Statistics for Neuroscience – Lab Activity

Livio

2026-02-20

Activity: Discover how contrasts and centering change your model

Goal
By fitting models with different encodings, *discover* for yourself:

- How dummy vs sum contrasts change coefficient *meaning*
- Why centering continuous predictors matters (especially with interactions)
- How to get "clean" interpretations of main effects

Instructions
Work step-by-step. Compare `summary()` outputs and plots. Explain what you observe in your own words.

---

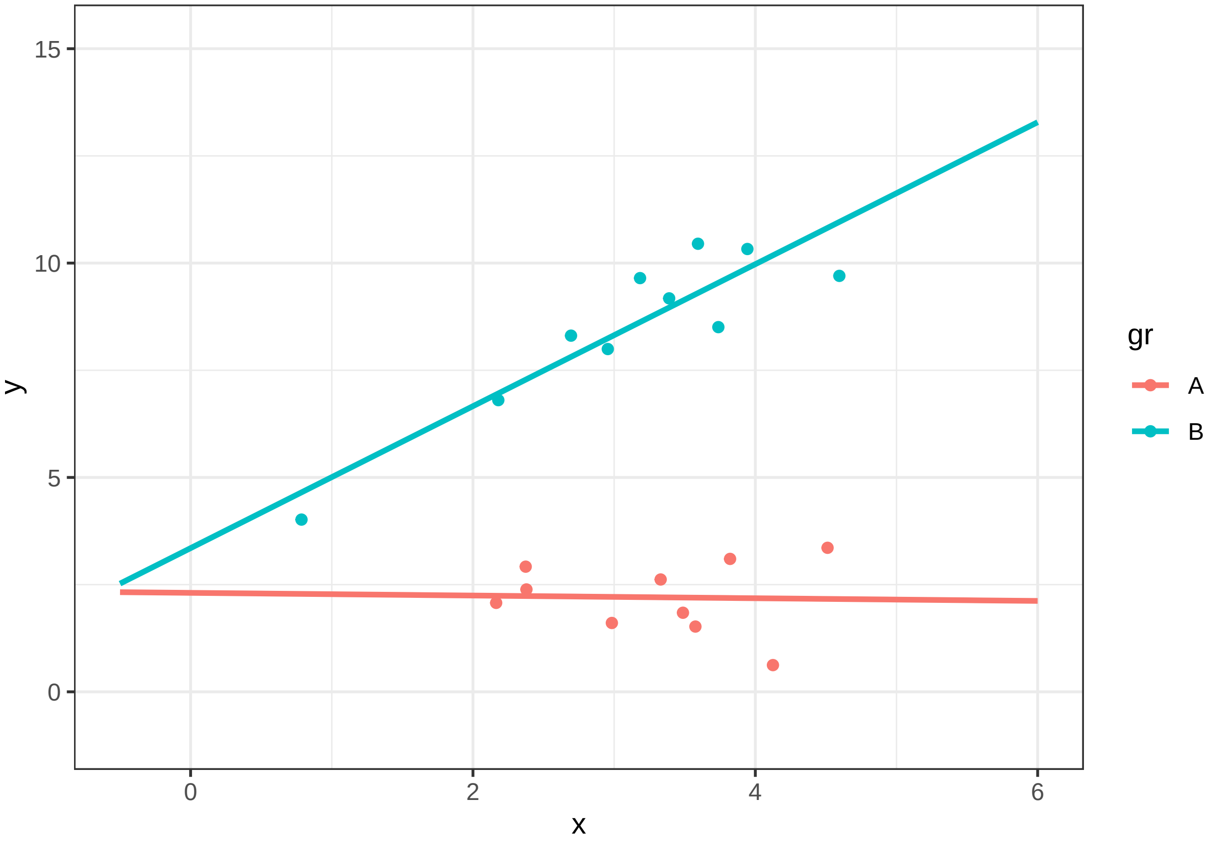
1. Generate the data and visualize it

Run:

```r
set.seed(1)
n0 <- 10
D <- data.frame(
  gr = as.factor(rep(LETTERS[1:2], n0)),
  x  = rnorm(n0 * 2) + 3
)
mu <- 2 + (D$gr == "B") * .5 + D$x * (D$gr == "B") * 2
D$y <- mu + rnorm(n0 * 2)

library(ggplot2)
ggplot(D, aes(x = x, y = y, color = gr)) +
  geom_point() +
  geom_smooth(method = "lm", fill = NA, fullrange = TRUE) +
  xlim(-.5, 6) +
  theme_bw()
```

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



Tasks

1. Write down (from the code) the *true* generative model for groups A and B (two separate regression equations).

2. From the plot, describe:
   - Does x affect y in group A?

   - Does x affect y in group B?

   - Is there an interaction between gr and x?

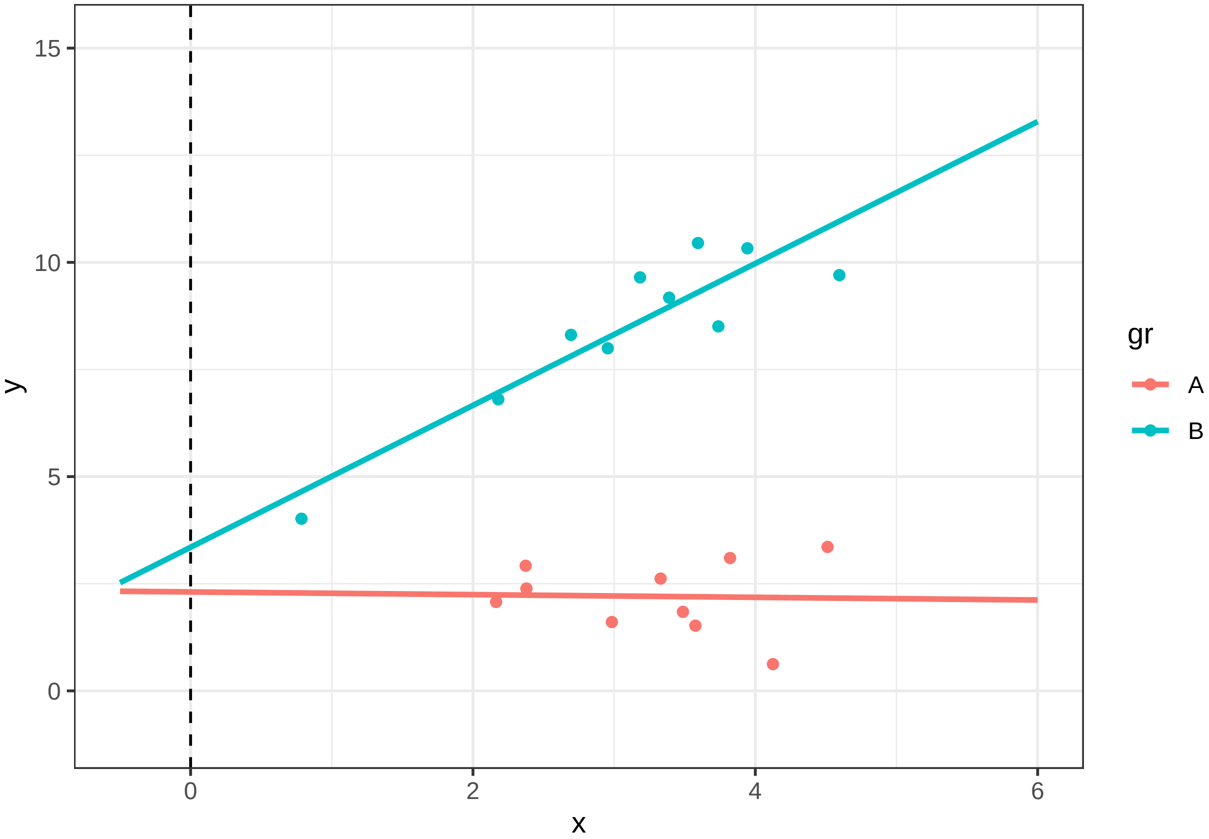
2. Model 1: Default dummy coding (no centering)

```
mod1 <- lm(y ~ gr * x, data = D)
summary(mod1)
```

```
##
## Call:
## lm(formula = y ~ gr * x, data = D)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.55585 -0.61528 -0.00131  0.54234  1.19203
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.30818    1.24368   1.856  0.08198 .
## grB          1.04292    1.53659   0.679  0.50701
## x           -0.03137    0.37016  -0.085  0.93352
## grB:x        1.68703    0.46200   3.652  0.00215 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8778 on 16 degrees of freedom
## Multiple R-squared:  0.9481, Adjusted R-squared:  0.9384
## F-statistic: 97.49 on 3 and 16 DF,  p-value: 1.708e-10
```

```
p2 <- ggplot(D, aes(x = x, y = y, color = gr)) +
  geom_point() +
  geom_smooth(method = "lm", fill = NA, fullrange = TRUE) +
  geom_vline(xintercept = 0, linetype = "dashed") +
  xlim(-.5, 6) +
  theme_bw()
print(p2)
```

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



Discover

1. What does the grB coefficient mean? (Hint: at what x?)
2. Is the group difference meaningful given where data are?
3. Look at p-values. Which effects are “significant”? Interaction?

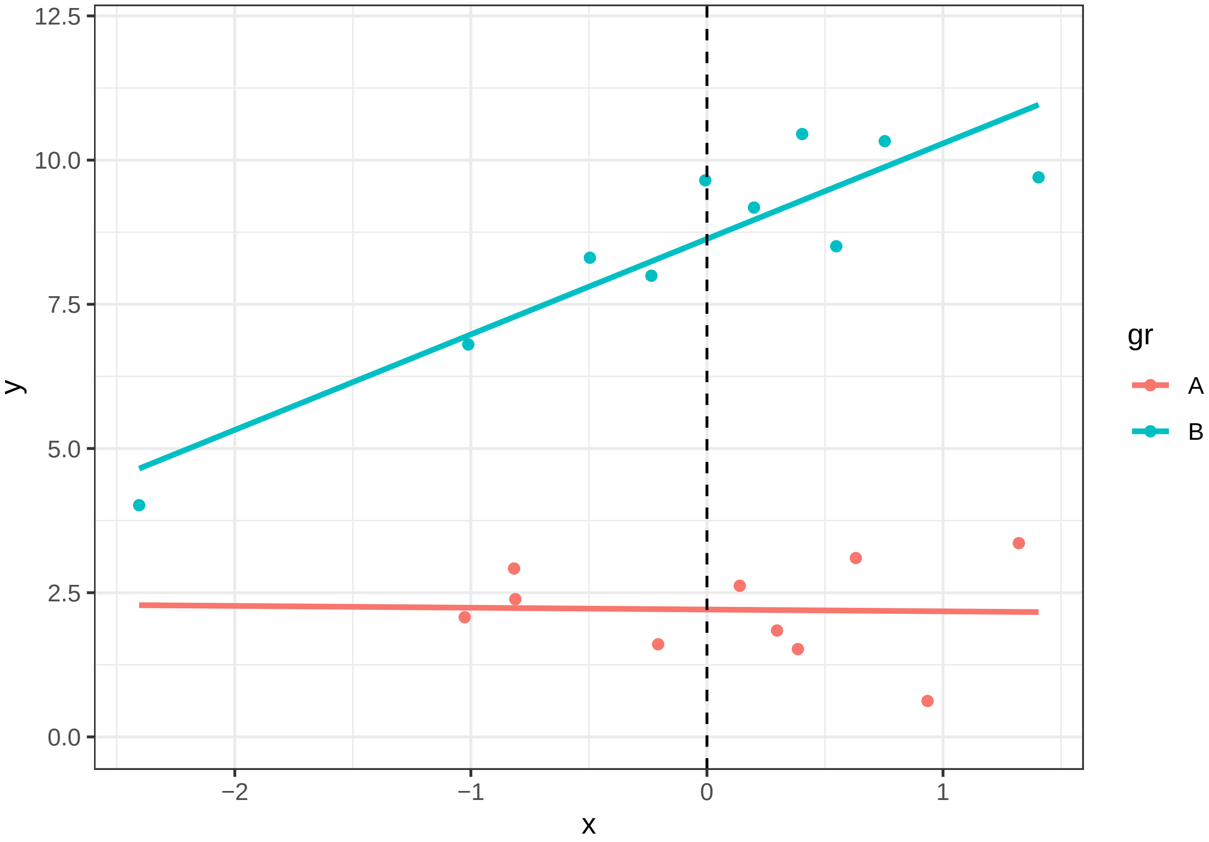
3. Model 2: Center x (same dummy coding)

```
D2 <- D
D2$x <- D$x - mean(D$x)
mod2 <- lm(y ~ gr * x, data = D2)
summary(mod2)
```

```
##
## Call:
## lm(formula = y ~ gr * x, data = D2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.55585 -0.61528 -0.00131  0.54234  1.19203
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.20810    0.27933   7.905 6.47e-07 ***
## grB          6.42543    0.39449  16.288 2.21e-11 ***
## x           -0.03137    0.37016  -0.085  0.93352
## grB:x        1.68703    0.46200   3.652  0.00215 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8778 on 16 degrees of freedom
## Multiple R-squared:  0.9481, Adjusted R-squared:  0.9384
## F-statistic: 97.49 on 3 and 16 DF,  p-value: 1.708e-10
```

```
p3 <- ggplot(D2, aes(x = x, y = y, color = gr)) +
  geom_point() +
  geom_smooth(method = "lm", fill = NA, fullrange = TRUE) +
  geom_vline(xintercept = 0, linetype = "dashed") +
  theme_bw()
print(p3)
```

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



Discover
1. Now what does `grB` mean? (Compare to plot.) Better?
2. How do p-values change for main effects? For interaction?
3. **Why** did centering `x` change the *main effect* tests, even though interaction was always in the model?

---

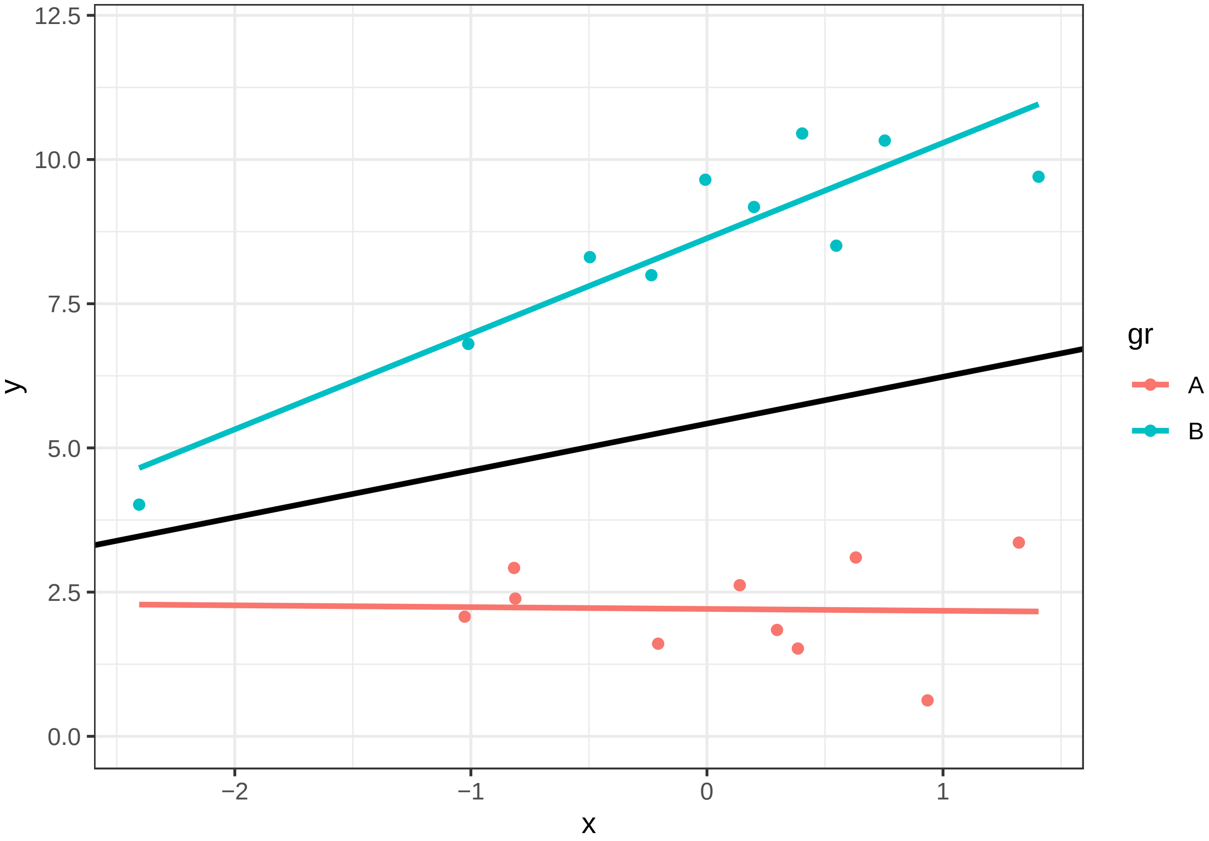
4. Model 3: Zero-sum contrasts for gr (+ centered x)

```
D3 <- D2
contrasts(D3$gr) <- contr.sum(2)
mod3 <- lm(y ~ gr * x, data = D3)
summary(mod3)
```

```
##
## Call:
## lm(formula = y ~ gr * x, data = D3)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.55585 -0.61528 -0.00131  0.54234  1.19203
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5.4208     0.1972  27.483 6.80e-15 ***
## gr1          -3.2127     0.1972 -16.288 2.21e-11 ***
## x             0.8121     0.2310   3.516  0.00287 **
## gr1:x        -0.8435     0.2310  -3.652  0.00215 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8778 on 16 degrees of freedom
## Multiple R-squared:  0.9481, Adjusted R-squared:  0.9384
## F-statistic: 97.49 on 3 and 16 DF,  p-value: 1.708e-10
```

```
p4 <- ggplot(D3, aes(x = x, y = y, color = gr)) +
  geom_point() +
  geom_smooth(method = "lm", fill = NA, fullrange = TRUE) +
  geom_abline(intercept = coef(mod3)[1] , slope = coef(mod3)[3], color = "black", linewidth = 1) +
  theme_bw()
print(p4)
```

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



Discover

1. What population does the intercept + x slope (black line) describe?
2. How is this different from Models 1–2?
3. Compare p-values across all 3 models. Which encoding gives "cleanest" tests?

---

5. Bonus: Rescale x instead of just centering

```
D4 <- D
D4$x <- scale(D$x)  # Centers AND divides by SD
mod4 <- lm(y ~ gr * x, data = D4)
summary(mod4)
```

```
##
## Call:
## lm(formula = y ~ gr * x, data = D4)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.55585 -0.61528 -0.00131  0.54234  1.19203
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.20810    0.27933   7.905 6.47e-07 ***
## grB          6.42543    0.39449  16.288 2.21e-11 ***
## x           -0.02865    0.33805  -0.085  0.93352
## grB:x        1.54069    0.42193   3.652  0.00215 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8778 on 16 degrees of freedom
## Multiple R-squared:  0.9481, Adjusted R-squared:  0.9384
## F-statistic: 97.49 on 3 and 16 DF,  p-value: 1.708e-10
```

Discover
1. Compare `mod4` coefficients to `mod2` (just centering). Why are slopes ~2x larger now?
2. Is the group difference (`grB`) at `x=0` the *same*?
3. **Pro**: What does "per 1-SD change in `x`" mean for interpretation?
4. **Con**: Does rescaling change p-values or tests? Why/why not? When would you rescale?

---

6. Your discoveries

Answer these (no peeking at solutions):
1. Why is the "main effect" of `gr` meaningless without specifying *at what* `x`?
2. How does centering `x` make main effects interpretable **despite** interaction?
3. When interactions are present, why do defaults hurt power/interpretation?
4. Rule of thumb: when to center? When to use `contr.sum()`?

Extension (groups): Fit on *your own* simulated data (change slopes/groups). Does the lesson hold?

Answer to your side question: Rescaling vs centering

Rescaling (`scale(x)` = centering + dividing by SD) has these effects:

Pros:
- Coefficients are **standardized effect sizes** ("per 1-SD change in `x`"). Great for comparing predictor strength (e.g., BMI vs age). - Group difference still at meaningful `x=0` (mean).

Cons vs pure centering:
- **No change to p-values, F-tests, or R^2** (pure linear rescaling). Tests identical to centered-only model.
- Slopes scale up by $1/\text{SD}_x$, so intercept adjusts down accordingly.
- Less intuitive if SD lacks meaning (e.g., arbitrary imaging parameter).

When to rescale: For predictor comparison or publication tables. Otherwise, centering suffices for clean main effects.