EDUC 640

Two-Way Factorial ANOVA

Chris Ives

Contents

- Plotting Means
- Recoding/Reordering Factors
- Contrasts using `emmeans`

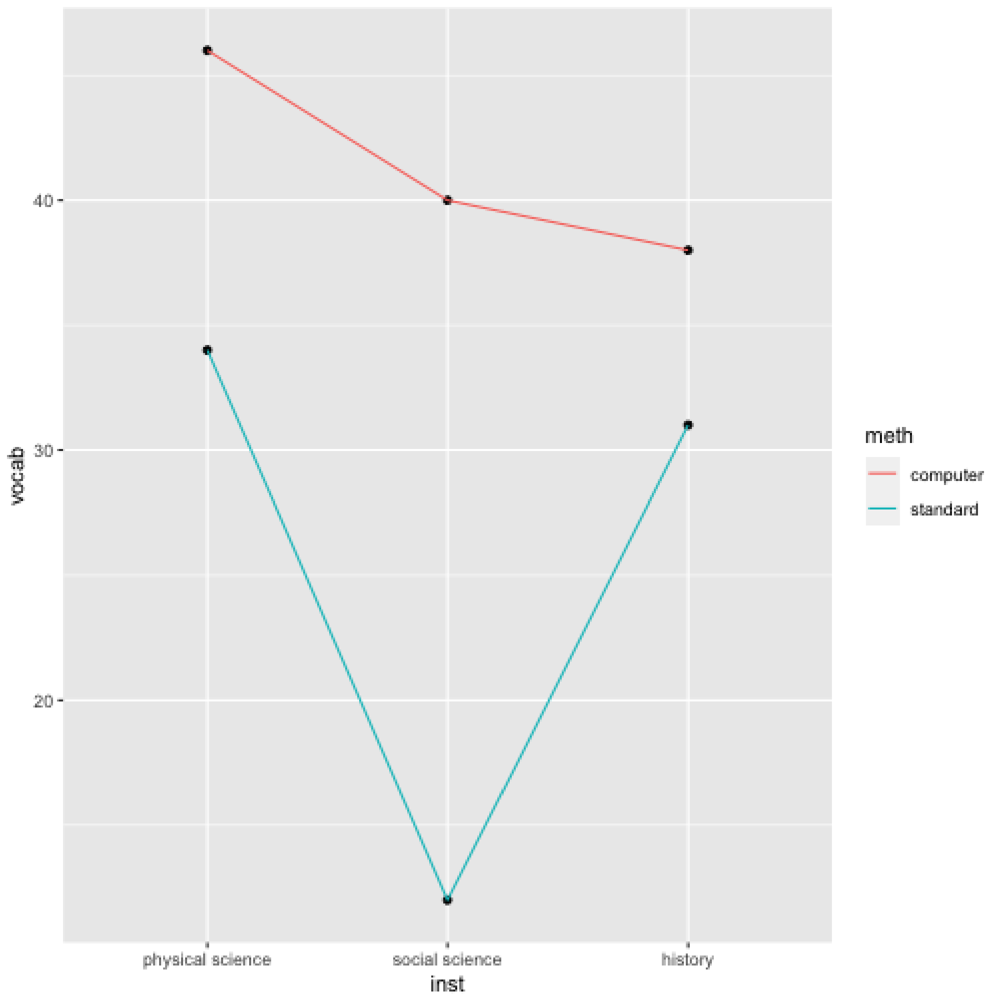
Plot Marginal Means

Here I am creating a new summary data frame. After grouping by `meth` and `inst`, I just make `vocab` equal to the mean of those groups. This dataframe is another way of getting marginal mean values.

```
means <- lb5 %>%
  group_by(inst, meth) %>%
  summarise(vocab = mean(vocab))
```

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

```r
ggplot(means, aes(x = inst, y = vocab, group = meth)) +
  geom_point() +
  geom_line(aes(color = meth))
```



Contrasts using `emmeans`

To do two–factor contrasts we will have to use the `emmeans` package. You might find this easier than the previous approaches where I had you specify your coding in a matrix. Whether you prefer this for one–way contrasts will be up to you.

Recoding Data

We have to start by combining our two factors into one factor with six levels. The level names are a little long for my taste so here I am shortening them. First check the orders of the levels so your names are assigned appropriately.

```r
levels(lb5$meth)
```

```
## [1] "computer" "standard"
```

```r
levels(lb5$meth) <- c("comp", "stan")
```

```r
levels(lb5$inst)
```

```
## [1] "physical science" "social science"   "history"
```

```r
levels(lb5$inst) <- c("phys", "soc", "hist")
```

Next I join the two factors into one with `unite`. `col =` specifies the name of the new column, followed by the columns I am joining. I chose to separate them with "_" and set `remove = FALSE` so I don't delete the old variables.

```
lb5 <- lb5 %>%
  unite(col = ivs, meth, inst, sep = "_", remove = FALSE)

head(lb5)
```

```
##   idnum vocab       ivs inst meth
## 1     1    53 comp_phys phys comp
## 2     2    49 comp_phys phys comp
## 3     3    47 comp_phys phys comp
## 4     4    42 comp_phys phys comp
## 5     5    51 comp_phys phys comp
## 6     6    34 comp_phys phys comp
```

Last bit of data prep is order them in a way that will make sense for me when I code out contrasts later. This order reflects what's in Gina's slides.

```
lb5$ivs <- ordered(lb5$ivs, c("comp_phys", "comp_soc", "comp_his

levels(lb5$ivs)
```

```
## [1] "comp_phys" "comp_soc"  "comp_hist" "stan_phys" "stan_soc"  "stan_hi
```

Emmeans

Start by specifying your model and then running emmeans on that model. Note that I am using `ivs`, which is our combined two factor variable.

```
m1 <- lm(vocab ~ ivs, data = lb5)
emm <- emmeans(m1, ~ ivs)
```

Then we will check the level order and assign a vectors to each. We have 6 levels so the vector is 6 numbers long. A `1` in the vector means I am saving the mean of that level to my object. So, the mean of "comp_phys" is saved to `A1B1`.

```
levels(lb5$ivs)
```

```
## [1] "comp_phys" "comp_soc"  "comp_hist" "stan_phys" "stan_soc"  "stan_hi
```

```
A1B1 <- c(1, 0, 0, 0, 0, 0)
A1B2 <- c(0, 1, 0, 0, 0, 0)
A1B3 <- c(0, 0, 1, 0, 0, 0)
A2B1 <- c(0, 0, 0, 1, 0, 0)
A2B2 <- c(0, 0, 0, 0, 1, 0)
A2B3 <- c(0, 0, 0, 0, 0, 1)
```

Hypothesis 1

Now we can specify our contrasts (reference slides 445–446 in Lab 5). This runs the contrasts so it prints the coding scheme.

```
contrast(emm, method = list(
  (A1B1 - (A1B2 + A1B3)/2) -
  (A2B1 - (A2B2 + A2B3)/2)
  ))
```

```
##  contrast                               estimate   SE df t.ratio p.value
##  c(1, -0.5, -0.5, -1, 0.5, 0.5)             -5.5 5.27 30 -1.043  0.3052
```

Here I name the contrast.

```
contrast(emm, method = list(
  "Hyp1" = (A1B1 - (A1B2 + A1B3)/2) -
  (A2B1 - (A2B2 + A2B3)/2)
))
```

```
##  contrast estimate   SE df t.ratio p.value
```

Hypothesis 2

Same process for our second contrast.

```
contrast(emm, method = list(
  (A1B1 - A1B2) - (A2B1 - A2B2)
  ))
```

```
##  contrast                    estimate   SE df t.ratio p.value
##  c(1, -1, 0, -1, 1, 0)            -16 6.09 30 -2.628  0.0134
```

```
contrast(emm, method = list(
  "Hyp2" = (A1B1 - A1B2) - (A2B1 - A2B2)
  ))
```

```
##  contrast estimate   SE df t.ratio p.value
##  Hyp2          -16 6.09 30 -2.628  0.0134
```

Combining Contrasts

If I was planning to put them into a markdown document, I'd probably want to write it all out with one command and output.

```
contrast(emm, method = list(
  "Hyp1" = (A1B1 - (A1B2 + A1B3)/2) -
  (A2B1 - (A2B2 + A2B3)/2),
  "Hyp2" = (A1B1 - A1B2) - (A2B1 - A2B2)
))
```

```
##  contrast estimate   SE df t.ratio p.value
##  Hyp1          -5.5 5.27 30 -1.043  0.3052
##  Hyp2         -16.0 6.09 30 -2.628  0.0134
```

Here's the process using the method I described in the Appendix of Wk1–3 slides.

```
contrast0 <- c(1, -.5, -.5, -1, .5, .5)
mat.temp <- rbind(constant = 1/2, contrast0)
mat.temp
```

```
##            [,1] [,2] [,3] [,4] [,5] [,6]
## constant    0.5  0.5  0.5  0.5  0.5  0.5
## contrast0   1.0 -0.5 -0.5 -1.0  0.5  0.5
```

```
mat <- MASS::ginv(mat.temp)
mat <- mat[ , -1]
mat
```

```
## [1]  0.3333333 -0.1666667 -0.1666667 -0.3333333  0.1666667  0.1666667
```

Remember I only specified one contrast so we only pay attention to the first coefficient (that's not the intercept).

```
m_contrasts <- lm(vocab ~ ivs, data=lb5, contrasts = list(ivs = r
m_contrasts
```

```
##
## Call:
## lm(formula = vocab ~ ivs, data = lb5, contrasts = list(ivs = mat))
##
## Coefficients:
## (Intercept)         ivs1         ivs2         ivs3         ivs4
##      33.500       -5.500       -1.019       -6.219      -24.620       -5
```