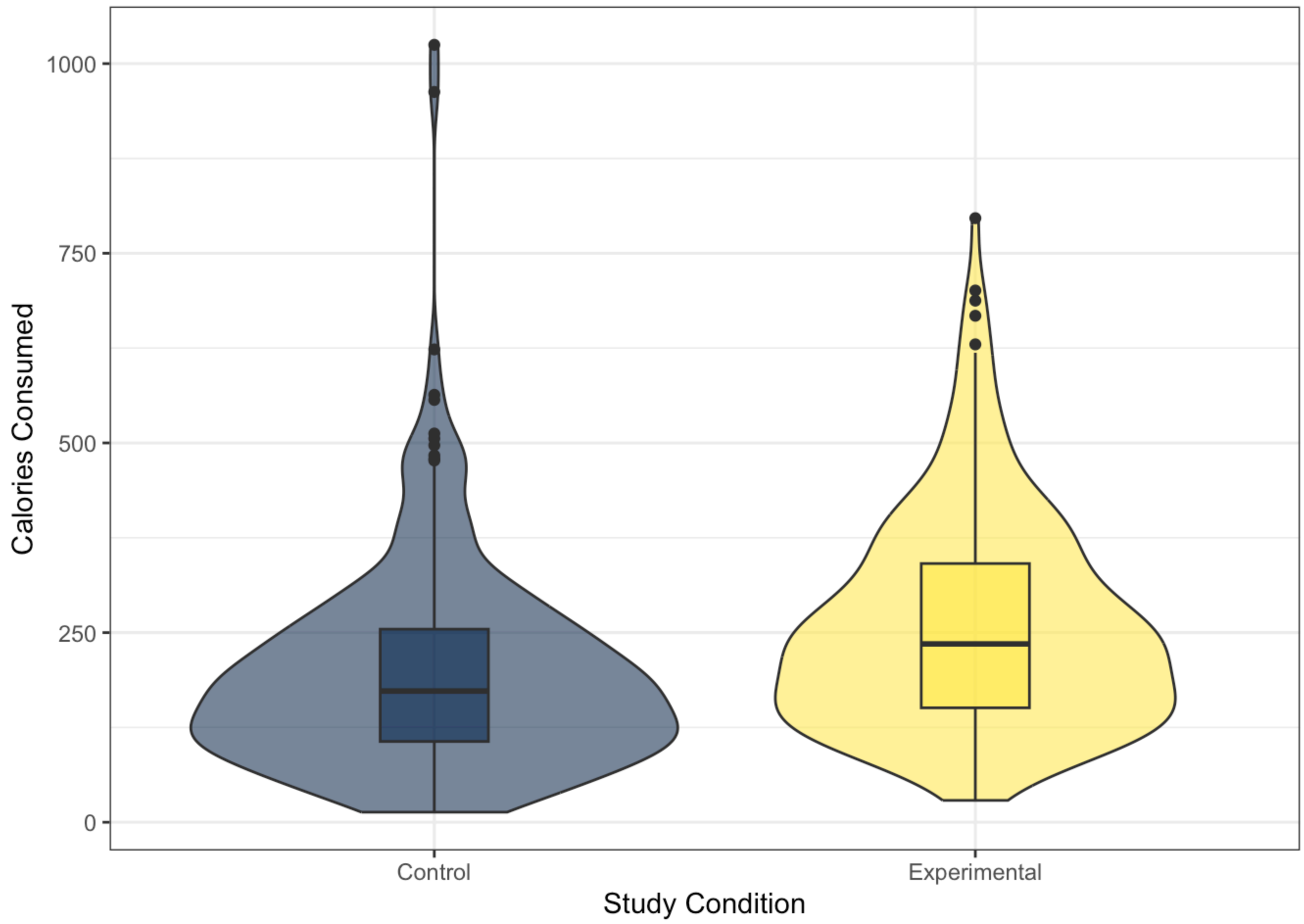
Tidy up a few things to make the data easier to interpret, and compute some descriptives for our final report.

```
lopez_clean <- lopez_data %>%
  mutate(Condition = as.factor(Condition),
         Condition_label = case_match(Condition,
                                      "0" ~ "Control",
                                      "1" ~ "Experimental"))


lopez_descriptives <- lopez_clean %>%
  group_by(Condition_label) %>%
  summarize(mean_calories = mean(F_CaloriesConsumed),
            sd_calories = sd(F_CaloriesConsumed),
            n = n())
```

Visualize differences between groups using something like a boxplot. Violin-boxplots, as shown here are more polished looking.

```
# call ggplot, specify data, set x, y, and fillcolor axes
ggplot(data = lopez_clean, aes(x=Condition_label, y=F_CaloriesConsumed, fill=Condit
  # add violin and boxplots
  geom_violin() +
  geom_boxplot(width=0.2) +
  # specify colors and axis labels; drop the color legend since it's redundant with
  scale_fill_viridis_d(option = "E",
                       alpha = 0.6) +
  scale_x_discrete(name="Study Condition") +
  scale_y_continuous(name="Calories Consumed") +
  guides(fill="none") +
  # specify a theme
  theme_bw()
```



The mean of the Experimental group is indeed higher than the mean of the Control group, but there's a lot of overlap in their distributions.

8.1.2 Using `t.test()` to assess the difference between groups

In a *t*-test, we express the difference in some outcome between two groups using a kind of standardized mean difference. That is, we take the difference between the two groups and divide by the standard error of the difference. The "classical" version of the *t*-test is Student's *t*-test, which assumes the variance is approximately equal in each group, and is easier to compute by hand. In this day and age when we have software doing this, you almost always will want Welch's *t*-test, which allows the groups to have different variances, and adjusts the degrees of freedom accordingly. By default, `t.test()` uses the Welch's test.

The function `t.test()` requires two inputs:

- A formula where you specify the outcome/dependent variable and the predictor/independent variable in the form `outcome ~ predictor` .
- The data you want to use.

Conducting a Welch's *t*-test then looks like this:

```
t.test(formula = F_CaloriesConsumed ~ Condition_label,
       data = lopez_clean)
```

```
##
##  Welch Two Sample t-test
##
## data:  F_CaloriesConsumed by Condition_label
## t = -4.8578, df = 453.45, p-value = 1.638e-06
## alternative hypothesis: true difference in means between group Control and group
## 95 percent confidence interval:
##  -88.55610 -37.54289
## sample estimates:
##      mean in group Control mean in group Experimental
##                   196.6818                   259.7313
```

8.1.3 Interpreting the output

This output gives the three key concepts of inferential statistics.

8.1.3.1 Estimating effect sizes

The `sample estimates` line at the bottom of the output gives the mean for each group, which is our best estimate of the population means $\hat{\mu}$. The mean calories consumed by the control group is 196.68 calories; the mean consumed by the Experimental group is 259.73 calories.

Somewhat annoyingly, we do not directly get the mean difference between groups as a raw/unstandardised mean difference. We must manually calculate it by subtracting the means of each group (196.6818 - 259.7313 = -63.05). So, those in the experimental group ate on average

63 more calories of soup than the control group, and this is our best estimate of the population mean difference between eating soup in these two conditions.

8.1.3.2 Confidence interval

`t.test()` gives the 95% confidence interval for the true difference between the group means: [-88.56, -37.54]. (Because the CI_{95} does not include zero, we know that this result is statistically significant at the $\alpha = .05$ level.)

8.1.3.3 Hypothesis testing

What does `p-value = 1.638e-06` mean? Remember R reports very small or very large numbers using scientific notation to save space. We normally report *p*-values to three decimals, so we report anything smaller as *p* < .001 to say it is smaller than this.

If you want to see the real number, you can use the following function:

```
format(1.638e-06, scientific = FALSE)
```

```
## [1] "0.000001638"
```

Regardless, this p value is comfortable below $\alpha = .05$, so we can say the result is statistically significant, and would report it as $p < .001$.

8.1.3.4 Statistical values

The top lines of the output give the *t* value and the degrees of freedom *(df)*. A few notes:

- The *t* value and the confidence interval are both negative in the statistical output, but what matters is the absolute value. If we had set our conditions up the other way, with Experimental first and then Control, we would get positive numbers. What's important is to make sure that whichever way you write about your results (Experimental is larger than control, or Control is smaller than Experimental) aligns with the sign of the numbers you report (the difference is positive, or the difference is negative).
- The df is not $N_{\text{observations}} - 2$, but has been adjusted to account for the differing variances between the two groups.

8.1.4 Reporting results

Here’s one way you could do it.

> Participants in the Experimental group ate more soup (198.68 calories, sd = 138.33 calories) than participants in the Control group (259.73 calories, sd = 140.59 calories). This difference was statistically significant (Welch’s *t*(453.45) = 4.86, *p* < .001); on average, Experimental participants ate on average 63.05 (95% CI = [37.54, 88.56]) more calories than Control participants.

8.1.5 One-sample and paired tests

In the example above, we compared means from two different groups. However, you may often want to compare a single group against a fixed value. Here, we filter only the Experimental participants from the Lopez data, and ask whether their calories consumed are significantly different from a theorized mean value of 200.

```r
lopez_experimental <- lopez_clean %>%
  filter(Condition == 1) %>%
  droplevels()

# specify the variable to use using the $ operator
t.test(x=lopez_experimental$F_CaloriesConsumed,
       mu = 200)
```

```
## 
##  One Sample t-test
## 
## data:  lopez_experimental$F_CaloriesConsumed
## t = 6.2729, df = 217, p-value = 1.894e-09
## alternative hypothesis: true mean is not equal to 200
## 95 percent confidence interval:
##  240.9636 278.4990
## sample estimates:
## mean of x 
##  259.7313
```

Other times, we might want to compare two measurements taken under different conditions but from the same participants. In the Lopez data, each participant estimated their soup calories consumed, and we can ask whether that estimate is different from the actual calories consumed.

Our data are already in *wide* format, with the two measurements in the same row but different columns; if your data are in *long* format, you will first need to wrangle the data into the correct shape using `pivot_wider()` .

```r
# specify both data vectors explicitly
t.test(x = lopez_clean$F_CaloriesConsumed,
       y = lopez_clean$CalEstimate,
       paired = TRUE)
```

```
## 
##  Paired t-test
## 
## data:  lopez_clean$F_CaloriesConsumed and lopez_clean$CalEstimate
## t = 14.589, df = 460, p-value < 2.2e-16
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
##   88.18327 115.63870
## sample estimates:
## mean difference 
##         101.911
```

8.2 Comparing three or more groups using one-way ANOVA

To practice, we're going to use data from experiment 2 of James et al (2015).[29] They were interested in whether you can reduce intrusive memories associated with a traumatic event. Participants watched a video designed to be traumatic, and then were randomly allocated to one of four groups that engaged in different followup activities:

1. Control
2. Reactivation + Tetris
3. Tetris only
4. Reactivation only

They measured the number of intrusive memories prior to the start of the study, then participants kept a diary to record intrusive memories about the film in the 7 days after watching it. The authors were interested in whether the combination of reactivation and playing Tetris would lead to the largest reduction in intrusive memories. We will recreate their analyses using a one-way ANOVA.

8.2.1 Initial exploration and descriptive statistics

Read in the data; add a participant ID variable; convert Condition to a factor; rename the variable with count of intrusive memories; select only the variables we need for our analysis.

```
datafile <- here("datasets/james/James_2015.csv")
james_data <- read.csv(datafile)

james_clean <- james_data %>%
  mutate(participant_id = row_number(),
         Condition = as.factor(Condition),
         Condition = fct_recode(Condition,        # Rename Condition levels using "
                                "Control"                = "1",
                                "Reactivation + Tetris" = "2",
                                "Tetris only"            = "3",
                                "Reactivation only"      = "4"),
         Intrusions = Days_One_to_Seven_Image_Based_Intrusions_in_Intrusion_Diary)
  select(participant_id,
         Condition,
         Intrusions)

glimpse(james_clean)
```

```
## Rows: 72
## Columns: 3
## $ participant_id <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, …
## $ Condition      <fct> Control, Control, Control, Control, Control, Control, C…
## $ Intrusions     <int> 4, 3, 6, 2, 3, 4, 0, 4, 2, 11, 16, 12, 2, 7, 7, 6, 2, 1…
```

Next, we want to calculate some descriptive statistics to see some overall trends in the data. We'll want these grouped by Condition, so we can see them for each experimental group.

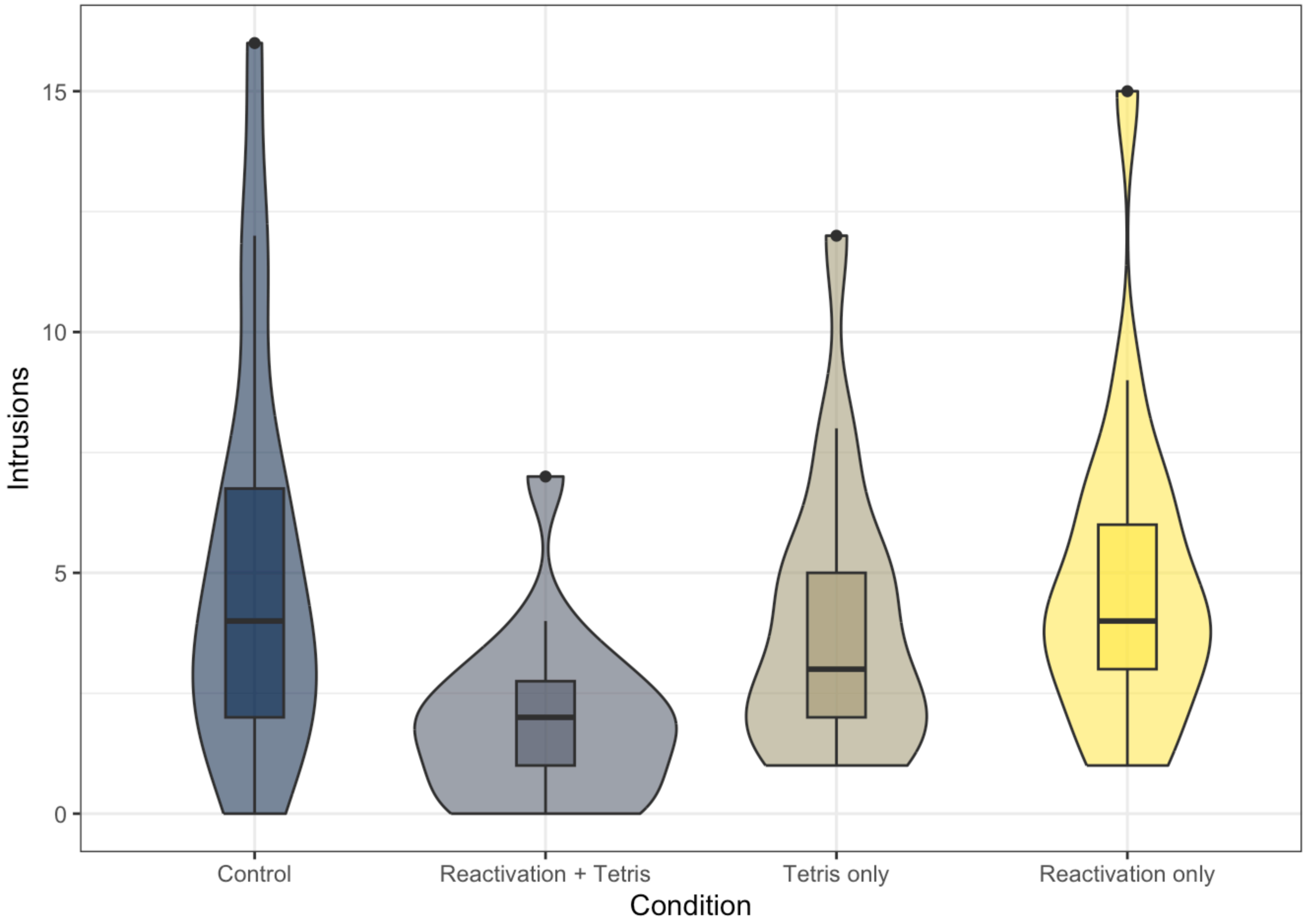
```
james_descriptives <- james_clean %>%
  group_by(Condition) %>%
  summarize(mean_intrusions = mean(Intrusions),
            sd_intrusions = sd(Intrusions),
            se_intrusions = sd(Intrusions)/sqrt(n()),
            n = n())

james_descriptives
```

```
## # A tibble: 4 × 5
##   Condition             mean_intrusions sd_intrusions se_intrusions     n
##   <fct>                           <dbl>         <dbl>         <dbl> <int>
## 1 Control                          5.11          4.23         0.996    18
## 2 Reactivation + Tetris            1.89          1.75         0.411    18
## 3 Tetris only                      3.89          2.89         0.681    18
## 4 Reactivation only                4.83          3.33         0.785    18
```

Visualize the data.

```
ggplot(data = james_clean, (aes(x = Condition, y = Intrusions, fill = Condition)))
  geom_violin() +
  geom_boxplot(width = 0.2) +
  # make things look nice
  scale_fill_viridis_d(option = "E",
                       alpha = 0.6) +
  guides(fill="none") +
  theme_bw()
```



8.2.2 Using `aov_ez()` to compare groups

We can run the one-way ANOVA using `aov_ez()` (from the `afex` package), and save it to the object `mod`.

```
mod <- aov_ez(data = james_clean,

              # specify columns
              id = "participant_id", # column containing subject IDs
              dv = "Intrusions",     # column containing outcome measurements
              between = "Condition", # column containing between-subjects predictor

              # specify details of how to run the test
              es = "pes",            # set the effect size to partial eta squared
              type = 3,              # this affects how the sum of squares is calcu
              include_AOV = TRUE
              )
```

```
## Contrasts set to contr.sum for the following variables: Condition
```

```
# pipe the output to tidy() for readability
mod$anova_table %>% tidy()
```

```
## Warning: The column names num.Df, den.Df, MSE, and ges in
## ANOVA output were not recognized or transformed.
```

```
## # A tibble: 1 × 7
##   term      num.Df den.Df   MSE statistic   ges p.value
##   <chr>      <dbl>  <dbl> <dbl>     <dbl> <dbl>   <dbl>
## 1 Condition      3     68  10.1      3.79 0.143  0.0141
```

Note: `aov_ez()` may produce some messages that look like errors, do not worry about these, they are just letting you know what it's done.

8.2.3 Interpreting the output

In an ANOVA, the initial output only tells you the results of the hypothesis test. The null hypothesis (H_0) is "All the groups have the same mean value." What is the alternative hypothesis (H_1)?

8.2.3.1 Hypothesis testing

The *p*-value is 0.014, so we can say that this result is statistically significant.

8.2.3.2 Statistical values

For the predictor variable `Condition`, this output gives the numerator degrees of freedom (3), the denominator degrees of freedom (68), and the F value (3.79). It also gives the mean square error associated with this predictor (10.08), and the proportion of variance explained by this predictor (0.14).

8.2.4 Checking assumptions

As our statistical models become more sophisticated, it becomes more important to check that the requirements are true. For one-way between-subjects ANOVA, we need two design features:

1. DV that is interval or ratio (yes, because our DV is a count of events).
2. Observations that are independent (yes, each observation comes from a different participant).

We also need to check two characteristics of the data's distribution:

3. The residuals (the leftover variance after the model's best guess) should be normally distributed.
4. There should be homogeneity of variance among the groups.

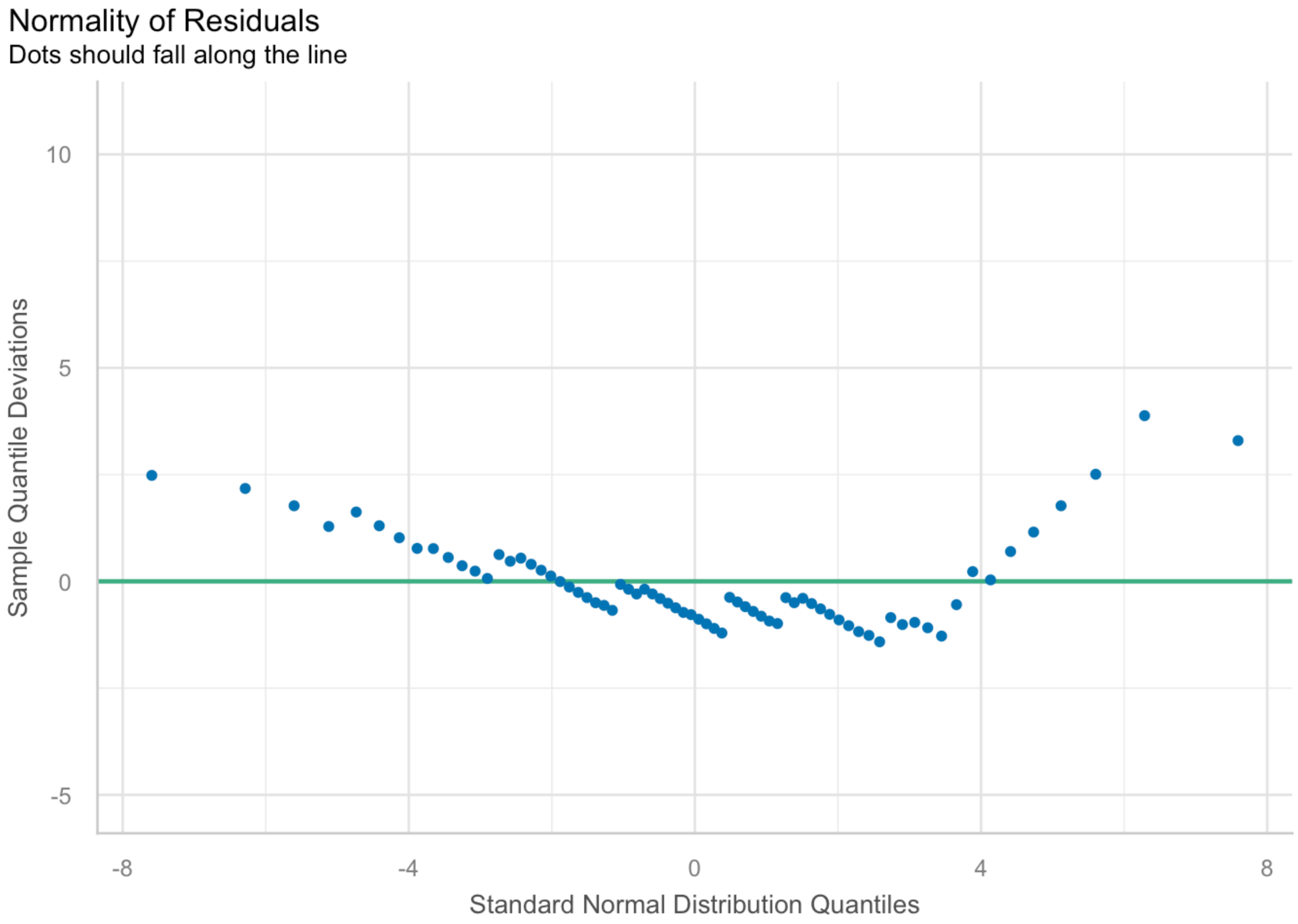
`aov_ez()` made diagnostic plots that can be used to check these.

8.2.4.1 Normality of residuals

A quantile-quantile (qq) plot of the residuals tells us how close they fall to a true normal distribution. (`check_normality()` comes from the `performance` package.)

```
# to extract a qq plot from aov_ez() output
plot(check_normality(mod))
```

```
## For confidence bands, please install `qqplotr`.
```



Hm. The assumption of normality might not be ideal. Is this a problem? If the sample sizes for each group are equal, then ANOVA is robust to violations of both normality and of homogeneity of variance.[30] We can refer back to our descriptives to check how many participants are in each condition.

```
james_descriptives$n
```

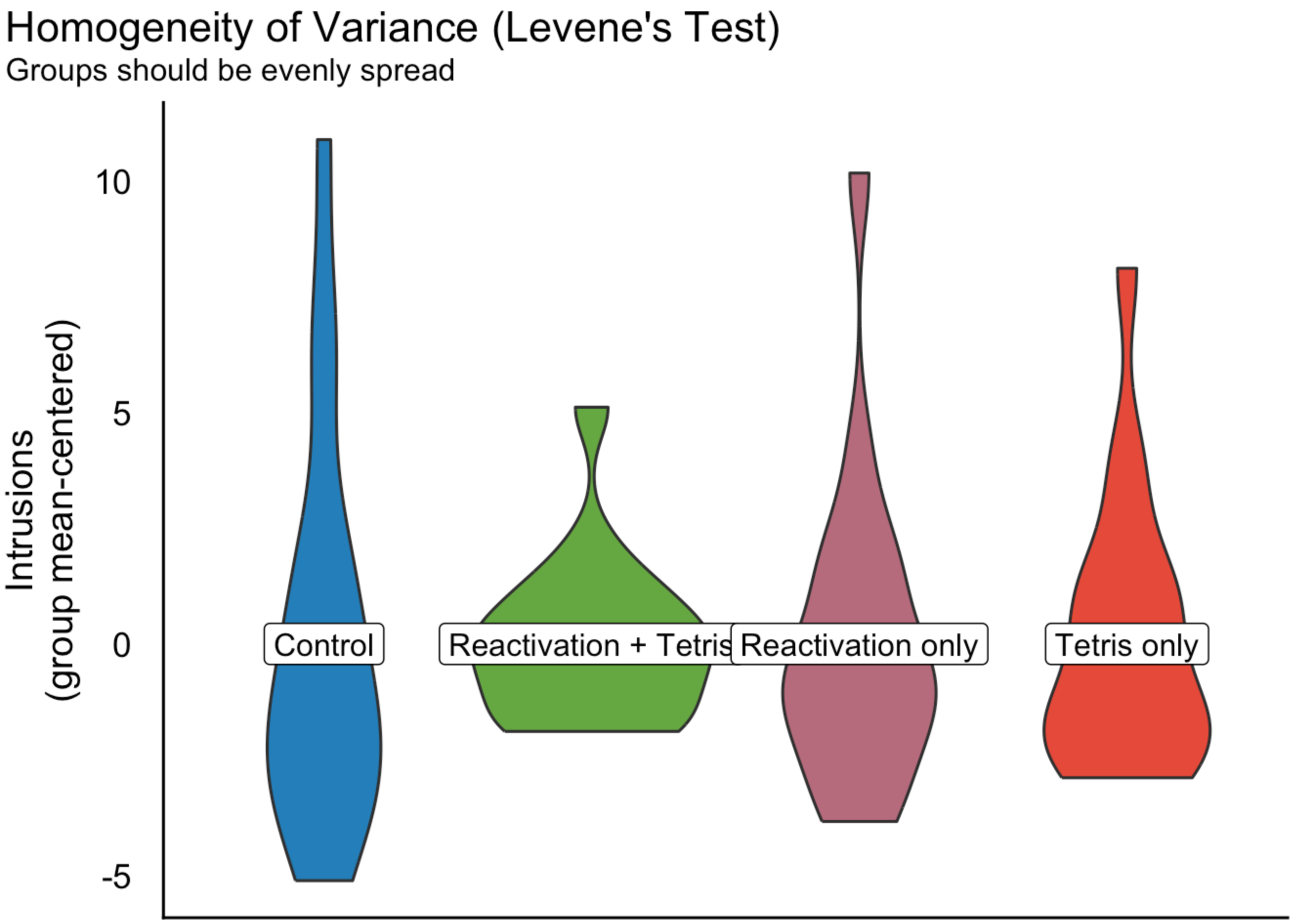
```
## [1] 18 18 18 18
```

Thankfully, the sample sizes are equal, so we should be OK to proceed with the ANOVA. It is not clear whether normality was checked in the original paper.

8.2.4.2 Homogeneity of variance

We can plot mean-centered violin plots of each condition to see how similar their variance is.

```
plot(check_homogeneity(mod))
```



Again, there is some inconsistency in the variance (and looking at our initial visualization, we can see that this is in part that the Reactivation + Tetris group has a strong floor effect at Intrusions = 0).

James et al. (2015) suspect there might be issues with this assumption as they mention that the ANOVAs do not assume equal variance, however, the results of the ANOVA that are reported are identical to our results above where no correction has been made (although their post-hoc tests are Welch t-tests, which account for unequal variance).

8.2.4.3 What is the point of assumption testing if we're just going to ignore it?

While all of this might seem very confusing, we are showing you this for three reasons.

1. To reassure you that sometimes the data can fail to meet the assumptions and it is still ok to use the test. To put this in statistical terms, many tests are **robust** to mild deviations of normality and unequal variance, particularly with equal sample sizes.
2. As a critical thinking point, to remind you that just because a piece of research has been published does not mean it is perfect and you should always evaluate whether the methods used are appropriate.
3. To reinforce the importance of pre-registration where these decisions could be made in advance, and/or open data and code so that analyses can be reproduced exactly to avoid any ambiguity about exactly what was done. In this example, given the equal sample sizes and the difference in variance between the groups isn't too extreme, it looks like it is still appropriate to use an ANOVA but the decisions and justification for those decisions could have been more transparent.

8.2.5 Post-hoc testing

The ANOVA results above tell us that there is some difference among the four conditions, but not exactly which conditions are significantly different from one another. We need to explicitly test pairs of conditions to understand this. You could step through using `t.test()` for each pair of conditions, but a quicker and better way is to use `emmeans()` which computes all possible pairwise comparison t-tests and applies a multiple-comparisons correction to the *p*-values.

```
mod_pairwise <- emmeans(mod,
                        pairwise ~ "Condition",
                        adjust = "bonferroni")

mod_pairwise$contrasts
```

```
##  contrast                                    estimate   SE df t.ratio p.value
##  Control - (Reactivation + Tetris)              3.222 1.06 68   3.044  0.0199
##  Control - Tetris only                          1.222 1.06 68   1.155  1.0000
##  Control - Reactivation only                    0.278 1.06 68   0.262  1.0000
##  (Reactivation + Tetris) - Tetris only         -2.000 1.06 68  -1.889  0.3787
##  (Reactivation + Tetris) - Reactivation only   -2.944 1.06 68  -2.781  0.0420
##  Tetris only - Reactivation only               -0.944 1.06 68  -0.892  1.0000
##
## P value adjustment: bonferroni method for 6 tests
```

This output shows the estimated mean difference between each pair of conditions, the standard error of the estimate (identical across all pairs, because it’s a characteristic of the model), the modeled degrees of freedom, *t*-statistic, and resulting **adjusted** *p*-value.

Which pairs of conditions are significantly different from one another after adjusting for multiple comparisons? Why is that adjustment important?

8.2.6 Reporting results

Because there are 4 conditions, I probably would not report descriptives for each level directly in the text, but would refer readers to the figure. (I’ve also used negative values for both post-hoc t-tests, because I’m describing one condition being lower than another.)

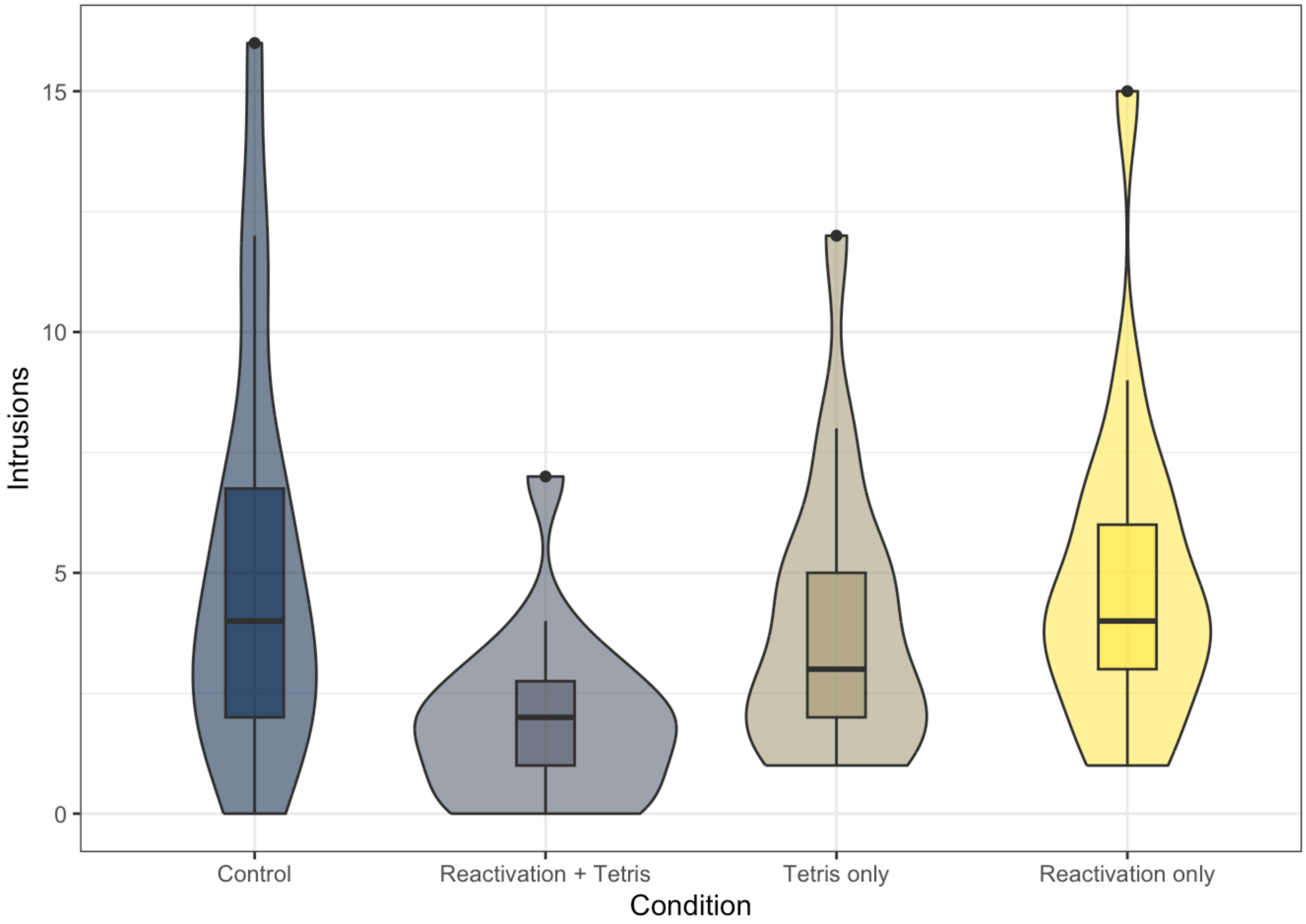


Figure 8.1: Violin plots showing the number of intrusive memories by participants in each experimental condition. Intrusive memories were most frequent for the Control group, and least frequent for the Reactivation + Tetris group.

I might write:

> Although intrusive memories were reported by participants in all groups (Figure 8.1), the number of intrusions was significantly different across experimental conditions (F(3,68) = 3.79, *p* = .014, partial η^2 = .14). Post-hoc *t*-tests using Bonferroni correction for six comparisons revealed that the Reactivation + Tetris group had significantly fewer intrusions than either the Control group (*t*(68) = -3.04, adjusted *p* = .020) or the Reactivation only group (*t*(68) = -2.78, adjusted *p* = .042). No other condition differences were significant.

8.3 Comparing means in factorial designs using factorial ANOVA.

Previously, we saw how to conduct and interpret a one-way ANOVA in R. These are flexible models where you have one independent variable (IV) with three or more levels. They get you a long way but sometimes your research question and design calls for multiple IVs or factors.

In this section, we extend the ANOVA framework to include two or more IVs/factors and their interaction. We will show you how to run a factorial ANOVA using the **afex** package and break down interactions through post-hoc tests using the **emmeans** package. Visualising your data is particularly important for understanding interactions, so we put a lot of emphasis on data visualisation through violin-boxplots and interaction plots.

We will use open data from Zhang et al. (2014).[31] These researchers were interested in whether people could predict how interested they would be in rediscovering past experiences. They call it a “time capsule” effect, where people store photos or messages to remind themselves of past events in the future. They predicted participants in the ordinary group would underestimate their future feelings (i.e., there would be a bigger difference between time 1 and time 2 measures) compared to participants in the extraordinary group.

We can describe this experiment as a 2 x 2 mixed design. The first IV is time (time 1, time 2) and is within-subjects. The second IV is type of event (ordinary, extraordinary) and is between-subjects. Our DV is the rated interest, enjoyment, and meaningfulness (combined into one composite variable) that participants predicted (at time 1) and experienced (at time 2).

A design and analysis like this lets us ask three different research questions at the same time.

1. Does interest differ between time 1 and time 2? (Is there a *main effect of time*?)
2. Does interest differ between ordinary and extraordinary events? (Is there a *main effect of type of event*?)
3. Is the change in interest from time 1 to time 2 different for ordinary events than for extraordinary events? (Is there an *interaction between time and type of event*?)

8.3.1 Initial exploration and descriptives

Read in the data; select and rename key variables; add participant ID; recode type of event. Pivot to long format for `aov_ez()` .

```
datafile <- here("datasets/zhang/Zhang_2014.csv")
zhang_data <- read.csv(datafile)

zhang_wide <- zhang_data %>%

  select(
    Condition,
    time1_interest = T1_Predicted_Interest_Composite,
    time2_interest = T2_Actual_Interest_Composite) %>%

  rename(condition = Condition) %>%

  mutate(
    participant_id = row_number(),
    condition = as.factor(case_match(condition,
                           1 ~ "Ordinary",
                           2 ~ "Extraordinary")))

zhang_long <- pivot_longer(zhang_wide,
                            cols=c(time1_interest,time2_interest),
                            names_to = "time",
                            values_to = "interest") %>%

  mutate(time = as.factor(time),
         time = fct_recode(time,
             "time1" = "time1_interest",
             "time2" = "time2_interest"))
```

Before we start on the inferential statistics, one key part of understanding your data and reporting for context in a report is calculating descriptive statistics like the mean and standard deviation.

```
zhang_descriptives <- zhang_long %>%
  group_by(time,condition) %>%
  summarize(mean_interest = mean(interest),
            sd_interest = sd(interest),
            n = n())
```

```
## `summarise()` has grouped output by 'time'. You can
## override using the `.groups` argument.
```

```
zhang_descriptives
```

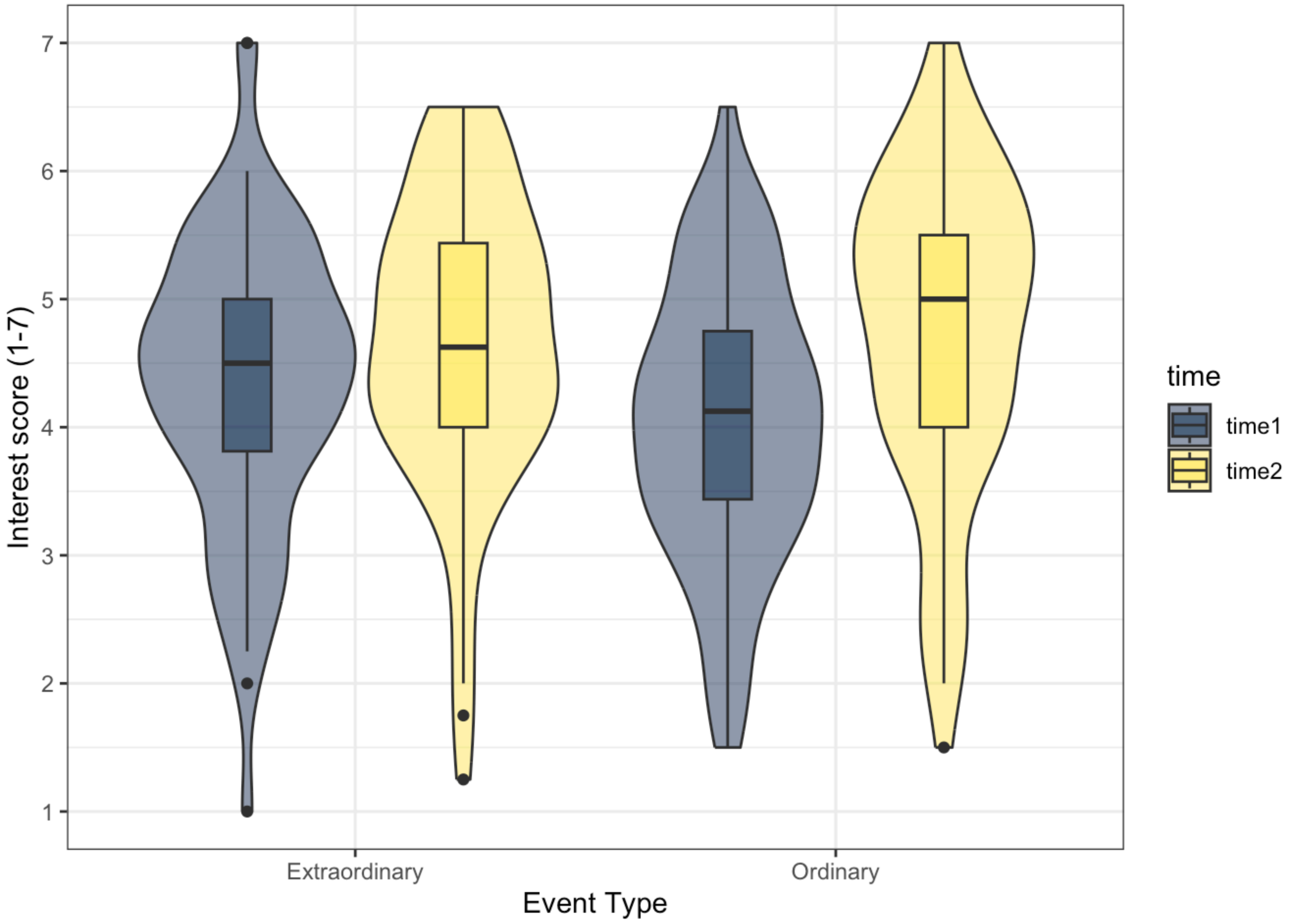
```
## # A tibble: 4 × 5
## # Groups:   time [2]
##   time  condition     mean_interest sd_interest     n
##   <fct> <fct>                 <dbl>       <dbl> <int>
## 1 time1 Extraordinary          4.36        1.13    66
## 2 time1 Ordinary               4.04        1.09    64
## 3 time2 Extraordinary          4.65        1.14    66
## 4 time2 Ordinary               4.73        1.24    64
```

It’s also important to visualize your data.

```
# We'll need to scoot some things sideways to make it all line up
dodge_value <- 0.9

ggplot(data = zhang_long, aes(x = condition, y = interest, fill = time)) +
  geom_violin() +
  geom_boxplot(width = 0.2,
               position = position_dodge(dodge_value)) +

  scale_fill_viridis_d(option = "E", alpha = 0.5) +
  scale_x_discrete(name = "Event Type") +
  scale_y_continuous(name = "Interest score (1-7)",
                     breaks = c(1:7)) +
  theme_bw()
```



Consistent with our descriptives calculated above, mean interest is lowest for ordinary events at time 1, and highest for ordinary events at time 2.

8.3.2 Using `aov_ez` to conduct a factorial ANOVA

We will again use `aov_ez()` from the `afex` package to run the ANOVA. Remember that we need to specify two IVs, one which is between-subjects and one which is within-subjects. This function needs our data in long format.

```
mod_factorial <- aov_ez (data = zhang_long,
                         id = "participant_id",
                         within = "time",
                         between = "condition",
                         dv = "interest",
                         type = 3,
                         es = "pes"
                         )
```

8.3.3 Interpreting the output

```
mod_factorial
```

```
## Anova Table (Type 3 tests)
##
## Response: interest
##           Effect     df  MSE         F  ges p.value
## 1      condition 1, 128 2.05      0.46 .003    .498
## 2           time 1, 128 0.61 25.88 *** .044   <.001
## 3 condition:time 1, 128 0.61    4.44 * .008    .037
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '+' 0.1 ' ' 1
```

This output is read just like the output from the one-way ANOVA, but our model now has three different terms (corresponding to our three research questions).

8.3.3.1 Hypothesis testing

We get three *p* values, corresponding to our three research questions. The main effect of condition is not significant, the main effect of time is significant at $\alpha = .001$, and the interaction between condition and time is significant at $\alpha = .05$.

8.3.3.2 Statistical values

Again, we get the df, MSE, F, partial η^2, and p-value for each of the main effects and the interaction.

8.3.4 Checking assumptions

Factorial ANOVA has the same assumptions as one-way ANOVA:

1. The DV is interval or ratio data.
2. The observations should be independent.
3. The residuals should be normally distributed.
4. There should be homogeneity of variance between the groups.

8.3.4.1 Data type

From the experiment design, the DV should be interval or ratio data, and the observations should be independent. This brings us to an interesting problem: can we treat this DV as interval or ratio? The Likert scale DV is ordinal, a type of data that are very common in psychology. The problem is that ordinal data are not interval or ratio data, there's a fixed number of integer values they can take (the values of the Likert scale) and you cannot claim that the distance between the values is equal (is the difference between strongly agree and agree the same as the difference between agree and neutral?).

Technically, we should not use an ANOVA to analyze ordinal data - *but almost everyone does*. Many people argue that if you take the average of multiple Likert scale items, you can interpret the data as if they are interval and they can be normally distributed. (The *interest* DV we are using here is exactly this, it's the average of three Likert ratings.) Other people argue you should use non-parametric methods or more complex models such as ordinal regression for this type of data, but that is out of scope for this course. Whichever route you choose, you should understand the data you have and you should be able to justify your decision.

8.3.4.2 Normality of residuals and homogeneity of variance

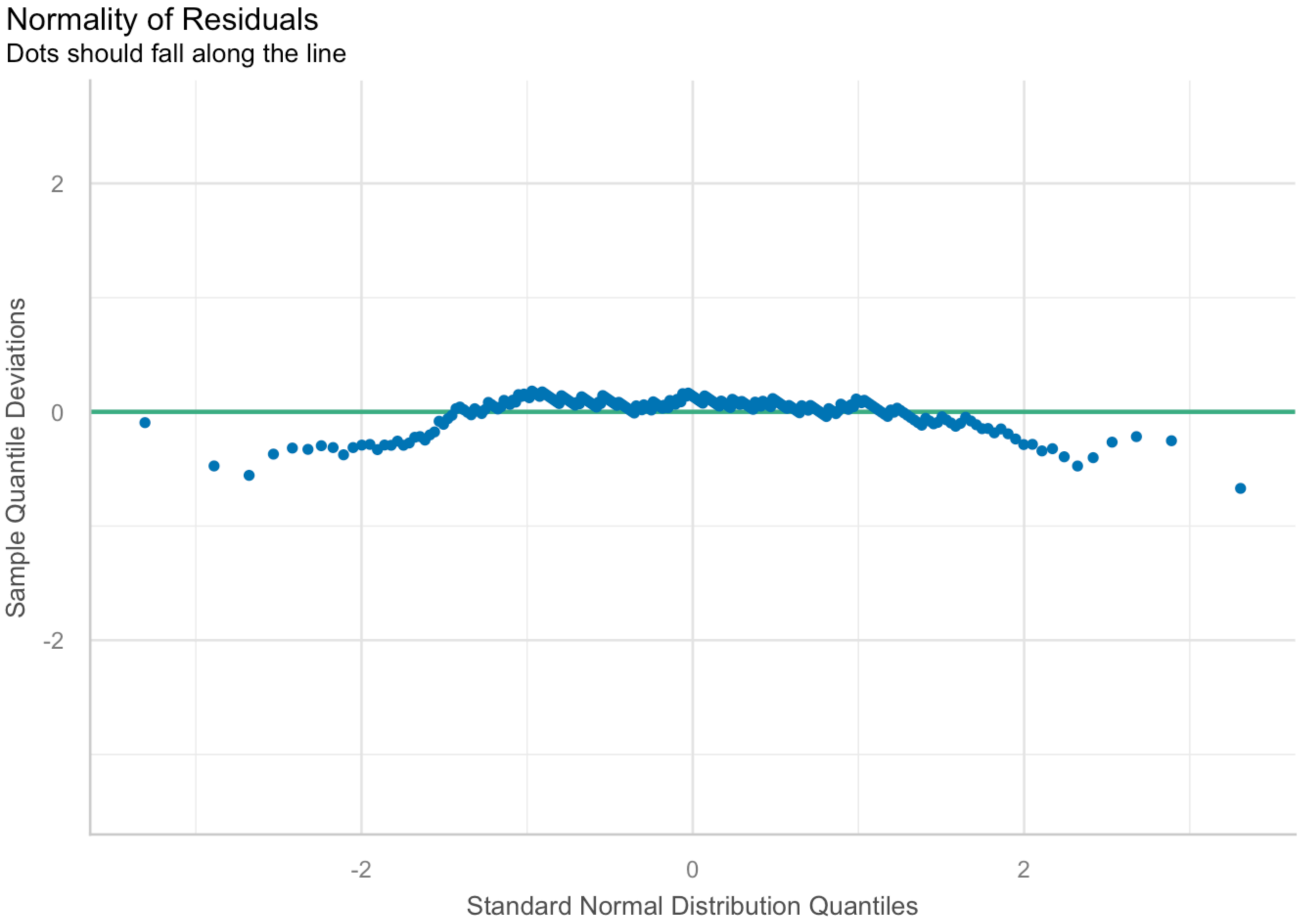
To check 3 and 4 we again use plots from the `performance` package. Let's also check our counts for each group to see if they're equal.

```
zhang_descriptives$n
```

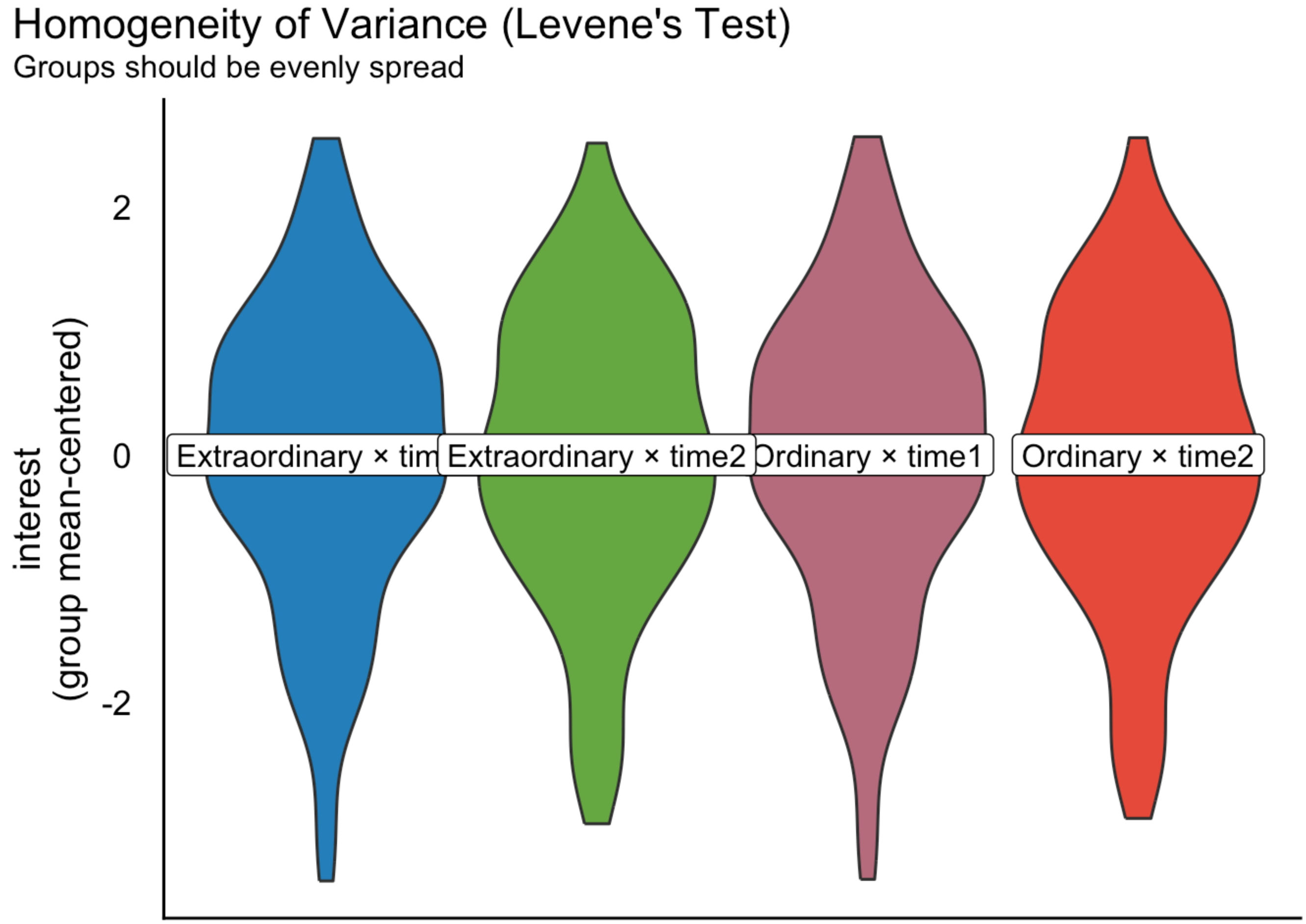
```
## [1] 66 64 66 64
```

```
plot(check_normality(mod_factorial))
```

```
## For confidence bands, please install `qqplotr`.
```



```
plot(check_homogeneity(mod_factorial))
```



This all looks pretty good. We have slightly more subjects in the Ordinary condition than the Extraordinary condition, but the residuals are close to the normal line, and the variance looks similar across all four groups.

8.3.5 Post-hoc comparisons

Our main effects (condition and time) each only have two levels, so we do not need to do any post hoc tests to determine which conditions differ from one another; if one of our factors had had three levels, however, we would use `emmeans()` to compute the contrasts for the main effect, as we did with the one-way ANOVA.

Because the interaction is significant, we should follow this up with post-hoc tests using `emmeans()` to determine which comparisons are significant. If the overall interaction is not significant, you should not conduct additional tests between individual cells of the design.

`emmeans()` requires you to specify the `aov` object, and then the factors you want to contrast. For an interaction, we use the notation `pairwise ~ IV1 | IV2` and you specify which multiple comparison correction you want to apply.

```
emmeans(mod_factorial,
        pairwise ~ time | condition,
        adjust = "bonferroni")
```

```
## $emmeans
## condition = Extraordinary:
##  time  emmean    SE  df lower.CL upper.CL
##  time1   4.36 0.137 128     4.09     4.63
##  time2   4.65 0.147 128     4.36     4.94
##
## condition = Ordinary:
##  time  emmean    SE  df lower.CL upper.CL
##  time1   4.04 0.139 128     3.76     4.31
##  time2   4.73 0.149 128     4.44     5.03
##
## Confidence level used: 0.95
##
## $contrasts
## condition = Extraordinary:
##  contrast      estimate    SE  df t.ratio p.value
##  time1 - time2   -0.288 0.136 128  -2.123  0.0357
##
## condition = Ordinary:
##  contrast      estimate    SE  df t.ratio p.value
##  time1 - time2   -0.695 0.138 128  -5.049  <.0001
```

The code above asks R to tell us the difference between time1 and time2 at each level of condition. We could swap the order of the two factors to instead see the difference between ordinary and extraordinary at each time:

```
emmeans(mod_factorial,
        pairwise ~ condition | time,
        adjust = "bonferroni")
```

```
## $emmeans
## time = time1:
##  condition     emmean    SE  df lower.CL upper.CL
##  Extraordinary   4.36 0.137 128     4.09     4.63
##  Ordinary        4.04 0.139 128     3.76     4.31
##
## time = time2:
##  condition     emmean    SE  df lower.CL upper.CL
##  Extraordinary   4.65 0.147 128     4.36     4.94
##  Ordinary        4.73 0.149 128     4.44     5.03
##
## Confidence level used: 0.95
##
## $contrasts
## time = time1:
##  contrast                 estimate    SE  df t.ratio p.value
##  Extraordinary - Ordinary   0.3246 0.195 128   1.661  0.0992
##
## time = time2:
##  contrast                 estimate    SE  df t.ratio p.value
##  Extraordinary - Ordinary  -0.0829 0.209 128  -0.397  0.6923
```

Look at how the contrasts are expressed subtly different when you switch the order. Think carefully about your research question and hypotheses for which way around is the most informative.

8.3.6 Creating an interaction plot

When you have a factorial design, one powerful way of visualising the data is through an interaction plot. This is essentially a line graph where the x-axis has one IV and separate lines for a second IV. However, once you have the factorial ANOVA model, you can add confidence intervals to the plot to visualise uncertainty. **afex** has it's own function called `afex_plot()` which you can use with the model object you created.

In the code below, there are a few key argument to highlight:

- `object` is the **afex** model you created.
- `x` is the variable you want on the x-axis.
- `trace` is the variable you want to plot as separate lines.
- `error` controls whether the error bars show confidence intervals for between-subjects or within-subjects. In a mixed design, these have different properties, so you must think about which you want to plot and highlight to the reader.
- `factor_levels` lets you edit the levels of factors you plot, such as renaming or reordering them. You add each factor into a list but check the documentation and vignettes for other options.

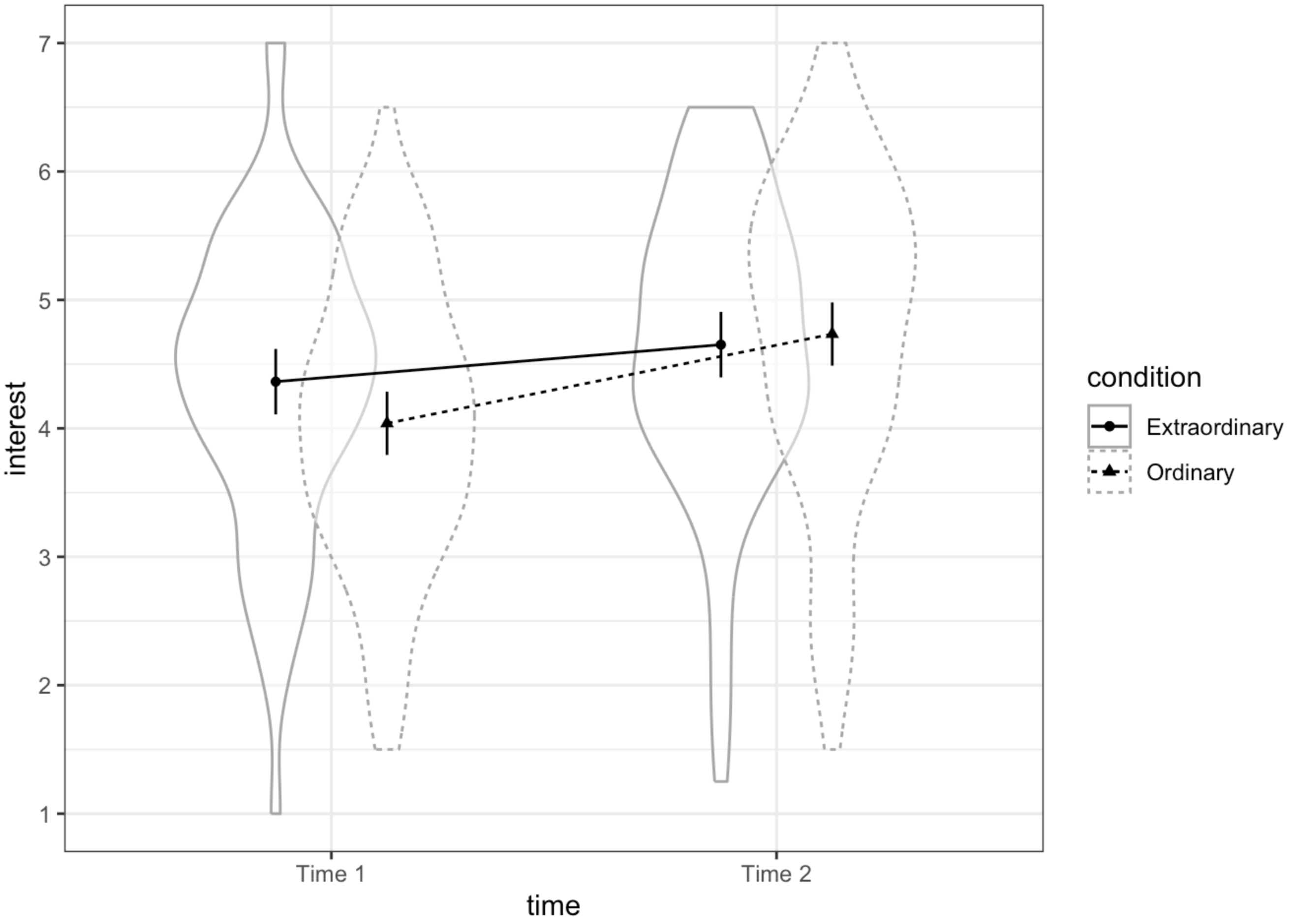
One handy feature about this function is it uses **ggplot2** in the background, so you can add layers to the initial function like other plots that we have created.

```
afex_plot(object = mod_factorial,
          x = "time",
          trace = "condition",
          error = "between",
          factor_levels = list(time = c("Time 1","Time 2")),
          data_geom = geom_violin) +

  scale_y_continuous(breaks = 1:7) +
  theme_bw()
```

```
## Renaming/reordering factor levels of 'time':
##   time1 -> Time 1
##   time2 -> Time 2
```

```
## Warning: Panel(s) show a mixed within-between-design.
## Error bars do not allow comparisons across all means.
## Suppress error bars with: error = "none"
```



8.3.7 Reporting the results

```
## Renaming/reordering factor levels of 'time':
##   time1 -> Time 1
##   time2 -> Time 2
```

```
## Warning: Panel(s) show a mixed within-between-design.
## Error bars do not allow comparisons across all means.
## Suppress error bars with: error = "none"
```

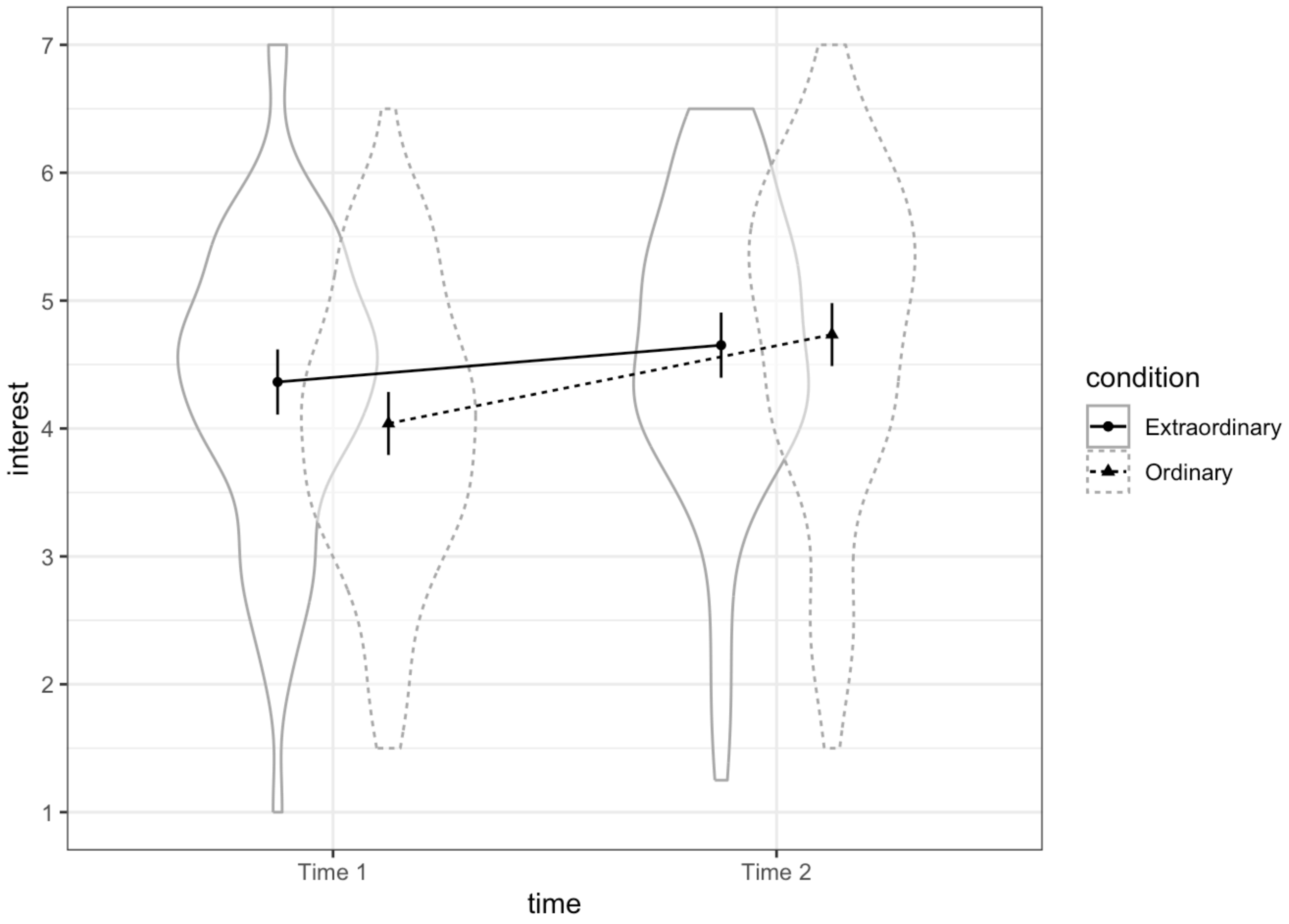


Figure 8.2: Predicted (Time 1) and actual (Time 2) interest ratings for ordinary events (dashed lines) and extraordinary events (solid lines). Error bars are between-subject standard error of the mean.

You could write something like:

> Across all conditions, participants rated their interest in the time capsule materials as mildly positive (Figure 8.2) . A factorial ANOVA with factors Condition (ordinary, extraordinary) and Time (Time 1, Time 2) revealed a significant main effect of time (*F*(1,128) = 25.88, *p* < .001, partial η^2 = .044), with ratings higher at Time 2 (4.69, SD = 1.19) than at Time 1 (4.20, sd = 1.12). The increase from Time 1 to Time 2 was larger for ordinary events (mean increase 0.70, SD = 1.13) than for extraordinary events (0.29, SD = 1.11), and this interaction was significant (*F*(1,128) = 4.44, *p* = .037, partial η^2 = .008). There was no significant effect of condition (*F*(1,128) = 0.46, *p* = .50, partial η^2 = .003)

26. https://psyteachr.github.io/quant-fun-v3/↩

27. Lopez, A., Choi, A. K., Dellawar, N. C., Cullen, B. C., Avila Contreras, S., Rosenfeld, D. L., & Tomiyama, A. J. (2024). Visual cues and food intake: A preregistered replication of Wansink et al. (2005). *Journal of experimental psychology. General*, *153*(2), 275–281. https://doi.org/10.1037/xge0001503↩

28. For a high-level overview, see this news article from 2018: https://www.npr.org/sections/thesalt/2018/09/26/651849441/cornell-food-researchers-downfall-raises-larger-questions-for-science↩

29. James, E. L., Bonsall, M. B., Hoppitt, L., Tunbridge, E. M., Geddes, J. R., Milton, A. L., & Holmes, E. A. (2015). Computer Game Play Reduces Intrusive Memories of Experimental Trauma via Reconsolidation-Update Mechanisms: *Psychological Science*, *26*(8), 1201–1215. https://doi.org/10.1177/0956797615583071↩

30. If you are interested, there is a good discussion of these issues in (1) Blanca, M. J., Alarcón, R., Arnau, J., Bono, R., & Bendayan, R. (2018). Effect of variance ratio on ANOVA robustness: Might 1.5 be the limit? *Behavior Research Methods*, *50*(3), 937–962. https://doi.org/10.3758/s13428-017-0918-2 and (2) Knief, U., & Forstmeier, W. (2021). Violating the normality assumption may be the lesser of two evils. *Behavior Research Methods*, *53*(6), 2576–2590. https://doi.org/10.3758/s13428-021-01587-5↩

31. Zhang, T., Kim, T., Brooks, A. W., Gino, F., & Norton, M. I. (2014). A “Present” for the Future: The Unexpected Value of Rediscovery. *Psychological Science*, *25*(10), 1851–1860. https://doi.org/10.1177/0956797614542274↩