Experiments

Claudius Gräbner-Radkowitsch

2025-06-13

Table of contents

1 Introduction

In this lab we will learn how to analyze data obtained from experiments. We will complement the lecture by also introducing some additiona, practically relevant concepts.

More precisely, we focus on the following aspects:

- Import and explore datasets as typically produced by experiments
- Conduct t-tests for simple experimental comparisons
- Perform ANOVA for multi-group comparisons
- Analyze factorial experimental designs
- Calculate and interpret effect sizes
- Create professional visualizations of experimental results
- Understand how ANOVA is a special case of linear regression

Throuout the tutorial we will use the following packages:

```r
library(dplyr)        # Data manipulation
```

```

Attaching package: 'dplyr'


The following objects are masked from 'package:stats':

    filter, lag


The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```r
library(ggplot2)      # Data visualization
library(ggdist)       # More visualization options
library(readr)        # Simple data import
library(broom)        # Extract model data
library(effectsize)   # Effect size calculations
library(car)          # Advanced ANOVA functions
```

```
Loading required package: carData


Attaching package: 'car'


The following object is masked from 'package:dplyr':

    recode
```

```r
library(emmeans)      # Post-hoc comparisons
```

```
Welcome to emmeans.
Caution: You lose important information if you filter this package's results.
See '? untidy'
```

```r
library(knitr)        # For nice tables
library(kableExtra)   # For enhanced table formatting
```

```
Attaching package: 'kableExtra'

The following object is masked from 'package:dplyr':

    group_rows
```

```r
library(patchwork)    # For aligning multiple plots
library(pwr)          # For power analysis and sample size planning
```

We will use the following data sets, which are available for download from the lab webpage.

```r
leadership_study_between <- read_csv("leadership_study_between.csv")
```

```
Rows: 60 Columns: 3
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (1): group
dbl (2): participant_id, team_performance

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
leadership_study_within <- read_csv("leadership_study_within.csv")
```

```
Rows: 30 Columns: 5
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (1): group
dbl (4): participant_id, team_performance, pre_performance, post_performance

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
communication_study <- read_csv("communication_study.csv")
```

```
Rows: 90 Columns: 4
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (1): communication_method
dbl (3): participant_id, satisfaction_score, task_completion_time

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
factorial_study <- read_csv("factorial_study.csv")
```

```
Rows: 120 Columns: 4
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (2): feedback_type, experience_level
dbl (2): participant_id, performance_improvement

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

2 Data Exploration

As usual, it is a good idea to start with looking at the data sets, such that you know what the data looks like:[1]

```r
head(leadership_study_between) |>
  kable()
```

participant_id	group	team_performance
1	control	69.39524
2	control	72.69823
3	control	90.58708
4	control	75.70508
5	control	76.29288
6	control	92.15065

```r
summary(leadership_study_between) |>
  kable()
```

participant_id	group	team_performance
Min. : 1.00	Length:60	Min. : 55.33
1st Qu.:15.75	Class :character	1st Qu.: 70.71
Median :30.50	Mode :character	Median : 79.33
Mean :30.50	NA	Mean : 79.16
3rd Qu.:45.25	NA	3rd Qu.: 87.32
Max. :60.00	NA	Max. :103.69

[1] I use the function `kable()` for nicer output in the html file. When you replicate the code in R-Studio its best to skip the part `|> kable()`.

```r
glimpse(communication_study) |>
  kable()
```

```
Rows: 90
Columns: 4
$ participant_id       <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15~
$ communication_method <chr> "face_to_face", "face_to_face", "face_to_face", "~
$ satisfaction_score   <dbl> 5.587774, 7.946131, 8.161050, 5.533329, 6.342772,~
$ task_completion_time <dbl> 22.63382, 21.65397, 31.79263, 32.06459, 21.33613,~
```

participant_id	communication_method	satisfaction_score	task_completion_time
1	face_to_face	5.587774	22.63382
2	face_to_face	7.946131	21.65397
3	face_to_face	8.161050	31.79263
4	face_to_face	5.533329	32.06459
5	face_to_face	6.342772	21.33613
6	face_to_face	6.811127	24.59724
7	face_to_face	8.028772	29.05098
8	face_to_face	7.500657	27.51294
9	face_to_face	8.408823	30.62011
10	face_to_face	7.887882	24.14360
11	face_to_face	6.101027	25.59076
12	face_to_face	8.773317	29.34951
13	face_to_face	8.386472	24.54032
14	face_to_face	9.184714	25.34449
15	face_to_face	5.471034	16.58787
16	face_to_face	9.536828	30.58478
17	face_to_face	9.284323	18.24321
18	face_to_face	7.664980	22.31417
19	face_to_face	9.936041	23.14943
20	face_to_face	9.045460	26.77006
21	face_to_face	6.630475	20.95087
22	face_to_face	5.139229	24.10138
23	face_to_face	5.487804	19.26170
24	face_to_face	7.449883	25.48121
25	face_to_face	7.156997	29.07350
26	face_to_face	8.561141	29.91470
27	face_to_face	6.644574	29.25422
28	face_to_face	6.805939	31.03674
29	face_to_face	8.981447	17.83111
30	face_to_face	5.892747	19.49527
31	video_call	6.112567	26.33775
32	video_call	6.028069	24.15182
33	video_call	4.201410	33.14125
34	video_call	7.184999	29.88306
35	video_call	7.021813	35.33477

participant_id	communication_method	satisfaction_score	task_completion_time
36	video_call	9.160348	21.24203
37	video_call	5.941216	29.41778
38	video_call	6.617673	20.36578
39	video_call	6.248827	29.89410
40	video_call	6.749644	30.13366
41	video_call	6.762376	28.02431
42	video_call	7.310949	34.89517
43	video_call	6.475502	25.02031
44	video_call	6.908485	23.51922
45	video_call	9.502537	20.30340
46	video_call	6.957107	21.80937
47	video_call	6.953594	15.56772
48	video_call	7.801070	25.44360
49	video_call	5.271977	23.33815
50	video_call	7.331750	26.90913
51	video_call	5.935564	22.82616
52	video_call	6.466518	26.90475
53	video_call	7.682417	36.22417
54	video_call	7.965898	36.05631
55	video_call	7.603863	21.08943
56	video_call	7.750890	32.49259
57	video_call	6.262873	37.13289
58	video_call	8.825157	19.28188
59	video_call	7.504208	35.26455
60	video_call	7.550296	22.54393
61	email	2.743407	43.40287
62	email	4.525116	33.84209
63	email	5.619205	31.54055
64	email	7.154302	24.04801
65	email	5.839037	32.51943
66	email	8.206231	27.84625
67	email	8.088619	43.34264
68	email	3.896070	36.60520
69	email	5.137568	26.09195
70	email	4.820245	39.84311
71	email	5.662017	34.79872
72	email	5.842933	27.27318
73	email	3.491244	27.75308
74	email	7.921749	35.97967
75	email	5.871980	39.77910
76	email	6.278325	27.05108
77	email	5.741893	47.40781
78	email	4.703157	34.34252
79	email	4.452763	18.75168
80	email	6.210837	36.05728
81	email	4.388725	30.16197
82	email	5.282076	40.81300

participant_id	communication_method	satisfaction_score	task_completion_time
83	email	8.728889	43.34819
84	email	6.001999	37.16937
85	email	6.892344	24.15431
86	email	7.090713	32.69167
87	email	3.819324	25.11418
88	email	7.188742	28.00168
89	email	6.107660	18.82483
90	email	5.568385	41.57033

```
summary(communication_study) |>
  kable()
```

	participant_id	communication_method	satisfaction_score	task_completion_time
	Min. : 1.00	Length:90	Min. :2.743	Min. :15.57
	1st Qu.:23.25	Class :character	1st Qu.:5.840	1st Qu.:23.38
	Median :45.50	Mode :character	Median :6.784	Median :27.63
	Mean :45.50	NA	Mean :6.752	Mean :28.43
	3rd Qu.:67.75	NA	3rd Qu.:7.789	3rd Qu.:32.65
	Max. :90.00	NA	Max. :9.936	Max. :47.41

```
glimpse(factorial_study) |>
  kable()
```

```
Rows: 120
Columns: 4
$ participant_id          <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14,~
$ feedback_type           <chr> "positive", "positive", "positive", "positive"~
$ experience_level        <chr> "novice", "expert", "novice", "expert", "novic~
$ performance_improvement <dbl> 9.572290, 1.217696, 7.940961, 8.549420, 6.9159~
```

participant_id	feedback_type	experience_level	performance_improvement
1	positive	novice	9.5722901
2	positive	expert	1.2176963
3	positive	novice	7.9409608
4	positive	expert	8.5494197
5	positive	novice	6.9159456
6	positive	expert	6.5465480
7	positive	novice	6.0010612
8	positive	expert	7.4765922
9	positive	novice	4.9671210
10	positive	expert	10.2190882
11	positive	novice	6.7931111

participant_id	feedback_type	experience_level	performance_improvement
12	positive	expert	4.9916590
13	positive	novice	7.4668424
14	positive	expert	6.5363013
15	positive	novice	10.7837222
16	positive	expert	5.6767243
17	positive	novice	9.2686134
18	positive	expert	6.1791098
19	positive	novice	8.6281249
20	positive	expert	5.6679884
21	positive	novice	5.8938496
22	positive	expert	10.0504075
23	positive	novice	5.4269373
24	positive	expert	9.1032825
25	positive	novice	3.7109239
26	positive	expert	6.4630255
27	positive	novice	7.1956689
28	positive	expert	7.4023036
29	positive	novice	10.5697228
30	positive	expert	7.4996319
31	positive	novice	5.0608211
32	positive	expert	3.1625578
33	positive	novice	7.5232869
34	positive	expert	7.4146383
35	positive	novice	4.1598785
36	positive	expert	3.3536203
37	positive	novice	2.7053393
38	positive	expert	5.4890255
39	positive	novice	4.4965277
40	positive	expert	7.6515065
41	positive	novice	5.5882754
42	positive	expert	10.3988276
43	positive	novice	10.0301169
44	positive	expert	8.0079965
45	positive	novice	9.1630881
46	positive	expert	7.2774370
47	positive	novice	-1.7190502
48	positive	expert	7.1855728
49	positive	novice	7.5647350
50	positive	expert	8.4529416
51	positive	novice	5.8807253
52	positive	expert	2.2009629
53	positive	novice	7.9873411
54	positive	expert	5.6394187
55	positive	novice	4.2337552
56	positive	expert	7.5267812
57	positive	novice	8.7126541
58	positive	expert	4.2216269

participant_id	feedback_type	experience_level	performance_improvement
59	positive	novice	8.8907201
60	positive	expert	9.0889070
61	critical	novice	2.0326341
62	critical	expert	5.1983262
63	critical	novice	2.1374135
64	critical	expert	10.0455398
65	critical	novice	4.0429921
66	critical	expert	1.9634775
67	critical	novice	0.2921397
68	critical	expert	1.8184332
69	critical	novice	0.1086236
70	critical	expert	5.3079254
71	critical	novice	4.6340587
72	critical	expert	6.1491606
73	critical	novice	6.5785096
74	critical	expert	1.9853834
75	critical	novice	-0.1735126
76	critical	expert	2.2069328
77	critical	novice	7.5052173
78	critical	expert	5.3630412
79	critical	novice	4.0040997
80	critical	expert	-1.8817379
81	critical	novice	8.5484264
82	critical	expert	4.4247997
83	critical	novice	0.0383440
84	critical	expert	3.7675867
85	critical	novice	-1.2327570
86	critical	expert	3.2635254
87	critical	novice	3.9437810
88	critical	expert	1.6835458
89	critical	novice	-1.3500856
90	critical	expert	6.9711815
91	critical	novice	7.8052576
92	critical	expert	11.6313395
93	critical	novice	6.9572997
94	critical	expert	6.1576483
95	critical	novice	2.4506501
96	critical	expert	11.4644559
97	critical	novice	9.4960547
98	critical	expert	7.4417990
99	critical	novice	10.1250463
100	critical	expert	12.7480371
101	critical	novice	7.6358686
102	critical	expert	8.0025776
103	critical	novice	10.4634516
104	critical	expert	9.4096561
105	critical	novice	9.3525635

participant_id	feedback_type	experience_level	performance_improvement
106	critical	expert	10.4294579
107	critical	novice	10.7682597
108	critical	expert	6.3096269
109	critical	novice	10.7119416
110	critical	expert	8.5598122
111	critical	novice	9.5967156
112	critical	expert	4.1034680
113	critical	novice	1.5549797
114	critical	expert	10.7778670
115	critical	novice	8.1236984
116	critical	expert	11.1705505
117	critical	novice	9.9640998
118	critical	expert	7.0220472
119	critical	novice	9.8529584
120	critical	expert	12.4556351

```
summary(factorial_study) |>
  kable()
```

	participant_id	feedback_type	experience_level	performance_improvement
	Min. : 1.00	Length:120	Length:120	Min. :-1.882
	1st Qu.: 30.75	Class :character	Class :character	1st Qu.: 4.206
	Median : 60.50	Mode :character	Mode :character	Median : 6.937
	Mean : 60.50	NA	NA	Mean : 6.367
	3rd Qu.: 90.25	NA	NA	3rd Qu.: 8.757
	Max. :120.00	NA	NA	Max. :12.748

Short recap: How have these data sets been created? How do they connect to the experimental designs discussed in the lecture?

::: {.callout-tip title="Possible answers", collapse="true"}

- **Dataset 1**: Classic randomized controlled trial (RCT) with treatment and control groups
- **Dataset 2**: One-way experimental design with three conditions (between-subjects)
- **Dataset 3**: 2×2 factorial design allowing us to test main effects and interactions
- **Connection to lecture**: These represent the three main experimental designs we discussed - simple, multi-group, and factorial :::

---

3 Part 2: Simple Experiments - t-tests

Assume we are asking the following research question:

Does leadership training improve team performance?

One way to tackle this question is to compare a treatment group, which has received a leadership training, to a control group, which has not received such training. If the groups are otherwise similar, then this setting should help us to identify the causal effect of the leadership training.[2]

3.1 Descriptive statistics

For this task, we will use the first data set. Let us first compute the standard statistics:

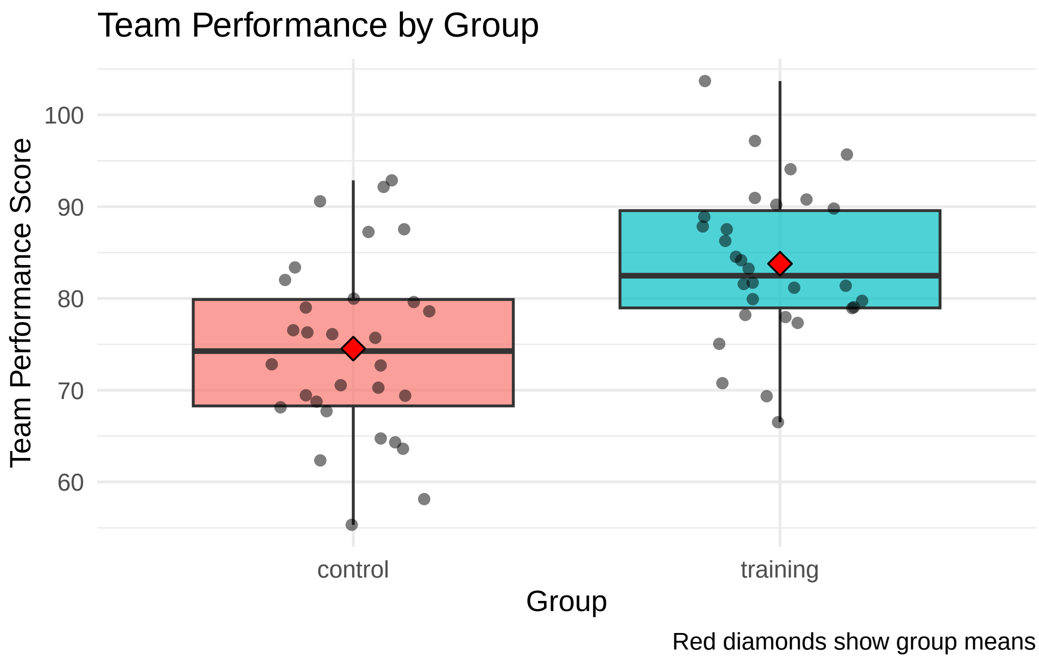
```
descriptive_stats <- leadership_study_between %>%
  group_by(group) %>%
  summarise(
    n = n(),
    mean = mean(team_performance),
    sd = sd(team_performance),
    median = median(team_performance),
    .groups = 'drop'
  )

kable(descriptive_stats, digits = 2)
```

group	n	mean	sd	median
control	30	74.53	9.81	74.26
training	30	83.78	8.35	82.48

As usual, it is also strongly recommended to complement the quantitative info with a visualization. Data such as those is often presented using boxplots:

```
ggplot(leadership_study_between, aes(x = group, y = team_performance, fill = group)) +
  geom_boxplot(alpha = 0.7) +
  geom_jitter(width = 0.2, alpha = 0.5) +
  stat_summary(fun = mean, geom = "point", shape = 23, size = 3, fill = "red") +
  labs(title = "Team Performance by Group",
       x = "Group",
       y = "Team Performance Score",
       caption = "Red diamonds show group means") +
  theme_minimal() +
  theme(legend.position = "none")
```

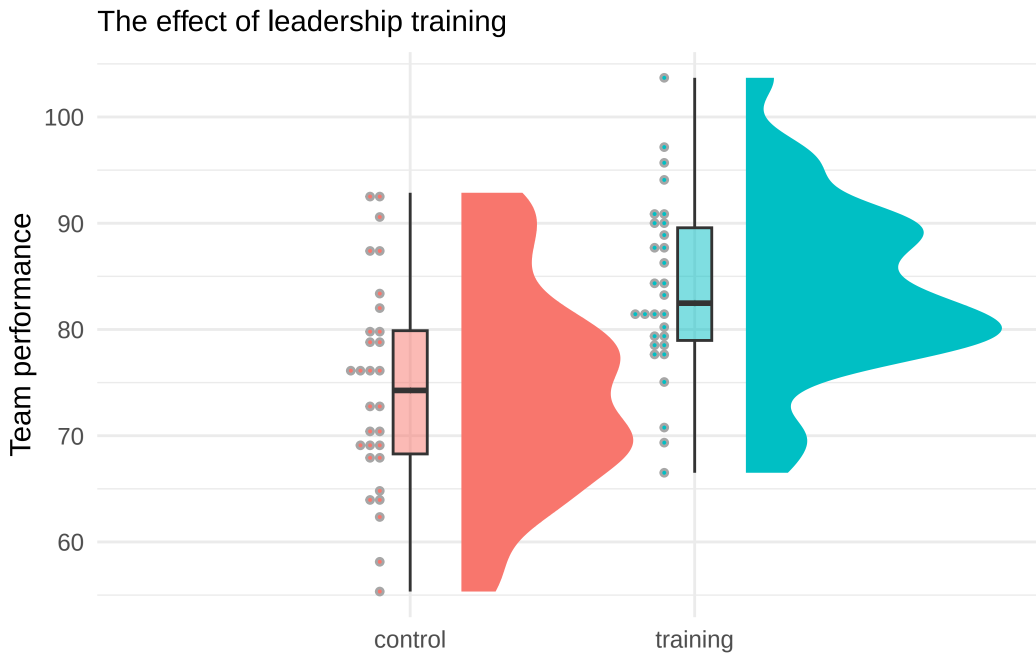
[2] At this point we assume that the groups were similar before the training. In practice, it would be good to first make sure the performances of the groups before the training were similar.



But boxplots might shallow important distributional info, so you should use them carefully or complement them with other tools. Below is an alternative that provides more information on the distribution of the data. For more on this issue see Holtz (2025).

```
ggplot(
  data = leadership_study_between,
  mapping = aes(x = group, y = team_performance, fill = group)
  ) +
    stat_halfeye(
    adjust = 0.5,
    justification = -0.2,
    .width = 0,
    point_colour = NA
  ) +
  geom_boxplot(
    width = 0.12,
    outlier.color = NA,
    alpha = 0.5
  ) +
  stat_dots(
    side = "left",
    justification = 1.1,
    binwidth = 0.85
  ) +
  labs(
    title = "The effect of leadership training",
    y = "Team performance") +
  theme_minimal() +
  theme(
```

```
    legend.position = "none",
    plot.title = element_text(size = 11),
    axis.title.x = element_blank()
  )
```



3.2 Assumption Checking

In the following we want to compare the means across independent groups. To this end, we may use a t-test.

But such statistical tests make specific assumptions about the data. If these assumptions are violated, the results may be unreliable or incorrect. Therefore, it is important to check the adequacy of the data first.

And no worries if the assumptions for one test are violated - usually there are alternatives available.

In the present case, we want to use a simple t-test. This test makes two assumptions:

1. The two groups each are normally distributed.
2. The variances of both groups are the same.

To test the first assumption, we can use the **Shapiro-Wilk Test for Normality**. Here we test the following hypotheses:

- H_0: The data is normally distributed
- H_1: The data is not normally distributed

Thus, we we get $p > 0.05$, we cannot reject H_0. But for smaller p-values, we should reject H_0 and need to look for alternative tests.

```
leadership_study_between |>
  filter(group=="control") |>
  pull(team_performance) |>
  shapiro.test()
```

```
	Shapiro-Wilk normality test

data:  pull(filter(leadership_study_between, group == "control"), team_performance)
W = 0.97894, p-value = 0.7966
```

```
leadership_study_between |>
  filter(group=="training") |>
  pull(team_performance) |>
  shapiro.test()
```

```
	Shapiro-Wilk normality test

data:  pull(filter(leadership_study_between, group == "training"), team_performance)
W = 0.98662, p-value = 0.9614
```

Good! We cannot reject the hypothesis of normally distributed data!

The next step is to test, whether both groups have the same variance. Levene's test can be used to do exactly this. It tests:

- H_0: The variances are equal across groups
- **H__1**: The variances are not equal across groups

If $p > 0.05$, we do not reject H_0 and we can use a simple t-test. If we have to reject H_0, however, it would be better to use the more robust Welch test.

```
car::leveneTest(team_performance ~ group, data = leadership_study_between)
```

```
Warning in leveneTest.default(y = y, group = group, ...): group coerced to
factor.
```

```
Levene's Test for Homogeneity of Variance (center = median)
      Df F value Pr(>F)
group  1  1.0763 0.3038
      58
```

3.3 Independent t-test

Why we use t-tests: To compare means between two groups when we have continuous data and want to test if there's a statistically significant difference.

```
t_test_result <- t.test(
  team_performance ~ group,
  data = leadership_study_between,
  var.equal = TRUE # Use FALSE if variances unequal
  )
t_test_result
```

```

    Two Sample t-test

data:  team_performance by group
t = -3.9344, df = 58, p-value = 0.0002256
alternative hypothesis: true difference in means between group control and group training
95 percent confidence interval:
 -13.962870  -4.545972
sample estimates:
 mean in group control mean in group training
              74.52896               83.78338
```

3.4 Effect Size Calculation

What are effect sizes? The previous result tells us that the difference in means between the groups appeaers to be about -9.25. But is this a lot? Effect sizes tell us about the practical significance of our findings - how big is the difference we found? Unlike *p*-values, effect sizes are not influenced by sample size and help us understand if our statistically significant result is also practically meaningful. **Cohen's d** is often used: This standardized effect size tells us how many standard deviations apart the two group means are.

- **Small effect**: d 0.2 (groups overlap about 85%)
- **Medium effect**: d 0.5 (groups overlap about 67%)

- **Large effect**: d 0.8 (groups overlap about 53%)

The implementation in R is trivial:

```
cohens_d <- effectsize::cohens_d(team_performance ~ group, data = leadership_study_betwe
print(cohens_d)
```

```
Cohen's d |         95% CI
--------------------------
-1.02     | [-1.55, -0.47]

- Estimated using pooled SD.
```

The key value here is Cohen's of -1.02! (For the interpretation see the exercise below).

3.5 Paired t-test Example

Next, we might want to look at our research question from a slightly different angle. Rather than the between-subject design from above, we now take a *within-subject* view: to this end, we want to check whether the training had an effect on those people who were in the training (treatment) group by comparing their performance before and after the training.

To this end, we focus on the training group, and then use the function `t.test()` with the argument `paired = TRUE`. This makes sure we are using the version of the test for the within-subjects context:

```
training_group <- leadership_study_within %>% filter(group == "training")
paired_result <- t.test(training_group$post_performance,
                       training_group$pre_performance,
                       paired = TRUE)
print(paired_result)
```

```
    Paired t-test

data:  training_group$post_performance and training_group$pre_performance
t = 7.4774, df = 29, p-value = 3.059e-08
alternative hypothesis: true mean difference is not equal to 0
95 percent confidence interval:
 10.55898 18.51001
sample estimates:
mean difference
       14.53449
```

Exercise (5 minutes): Interpret the results. What can we conclude about the effectiveness of leadership training?

::: {.callout-tip title="Possible answers", collapse="true"} - **Statistical significance**: If $p < 0.05$, training significantly improved performance; since $p \approx 0$, the training has a highly significant effect - **Effect size interpretation**: Cohen's d is large, so we have a large effect. This suggests the effect of the training is also practically meaningful. - **Confidence interval**: If the CI doesn't include 0, we're confident there's a real difference; even if we are very conservative, we would still expect a 10 point improvement of the training. - **Business implication**: Training appears effective and worth the investment :::

---

4 Part 3: Multi-Group Experiments - ANOVA (25 minutes)

Let us now turn to the following research question:

> Which communication method (face-to-face, video call, email) leads to highest satisfaction?

Note that this time we not only compare one group to another as in the previous section, but we need to compare three groups with each other as we have three different communication methods. Therefore, we cannot use simple t-tests, but need to use an ANOVA.

4.1 Descriptive statistics

But first, let us again look at the data:

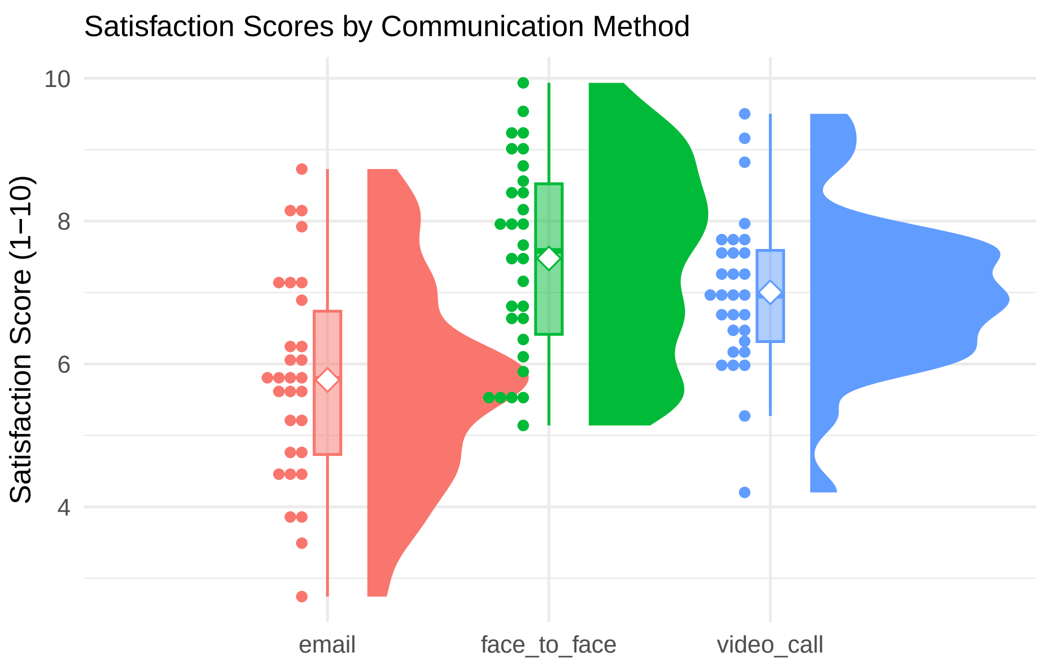
```
communication_study %>%
  group_by(communication_method) %>%
  summarise(
    n = n(),
    mean = mean(satisfaction_score),
    sd = sd(satisfaction_score),
    min = min(satisfaction_score),
    max = max(satisfaction_score),
    .groups = 'drop'
  ) |>
  kable(digits = 2)
```

communication_method	n	mean	sd	min	max
email	30	5.78	1.46	2.74	8.73
face_to_face	30	7.48	1.39	5.14	9.94
video_call	30	7.00	1.10	4.20	9.50

And complement this by a visualization:

```
ggplot(
  data = communication_study,
  mapping = aes(
    x = communication_method,
    y = satisfaction_score,
    fill = communication_method,
    color = communication_method)
  ) +
    stat_halfeye(
```

```
    adjust = 0.5,
    justification = -0.2,
    .width = 0,
    point_colour = NA
  ) +
  geom_boxplot(
    width = 0.12,
    outlier.color = NA,
    alpha = 0.5
  ) +
  stat_dots(
    side = "left",
    justification = 1.1,
    binwidth = 0.15
  ) +
  stat_summary(fun = mean, geom = "point", shape = 23, size = 3, fill = "white") +
  labs(
    title = "Satisfaction Scores by Communication Method",
    x = "Communication Method",
    y = "Satisfaction Score (1-10)") +
  theme_minimal() +
  theme(
    legend.position = "none",
    plot.title = element_text(size = 11),
    axis.title.x = element_blank()
  )
```



4.2 ANOVA Assumptions

ANOVA is more robust than t-tests but still requires certain conditions to be met for valid results. In fact, we are testing the same assumptions as in the t-test case:

- **Normality of Residuals**: For ANOVA, we check if the residuals (not the raw data) are normally distributed.

Homogeneity of Variances: ANOVA assumes that the variance of the dependent variable is equal across all groups.

Let us start with testing the normality of the residuals. We again use the Shapiro test, which tests the following hypothesis:

- H_0: Residuals are normally distributed
- H_1: Residuals are not normally distributed

Thus, if $p > 0.05$, the Null cannot be rejected and we can assume the residuals to follow a normal distribution. If $p \leq 0.05$, however, the hypothesis of normally distributed residuals must be rejected and we need to consider transforming the data or using a non-parametric test.

```
aov_model <- aov(satisfaction_score ~ communication_method, data = communication_study)

shapiro.test(residuals(aov_model))
```

```

	Shapiro-Wilk normality test

data:  residuals(aov_model)
W = 0.99149, p-value = 0.8342
```

Since $p > 0.05$ we are on the save side!

We then check the equality of variances and again use Levene's test with the following hypotheses:

- H_0: Variances are equal across all groups

- H_1: Variances are not equal across groups

Thus, if $p > 0.05$, the Null cannot be rejected and we can assume the variances to be equal. If $p \leq 0.05$, however, the hypothesis of equal variances must be rejected and we need to consider transforming the data or using Welch's ANOVA.

```
car::leveneTest(satisfaction_score ~ communication_method, data = communication_study)
```

```
Warning in leveneTest.default(y = y, group = group, ...): group coerced to
factor.
```

```
Levene's Test for Homogeneity of Variance (center = median)
      Df F value Pr(>F)
group  2  1.8801 0.1587
      87
```

Again, the Null cannot be rejected and we can continue with the ANOVA as planned.

Before we enter the actual analysis we can also create the common diagnostics plots:

```
model_data <- augment(aov_model) %>%
  mutate(
    sqrt_abs_resid = sqrt(abs(.std.resid)),
    obs_number = row_number()
  )
```

```
Warning: The `augment()` method for objects of class `aov` is not maintained by the broom

This warning is displayed once per session.
```

```
residuals_vs_fitted <- model_data %>%
  ggplot(aes(x = .fitted, y = .resid)) +
  geom_point(alpha = 0.7) +
  geom_smooth(method = "loess", se = FALSE, color = "red") +
  geom_hline(yintercept = 0, linetype = "dashed", alpha = 0.7) +
  labs(title = "Residuals vs Fitted", x = "Fitted values", y = "Residuals") +
  theme_minimal()

qq_plot <- model_data %>%
  ggplot(aes(sample = .std.resid)) +
  stat_qq(alpha = 0.7) +
  stat_qq_line(color = "red") +
  labs(title = "Normal Q-Q", x = "Theoretical Quantiles", y = "Standardized Residuals") -
  theme_minimal()

scale_location <- model_data %>%
  ggplot(aes(x = .fitted, y = sqrt_abs_resid)) +
  geom_point(alpha = 0.7) +
  geom_smooth(method = "loess", se = FALSE, color = "red") +
  labs(title = "Scale-Location", x = "Fitted values",
       y = expression(sqrt("|Standardized residuals|"))) +
  theme_minimal()

residuals_vs_leverage <- model_data %>%
  ggplot(aes(x = .hat, y = .std.resid)) +
  geom_point(alpha = 0.7) +
  geom_smooth(method = "loess", se = FALSE, color = "red") +
  geom_hline(yintercept = 0, linetype = "dashed", alpha = 0.7) +
  labs(title = "Residuals vs Leverage", x = "Leverage", y = "Standardized Residuals") +
```

```r
  theme_minimal()

combined_patchwork <- (residuals_vs_fitted | qq_plot) /
                            (scale_location | residuals_vs_leverage) +
  plot_annotation(
    title = "ANOVA Model Diagnostic Plots",
    subtitle = "Checking model assumptions",
    theme = theme(plot.title = element_text(size = 16, hjust = 0.5))
  )
combined_patchwork
```

```
`geom_smooth()` using formula = 'y ~ x'
```

```
Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: pseudoinverse used at 5.7673
```

```
Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: neighborhood radius 1.7108
```

```
Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: reciprocal condition number 5.2991e-16
```

```
Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: There are other near singularities as well. 2.9267
```

```
`geom_smooth()` using formula = 'y ~ x'
```

```
Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: pseudoinverse used at 5.7673
```

```
Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: neighborhood radius 1.7108
```

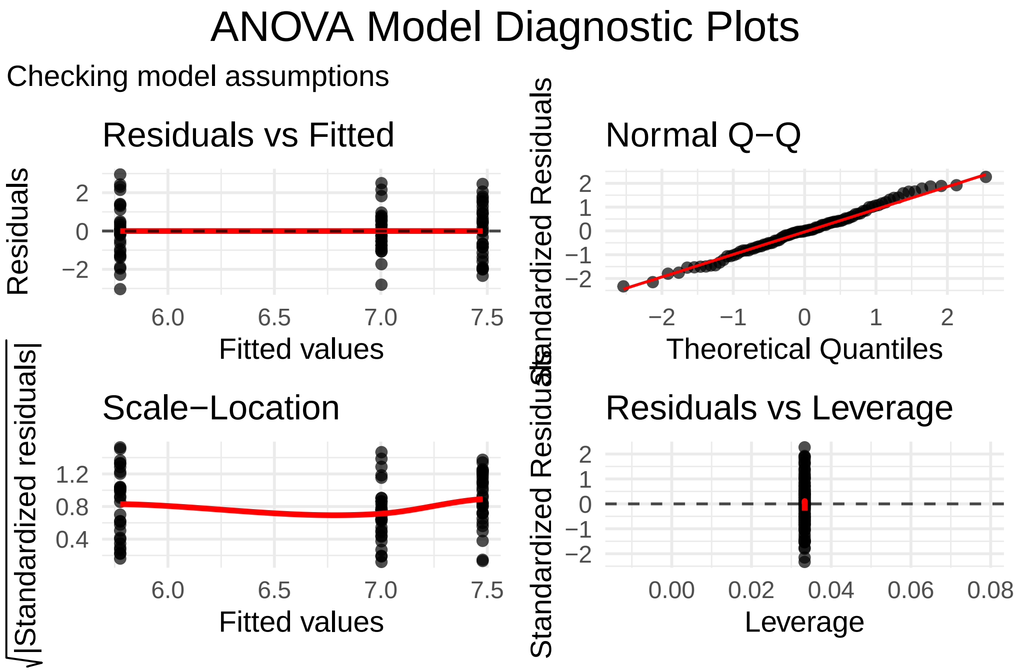
```
Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: reciprocal condition number 5.2991e-16
```

```
Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: There are other near singularities as well. 2.9267
```

```
`geom_smooth()` using formula = 'y ~ x'
```

```
Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: pseudoinverse used at 0.033333
```

```
Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: neighborhood radius 1.6695e-14

Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: reciprocal condition number 7.744e-16

Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: There are other near singularities as well. 4.3333e-34
```



- **Residuals vs Fitted**: Should show random scatter (no patterns)
- **Q-Q plot**: Points should follow the diagonal line (normality)
- **Scale-Location**: Should show random scatter (equal variances)
- **Residuals vs Leverage**: Identifies influential outliers

4.3 One-Way ANOVA

Since we compare three groups we do not use t-tests but an ANOVA.

As you know, ANOVA is actually a special case of linear regression. Therefore, we can get the ANOVA results in two equivalent ways.

The first option is to use the classical `aov()` function:

```
anova_result <- aov(satisfaction_score ~ communication_method, data = communication_study
summary(anova_result)
```

```
                     Df Sum Sq Mean Sq F value   Pr(>F)
communication_method  2  46.29  23.146    13.2 9.81e-06 ***
Residuals            87 152.50   1.753
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The second option is to use `lm()` as we know it from linear regression:

```r
lm_model <- lm(satisfaction_score ~ communication_method, data = communication_study)
anova(lm_model)
```

```
Analysis of Variance Table

Response: satisfaction_score
                     Df  Sum Sq Mean Sq F value    Pr(>F)
communication_method  2  46.292 23.1460  13.205 9.813e-06 ***
Residuals            87 152.497  1.7528
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- The overall p-value is extremely small, so the Null hypothesis of no difference between the groups should definitely be rejected
- In other words: Communication method significantly affects satisfaction scores
- This means that at least one communication method produces significantly different satisfaction scores than the others
- But the result does not tell us which specific methods differ from each other (need post-hoc tests)
- The result also does not contain information about the direction of differences (which method is best/worst) or the effect size

4.4 Detour: categorical variables in Regression

When you add a categorial variable as a predictor to your regression, R automatically creates dummy variables for categorical predictors. The first level alphabetically becomes the reference group:

```r
summary(lm_model)$coefficients |>
  kable(digits = 3)
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	5.776	0.242	23.895	0.000
communication_methodface_to_face	1.702	0.342	4.980	0.000
communication_methodvideo_call	1.227	0.342	3.590	0.001

Since ‘email’ comes first alphabetically, it is the reference group.

- **Intercept** = mean of reference group (email, comes first alphabetically)
- **communication_methodface_to_face** = difference between face_to_face and email
- **communication_methodvideo_call** = difference between video_call and email

We can verify this manually:

```
communication_study %>%
  group_by(communication_method) %>%
  summarise(mean = mean(satisfaction_score), .groups = 'drop') |>
  kable(digits = 2)
```

communication_method	mean
email	5.78
face_to_face	7.48
video_call	7.00

4.4.1 Detour: When to Use `lm()` vs `aov()`

As shown above, both approaches give identical results. Still, they offer different perspectives:

Use aov() when: - You want traditional ANOVA output - Focus is on group comparisons - Need post-hoc tests such as `TukeyHSD()`, which take the `aov`-model as an input

Use lm() when: - You want to see specific contrasts - Planning to add continuous covariates later - Want regression-style interpretation - Building toward more complex models

Example: From ANOVA to ANCOVA

If we add a continuous variable to an ANOVA, we get an ANCOVA:

```
ancova_model <- lm(satisfaction_score ~ communication_method + task_completion_time,
                   data = communication_study)
summary(ancova_model)
```

```

Call:
lm(formula = satisfaction_score ~ communication_method + task_completion_time,
    data = communication_study)

Residuals:
     Min       1Q   Median       3Q      Max
-3.07895 -0.87806 -0.00768  0.78530  2.90678

Coefficients:
```

```
                                 Estimate Std. Error t value Pr(>|t|)
(Intercept)                      5.628267   0.807830   6.967 6.15e-10 ***
communication_methodface_to_face 1.737394   0.389664   4.459 2.48e-05 ***
communication_methodvideo_call   1.253306   0.369901   3.388  0.00106 **
task_completion_time             0.004472   0.023342   0.192  0.84853
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.331 on 86 degrees of freedom
Multiple R-squared:  0.2332,    Adjusted R-squared:  0.2064
F-statistic: 8.718 on 3 and 86 DF,  p-value: 4.108e-05
```

In the example, we now control for individual differences in task completion time. While trivial in the `lm()`-context, this would be much harder to do with the aov() approach!

4.5 Effect Size for ANOVA

η^2 tells us what proportion of the total variance in the dependent variable is explained by the independent variable and, as explained above, serves as a standardized measure for comparing effect sizes:

Remember that:

- **Small effect**: ^{2} 0.01 (1% of variance)
- **Medium effect**: ^{2} 0.06 (6% of variance)
- **Large effect**: ^{2} 0.14 (14% of variance)
- **Example**: $^2 = 0.23$ means communication method explains 23% of satisfaction variance

In our case:

```r
eta_squared <- effectsize::eta_squared(anova_result)
```

```
For one-way between subjects designs, partial eta squared is equivalent
  to eta squared. Returning eta squared.
```

```r
print(eta_squared)
```

```
# Effect Size for ANOVA

Parameter            | Eta2 |       95% CI
------------------------------------------
communication_method | 0.23 | [0.11, 1.00]

- One-sided CIs: upper bound fixed at [1.00].
```

4.6 Post-Hoc Comparisons

The ANOVA tells us there's a difference somewhere among the groups, but not which specific groups differ. This is why we need post-hoc tests: they provide these pairwise comparisons while controlling for multiple testing.

> **The multiple testing problem**
>
> If we do multiple t-tests (e.g. one for each pairwise comparison), our Type I error rate will inflate. Post-hoc tests adjust critical values to maintain overall $\alpha = 0.05$.

The most common post-hoc test has a nice name: Tukey's Honestly Significant Difference (HSD). It takes the fitted ANOVA model as its input:

```r
tukey_result <- TukeyHSD(anova_result)
print(tukey_result)
```

```
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = satisfaction_score ~ communication_method, data = communication_study)

$communication_method
                             diff        lwr       upr     p adj
face_to_face-email       1.702240  0.8871253 2.5173543 0.0000095
video_call-email         1.227135  0.4120202 2.0422492 0.0015714
video_call-face_to_face -0.475105 -1.2902195 0.3400094 0.3506328
```

This suggests that the following differences exist:

- Face-to-face > Email: 1.70 points higher satisfaction ($p < 0.001$)
- Video call > Email: 1.23 points higher satisfaction ($p = 0.002$)

But there are no significant differences between video calls and face-to-face ($p = 0.35$)

Thus, both face-to-face and video call communication methods produce significantly higher satisfaction scores than email, but face-to-face and video call don't differ significantly from each other.

5 Part 4: Factorial Designs (20 minutes)

For this last part we consider the following research question:

> How do feedback type and experience level interact to affect performance improvement?

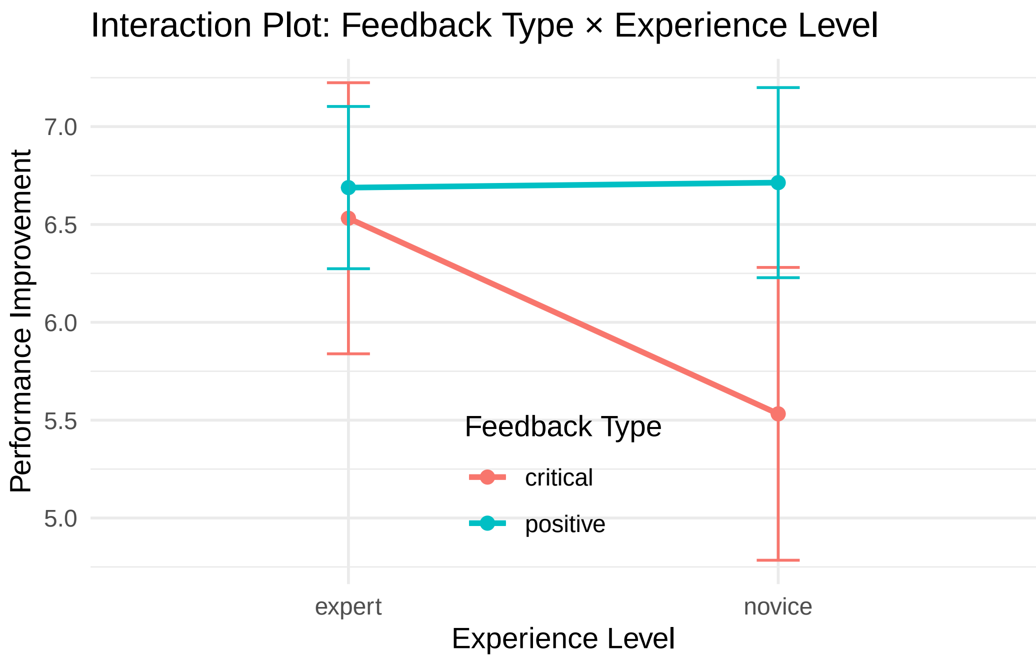
We use the data stored as `factorial_study`:

```
factorial_study %>%
  group_by(feedback_type, experience_level) %>%
  summarise(
    n = n(),
    mean = mean(performance_improvement),
    sd = sd(performance_improvement),
    .groups = 'drop'
  ) |>
  kable(digits = 2)
```

feedback_type	experience_level	n	mean	sd
critical	expert	30	6.53	3.79
critical	novice	30	5.53	4.10
positive	expert	30	6.69	2.27
positive	novice	30	6.71	2.66

We then visualize the relationship using an interaction plot as discussed in the lecture:

```
ggplot(
  data = factorial_study,
  mapping = aes(
    x = experience_level,
    y = performance_improvement,
    color = feedback_type,
    group = feedback_type)
  ) +
  stat_summary(fun = mean, geom = "point", size = 2) +
  stat_summary(fun = mean, geom = "line", linewidth = 1) +
  stat_summary(fun.data = mean_se, geom = "errorbar", width = 0.1) +
  guides(color = guide_legend(position = "inside")) +
  labs(
    title = "Interaction Plot: Feedback Type × Experience Level",
    x = "Experience Level",
    y = "Performance Improvement",
    color = "Feedback Type") +
  theme_minimal() +
  theme(legend.position.inside = c(0.5, 0.2))
```



This already gives us a good visual impression of the results, but we also want to analyze the results quantitatively.

5.0.1 Two-Way ANOVA

The factorial design allows us to consider interaction effects among factors. But to detect such interaction, we must use a two-way ANOVA, not the traditional one!

To do this, we still use the same function `aov()` (or `lm()`), but add the additional factor to the formula:

```
factorial_model <- aov(
  formula = performance_improvement ~ feedback_type * experience_level,
  data = factorial_study)
summary(factorial_model)
```

```
                                  Df Sum Sq Mean Sq F value Pr(>F)
feedback_type                      1   13.4  13.430   1.237  0.268
experience_level                   1    7.1   7.115   0.656  0.420
feedback_type:experience_level     1    7.9   7.877   0.726  0.396
Residuals                        116 1259.1  10.854
```

And while the classical ANOVA was the same as simple linear regression with categorial variables, two-way ANOVA is the same as *multiple* regression with interaction effects:

```
lm_factorial <- lm(performance_improvement ~ feedback_type * experience_level,
                   data = factorial_study)
anova(lm_factorial)
```

```
Analysis of Variance Table

Response: performance_improvement
                               Df  Sum Sq Mean Sq F value Pr(>F)
feedback_type                   1   13.43 13.4301  1.2373 0.2683
experience_level                1    7.12  7.1155  0.6556 0.4198
feedback_type:experience_level  1    7.88  7.8766  0.7257 0.3960
Residuals                     116 1259.07 10.8541
```

Dummy coding in the two-way context

With two factors, R creates dummy variables for each factor plus their interaction:

```r
coefficients(lm_factorial)
```

```
                              (Intercept)
                                6.5317033
                    feedback_typepositive
                                0.1566833
                     experience_levelnovice
                               -0.9994123
feedback_typepositive:experience_levelnovice
                                1.0247960
```

Their interpretation is as follows:

- **Intercept** = mean of reference group (critical feedback + expert)
- **feedback_typepositive** = main effect of positive vs critical for experts only
- **experience_levelnovice** = main effect of novice vs expert for critical feedback only
- **interaction** = additional effect of being novice AND receiving positive feedback

5.1 Effect Sizes for Factorial Design

Effect sizes are computed in the same way:

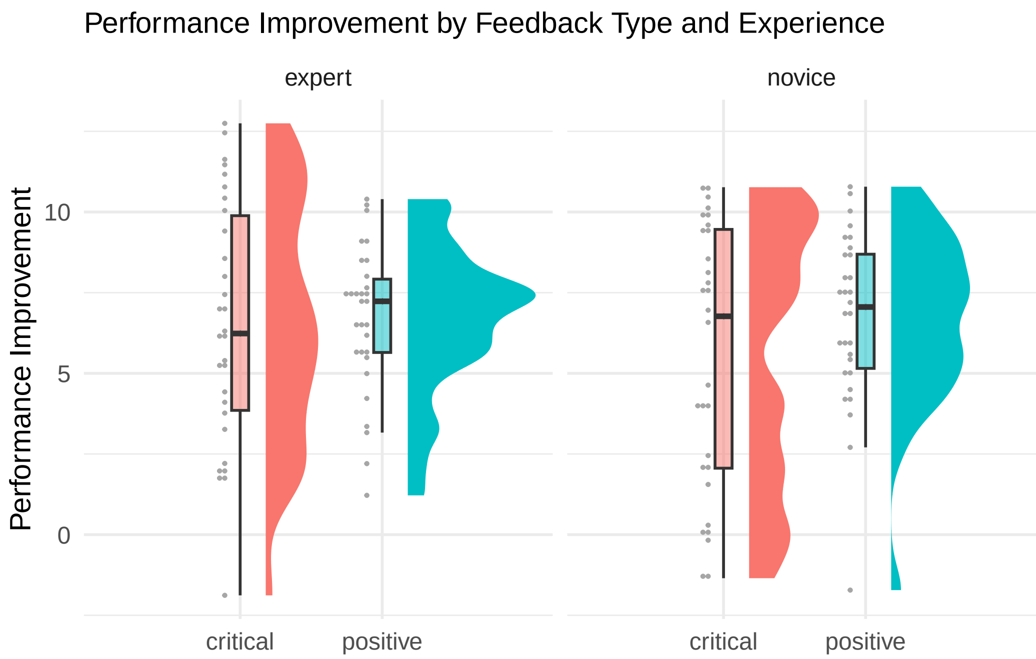
```r
eta_squared_factorial <- effectsize::eta_squared(factorial_model)
print(eta_squared_factorial)
```

```
# Effect Size for ANOVA (Type I)

Parameter                      | Eta2 (partial) |       95% CI
---------------------------------------------------------------
feedback_type                  |           0.01 | [0.00, 1.00]
experience_level               |       5.62e-03 | [0.00, 1.00]
```

```
feedback_type:experience_level |       6.22e-03 | [0.00, 1.00]

- One-sided CIs: upper bound fixed at [1.00].
```

5.2 Advanced Visualization of factorial designs

```r
ggplot(
  data = factorial_study,
  mapping = aes(
    x = feedback_type,
    y = performance_improvement,
    fill = feedback_type)
  ) +
    stat_halfeye(
    adjust = 0.5,
    justification = -0.2,
    .width = 0,
    point_colour = NA
  ) +
  geom_boxplot(
    width = 0.12,
    outlier.color = NA,
    alpha = 0.5
  ) +
  stat_dots(
    side = "left",
    justification = 1.1,
    binwidth = 0.15
  ) +
  facet_wrap(~ experience_level) +
  labs(
    title = "Performance Improvement by Feedback Type and Experience",
    x = "Feedback Type",
    y = "Performance Improvement") +
  theme_minimal() +
  theme(
    legend.position = "none",
    plot.title = element_text(size = 11),
    axis.title.x = element_blank()
  )
```



Intermediate exercise: 1. Interpret the main effects and interaction 2. What practical recommendations would you make based on these results? 3. How does this connect to the i-frame vs s-frame discussion from the lecture?

::: {.callout-tip title="Possible answers", collapse=} 2 **1. Interpretation:** - **Main effect of feedback**: If significant, one type of feedback is generally better - **Main effect of experience**: If significant, novices and experts respond differently overall - **Interaction effect**: If significant, optimal feedback depends on experience level - **Visual cues**: Parallel lines = no interaction; crossing lines = interaction present

2. Practical recommendations: - **If interaction significant**: Customize feedback approach based on experience level - **For novices**: Might need more positive, encouraging feedback - **For experts**: Might benefit from more critical, detailed feedback - **Training programs**: Should differentiate based on employee experience

3. i-frame vs s-frame connection: - **i-frame approach**: Train managers to give different feedback to different employees - **s-frame approach**: Change organizational culture and systems to support appropriate feedback - **Individual focus**: Coaching managers on feedback skills - **Structural focus**: Performance management systems that account for experience levels :::

---

6 Part 5: Power Analysis and Sample Size Planning

Statistical power is the probability of detecting an effect when it truly exists. You know from the lecture that the decision about sample sizes determines in part statistical power. Power analysis helps us plan adequate sample sizes and evaluate our study's sensitivity.

6.1 General aspects of power analysis

Remember the **components of power analysis:**

- **Power**: Probability of detecting effect (usually we want 0.8)
- **Effect size**: How big a difference we want to detect
- **Sample size**: Number of participants needed
- **Alpha level**: Type I error rate (usually 0.05)

We can use power analysis in two different ways:

1. **Post-hoc (observed)**: What was our power given the sample size we had?
2. **A priori (prospective)**: How many participants do we need to detect an effect?

Regarding the first, we may ask: what was our power to detect the effect we found in the leadership study?

```
observed_power <- pwr.t.test(n = 30, d = as.numeric(cohens_d$Cohens_d), sig.level = 0.05
print(observed_power)
```

```

     Two-sample t test power calculation 

              n = 30
              d = 1.015848
      sig.level = 0.05
          power = 0.9718339
    alternative = two.sided

NOTE: n is number in *each* group
```

Lets turn to the *a priori power analysis*, i.e. what you should do BEFORE collecting data to determine how many participants you need.

In a first step, you always need to specify your research parameters:

- What effect size do you want to detect?
- What power level do you want? (typically 0.8 or 0.9)
- What alpha level will you use? (typically 0.05)

The remaining steps depend on the analysis method we wish to employ:

6.2 t-Tests

Assume we want to plan a new leadership training study, similar to the one above. We want to detect a medium effect ($d = 0.5$) with 80% power.

```
sample_size_medium <- pwr.t.test(d = 0.5, power = 0.8, sig.level = 0.05)
print(sample_size_medium)
```

```
     Two-sample t test power calculation

              n = 63.76561
              d = 0.5
      sig.level = 0.05
          power = 0.8
    alternative = two.sided

NOTE: n is number in *each* group
```

If instead we would like to identify a small effect (with $d = 0.5$). Everything else remains the same:

```
sample_size_small <- pwr.t.test(d = 0.2, power = 0.8, sig.level = 0.05)
print(sample_size_small)
```

```
     Two-sample t test power calculation

              n = 393.4057
              d = 0.2
      sig.level = 0.05
          power = 0.8
    alternative = two.sided

NOTE: n is number in *each* group
```

And what would happen if we wanted higher power (90%) for a medium effect?

```
sample_size_high_power <- pwr.t.test(d = 0.5, power = 0.9, sig.level = 0.05)
print(sample_size_high_power)
```

```
     Two-sample t test power calculation

              n = 85.03128
              d = 0.5
      sig.level = 0.05
          power = 0.9
    alternative = two.sided

NOTE: n is number in *each* group
```

We see that small design choices can have huge effects:

```r
power_results <- tibble(
  Scenario = c("Medium effect, 80% power", "Small effect, 80% power", "Medium effect, 90%
  Effect_Size = c(0.5, 0.2, 0.5),
  Power = c(0.8, 0.8, 0.9),
  Sample_per_Group = c(
    ceiling(sample_size_medium$n),
    ceiling(sample_size_small$n),
    ceiling(sample_size_high_power$n)
  ),
  Total_Sample = c(
    ceiling(sample_size_medium$n) * 2,
    ceiling(sample_size_small$n) * 2,
    ceiling(sample_size_high_power$n) * 2
  )
)

kable(power_results)
```

Scenario	Effect_Size	Power	Sample_per_Group	Total_Sample
Medium effect, 80% power	0.5	0.8	64	128
Small effect, 80% power	0.2	0.8	394	788
Medium effect, 90% power	0.5	0.9	86	172

##ANOVA

Assume we are planning a communication study with 3 groups and we want to detect a medium effect (f = 0.25) with 80% power. Note that what was Cohen's d for the t-test case, has now become Cohens f for the ANOVA case:

```r
sample_size_anova <- pwr.anova.test(k = 3, f = 0.25, sig.level = 0.05, power = 0.8)
print(sample_size_anova)
```

```
     Balanced one-way analysis of variance power calculation

              k = 3
              n = 52.3966
              f = 0.25
      sig.level = 0.05
          power = 0.8

NOTE: n is number in each group
```

7 Summary and Key Takeaways

7.1 Key take-aways

1. **Data exploration** is crucial before statistical testing
2. **Assumption checking** ensures valid results and guides method selection
3. **Effect sizes** provide practical significance context beyond p-values
4. **Visualization** aids interpretation and communication of results
5. **ANOVA is just regression** with categorical predictors
6. **Both aov() and lm()** give identical results but offer different perspectives
7. **Post-hoc tests** control for multiple comparisons when making pairwise comparisons
8. **Factorial designs** allow detection of interactions between factors

7.1.1 Key Decision Points in Analysis

- **Assumptions violated**: Choose appropriate alternatives (Welch's tests, transformations, non-parametric)
- **Multiple groups**: ANOVA preferred over multiple t-tests
- **Factorial designs**: Allow testing of interactions between factors
- **Effect size interpretation**: Always consider practical alongside statistical significance
- **Post-hoc testing**: Required when ANOVA is significant to identify which groups differ

Holtz, Y. (2025) 'The boxplot and its pitfalls', *From Data to Viz*, available at https://www.data-to-viz.com/caveat/boxplot.html.