Revised Regression

Evan Dragich

Last compiled on 2021-12-16

Data Cleaning

Check out the .Rmd file for full code (omitted here for brevity).

Motivation for a new statistical method

My personal biggest concern with our methods, even before receiving reviewer comments, was the blatant violation of the independence condition for standard linear regression. I **knew** there was an alternative– something that could account for the repeated measures, or "nesting" of `blooms_level * task` within each student.

During my search, I finally found it– a two-way repeated measures ANOVA. This page has good vignettes for sample scenarios, which you can draw analogies to our data structure.

I tried to find out how to implement it, and ran into recent literature eschewing ANOVA in favor of regression. This page also pointed me to the family of linear mixed models, which extend simple linear regression by allowing for random effects, in addtion to fixed.

Random Effects

Random effects control for the repeated measures factor, by essentially allowing factors to vary along `student_id` without using up degrees of freedom to try to make sense of the results. In other words, `student_id` is a random effect in our case because we have to account for each student having a different, random intercept, before examining the global, fixed effects of `task` and `blooms_level`.

The `(1 | year/student_id)` term in these models indicates that we have to account for the random, nuisance variance caused by each student and each year/cohort before examining the effects of `blooms_level`, `task`, and their interaction. Specifically, the `/` nesting operator means that we have a first random intercept to account for variance across years, and after accounting for this variance we introduce a random intercept to further account for that particular student's variance.

Mass Analysis

Omnibus ANOVA

```
mass_lmm <- lmer(rating ~ blooms_level * task + (1 | year/student_id), data = mass)

anova(mass_lmm) %>%
  tidy()
```

```
## # A tibble: 3 x 7
##   term              sumsq meansq NumDF DenDF statistic  p.value
##   <chr>             <dbl>  <dbl> <int> <dbl>     <dbl>    <dbl>
```

```
## 1 blooms_level      182.    36.5     5 2555.      48.9 1.97e-48
## 2 task               99.4   33.1     3 2555.      44.4 6.12e-28
## 3 blooms_level:task 361.    24.0    15 2555.      32.2 5.34e-85
```

Great! Significant main effects for `task` and `blooms_level`, as well as their interaction. The next step is to break this down into pairwise comparisons. I chose to analyze pairs of `blooms_level` within each `task`; this is most analogous to our former method.

Marginal Means and Contrasts

```
mass_emm <- ref_grid(mass_lmm) %>%
  emmeans(specs = pairwise ~ blooms_level | task)

tidy(mass_emm$emmeans)
```

```
## # A tibble: 24 x 7
##    blooms_level task  estimate std.error    df statistic  p.value
##    <chr>        <chr>    <dbl>     <dbl> <dbl>     <dbl>    <dbl>
##  1 remember     lec       3.99     0.151  2.41      26.5 0.000494
##  2 understand   lec       4.24     0.151  2.39      28.2 0.000449
##  3 analyze      lec       3.34     0.151  2.43      22.1 0.000724
##  4 apply        lec       3.40     0.151  2.39      22.6 0.000763
##  5 evaluate     lec       3.06     0.152  2.46      20.2 0.000863
##  6 create       lec       2.64     0.151  2.41      17.5 0.00133
##  7 remember     icp       3.97     0.151  2.41      26.3 0.000501
##  8 understand   icp       4.14     0.151  2.41      27.4 0.000453
##  9 analyze      icp       3.84     0.151  2.39      25.5 0.000569
## 10 apply        icp       4.15     0.151  2.41      27.5 0.000450
## # ... with 14 more rows
```

```
mass_emm$contrasts %>%
  summary(infer = TRUE)
```

```
## task = lec:
##  contrast              estimate     SE   df lower.CL upper.CL t.ratio p.value
##  remember - understand -0.24681 0.1292 2555  -0.6154  0.12176  -1.910  0.3963
##  remember - analyze     0.64827 0.1299 2554   0.2777  1.01887   4.989  <.0001
##  remember - apply       0.59763 0.1292 2555   0.2291  0.96621   4.624  0.0001
##  remember - evaluate    0.93605 0.1303 2555   0.5644  1.30776   7.182  <.0001
##  remember - create      1.34831 0.1296 2554   0.9788  1.71783  10.406  <.0001
##  understand - analyze   0.89508 0.1296 2555   0.5254  1.26474   6.905  <.0001
##  understand - apply     0.84444 0.1288 2554   0.4770  1.21190   6.554  <.0001
##  understand - evaluate  1.18286 0.1300 2555   0.8121  1.55363   9.098  <.0001
##  understand - create    1.59513 0.1292 2555   1.2266  1.96370  12.342  <.0001
##  analyze - apply       -0.05064 0.1296 2555  -0.4203  0.31902  -0.391  0.9988
##  analyze - evaluate     0.28778 0.1307 2554  -0.0849  0.66049   2.202  0.2370
##  analyze - create       0.70004 0.1299 2554   0.3294  1.07064   5.387  <.0001
##  apply - evaluate       0.33842 0.1300 2555  -0.0323  0.70919   2.603  0.0968
##  apply - create         0.75068 0.1292 2555   0.3821  1.11925   5.808  <.0001
##  evaluate - create      0.41226 0.1303 2555   0.0406  0.78396   3.163  0.0197
##
## task = icp:
##  contrast              estimate     SE   df lower.CL upper.CL t.ratio p.value
##  remember - understand -0.16854 0.1296 2554  -0.5381  0.20097  -1.301  0.7848
##  remember - analyze     0.13072 0.1292 2555  -0.2379  0.49929   1.011  0.9143
```

```
##  remember - apply      -0.17978 0.1296 2554  -0.5493  0.18974  -1.388  0.7347
##  remember - evaluate    0.46265 0.1296 2555   0.0930  0.83228   3.569  0.0049
##  remember - create      1.07427 0.1299 2554   0.7037  1.44487   8.267  <.0001
##  understand - analyze   0.29926 0.1292 2555  -0.0693  0.66783   2.316  0.1880
##  understand - apply    -0.01124 0.1296 2554  -0.3807  0.35827  -0.087  1.0000
##  understand - evaluate  0.63119 0.1296 2555   0.2616  1.00082   4.870  <.0001
##  understand - create    1.24281 0.1299 2554   0.8722  1.61341   9.564  <.0001
##  analyze - apply       -0.31049 0.1292 2555  -0.6791  0.05808  -2.402  0.1556
##  analyze - evaluate     0.33193 0.1292 2554  -0.0366  0.70045   2.569  0.1054
##  analyze - create       0.94356 0.1296 2555   0.5739  1.31321   7.279  <.0001
##  apply - evaluate       0.64242 0.1296 2555   0.2728  1.01206   4.957  <.0001
##  apply - create         1.25405 0.1299 2554   0.8835  1.62464   9.650  <.0001
##  evaluate - create      0.61163 0.1300 2555   0.2409  0.98235   4.705  <.0001
##
## task = hw:
##  contrast              estimate     SE   df lower.CL upper.CL t.ratio p.value
##  remember - understand -0.06742 0.1292 2555  -0.4360  0.30115  -0.522  0.9953
##  remember - analyze     0.12174 0.1292 2554  -0.2468  0.49026   0.942  0.9355
##  remember - apply      -0.26967 0.1292 2555  -0.6382  0.09890  -2.087  0.2945
##  remember - evaluate    0.42893 0.1292 2554   0.0604  0.79745   3.319  0.0118
##  remember - create      0.96378 0.1300 2555   0.5930  1.33454   7.413  <.0001
##  understand - analyze   0.18916 0.1296 2555  -0.1805  0.55880   1.459  0.6903
##  understand - apply    -0.20225 0.1296 2554  -0.5718  0.16726  -1.561  0.6245
##  understand - evaluate  0.49636 0.1296 2555   0.1267  0.86599   3.830  0.0018
##  understand - create    1.03120 0.1303 2555   0.6595  1.40290   7.912  <.0001
##  analyze - apply       -0.39141 0.1296 2555  -0.7610 -0.02177  -3.020  0.0306
##  analyze - evaluate     0.30720 0.1296 2555  -0.0624  0.67678   2.370  0.1671
##  analyze - create       0.84204 0.1304 2555   0.4702  1.21387   6.458  <.0001
##  apply - evaluate       0.69860 0.1296 2555   0.3290  1.06824   5.390  <.0001
##  apply - create         1.23345 0.1303 2555   0.8618  1.60514   9.464  <.0001
##  evaluate - create      0.53485 0.1304 2555   0.1630  0.90667   4.102  0.0006
##
## task = pbl:
##  contrast              estimate     SE   df lower.CL upper.CL t.ratio p.value
##  remember - understand -0.30337 0.0916 2554  -0.5647 -0.04209  -3.311  0.0121
##  remember - analyze    -0.69621 0.0919 2555  -0.9583 -0.43413  -7.576  <.0001
##  remember - apply      -0.75843 0.0916 2554  -1.0197 -0.49714  -8.278  <.0001
##  remember - evaluate   -0.75281 0.0916 2554  -1.0141 -0.49153  -8.217  <.0001
##  remember - create     -0.95443 0.0914 2556  -1.2151 -0.69374 -10.441  <.0001
##  understand - analyze  -0.39284 0.0919 2555  -0.6549 -0.13076  -4.275  0.0003
##  understand - apply    -0.45506 0.0916 2554  -0.7163 -0.19377  -4.967  <.0001
##  understand - evaluate -0.44944 0.0916 2554  -0.7107 -0.18815  -4.906  <.0001
##  understand - create   -0.65106 0.0914 2556  -0.9117 -0.39037  -7.123  <.0001
##  analyze - apply       -0.06222 0.0919 2555  -0.3243  0.19986  -0.677  0.9845
##  analyze - evaluate    -0.05660 0.0919 2555  -0.3187  0.20548  -0.616  0.9899
##  analyze - create      -0.25822 0.0917 2556  -0.5197  0.00326  -2.816  0.0552
##  apply - evaluate       0.00562 0.0916 2554  -0.2557  0.26690   0.061  1.0000
##  apply - create        -0.19600 0.0914 2556  -0.4567  0.06468  -2.144  0.2649
##  evaluate - create     -0.20162 0.0914 2556  -0.4623  0.05907  -2.206  0.2353
##
## Degrees-of-freedom method: satterthwaite
## Confidence level used: 0.95
## Conf-level adjustment: tukey method for comparing a family of 6 estimates
## P value adjustment: tukey method for comparing a family of 6 estimates
```

(Surprisingly, there don't exist any simple packages to transform this output into a neater table–I think we'd have to do it ourselves.)

The first table is the estimated marginal means (EMM), which enhances bare-bones descriptive statistics by accounting for imbalances in data. This is huge, because our mass data has double the `pbl` observations of the other categories. This method also helps with the imbalance from missing data, but that is a trivial concern compared to the double-PBL issue.

The second table output contains the pairwise contrasts between each level for a particular task, with the $\alpha = 0.05$ p-value, associated 95% confidence interval, and Tukey family-wise adjustment.

My Thoughts

This is messier than our previous output, but displays similar effects which lead to similar interpretations. It isn't nearly as parsimonious as "these 3 coefficients are negative, but PBL is the only positive one!" However, examine the `lec` contrasts and you can see ample comparisons which estimate `Remember` and `Understand` well below the higher levels, with significance. Conversely, these same comparisons have negative estimated effects for `pbl`, indicating a significant difference in the opposite direction. Middle Bloom's Levels are hazier to tease apart, as they were before, but you can see the clear stratification between `lec` -> `hw/icp` -> `pbl`, in my opinion.

Kinetics Analysis

Omnibus ANOVA

Here is the same procedure for kinetics. Starting with model fitting and the omnibus ANOVA:

```
kinetics_lmm <- lmer(rating ~ blooms_level * task + (1 | year/student_id), data = kinetics)

anova(kinetics_lmm) %>%
  tidy()
```

```
## # A tibble: 3 x 7
##   term              sumsq meansq NumDF DenDF statistic  p.value
##   <chr>             <dbl>  <dbl> <int> <dbl>     <dbl>    <dbl>
## 1 blooms_level      135.   27.0      5 2028.      37.5 7.10e-37
## 2 task               86.8  28.9      3 2028.      40.2 3.23e-25
## 3 blooms_level:task 113.    7.51    15 2028.      10.4 1.36e-24
```

Estimated Marginal Means and Contrasts

```
kinetics_emm <- ref_grid(kinetics_lmm) %>%
  emmeans(specs = pairwise ~ blooms_level | task)

tidy(kinetics_emm$emmeans)
```

```
## # A tibble: 24 x 7
##    blooms_level task  estimate std.error    df statistic p.value
##    <chr>        <chr>    <dbl>     <dbl> <dbl>     <dbl>   <dbl>
##  1 remember     lec       3.86     0.180  1.73      21.5 0.00422
##  2 understand   lec       3.86     0.180  1.72      21.5 0.00432
##  3 analyze      lec       3.37     0.180  1.73      18.7 0.00534
##  4 apply        lec       3.39     0.180  1.72      18.8 0.00542
##  5 evaluate     lec       3.07     0.180  1.72      17.1 0.00639
##  6 create       lec       2.71     0.180  1.73      15.0 0.00779
```

```
##  7 remember     icp       3.94     0.180  1.72      21.9 0.00418
##  8 understand   icp       4.05     0.180  1.73      22.5 0.00388
##  9 analyze      icp       3.75     0.180  1.73      20.8 0.00444
## 10 apply        icp       3.94     0.180  1.73      21.9 0.00408
## # ... with 14 more rows
```

```
kinetics_emm$contrasts %>%
  summary(infer = TRUE)
```

```
## task = lec:
##  contrast               estimate    SE   df lower.CL upper.CL t.ratio p.value
##  remember - understand -2.70e-06 0.127 2028 -0.36195  0.36194   0.000  1.0000
##  remember - analyze     4.90e-01 0.127 2028  0.12747  0.85346   3.854  0.0017
##  remember - apply       4.78e-01 0.127 2028  0.11583  0.83972   3.765  0.0024
##  remember - evaluate    7.89e-01 0.127 2028  0.42694  1.15083   6.217  <.0001
##  remember - create      1.16e+00 0.127 2027  0.79444  1.52016   9.098  <.0001
##  understand - analyze   4.90e-01 0.127 2028  0.12858  0.85236   3.866  0.0016
##  understand - apply     4.78e-01 0.127 2027  0.11694  0.83861   3.777  0.0023
##  understand - evaluate  7.89e-01 0.127 2027  0.42805  1.14973   6.236  <.0001
##  understand - create    1.16e+00 0.127 2028  0.79536  1.51925   9.121  <.0001
##  analyze - apply       -1.27e-02 0.127 2028 -0.37458  0.34920  -0.100  1.0000
##  analyze - evaluate     2.98e-01 0.127 2028 -0.06347  0.66031   2.352  0.1740
##  analyze - create       6.67e-01 0.127 2028  0.30384  1.02984   5.240  <.0001
##  apply - evaluate       3.11e-01 0.127 2027 -0.04973  0.67195   2.459  0.1369
##  apply - create         6.80e-01 0.127 2028  0.31758  1.04147   5.355  <.0001
##  evaluate - create      3.68e-01 0.127 2028  0.00647  0.73036   2.903  0.0433
##
## task = icp:
##  contrast               estimate    SE   df lower.CL upper.CL t.ratio p.value
##  remember - understand -1.13e-01 0.127 2028 -0.47518  0.24871  -0.892  0.9484
##  remember - analyze     1.90e-01 0.127 2028 -0.17180  0.55208   1.498  0.6653
##  remember - apply      -8.71e-04 0.127 2028 -0.36282  0.36107  -0.007  1.0000
##  remember - evaluate    5.89e-01 0.127 2027  0.22805  0.94973   4.655  0.0001
##  remember - create      9.99e-01 0.127 2028  0.63718  1.36107   7.874  <.0001
##  understand - analyze   3.03e-01 0.127 2027 -0.05949  0.66623   2.385  0.1620
##  understand - apply     1.12e-01 0.127 2027 -0.25050  0.47522   0.883  0.9506
##  understand - evaluate  7.02e-01 0.127 2028  0.34018  1.06406   5.533  <.0001
##  understand - create    1.11e+00 0.127 2027  0.74950  1.47522   8.744  <.0001
##  analyze - apply       -1.91e-01 0.127 2027 -0.55387  0.17185  -1.502  0.6633
##  analyze - evaluate     3.99e-01 0.127 2028  0.03680  0.76069   3.142  0.0210
##  analyze - create       8.09e-01 0.127 2027  0.44613  1.17185   6.359  <.0001
##  apply - evaluate       5.90e-01 0.127 2028  0.22782  0.95170   4.648  0.0001
##  apply - create         1.00e+00 0.127 2027  0.63714  1.36286   7.861  <.0001
##  evaluate - create      4.10e-01 0.127 2028  0.04830  0.77218   3.233  0.0157
##
## task = hw:
##  contrast               estimate    SE   df lower.CL upper.CL t.ratio p.value
##  remember - understand  2.01e-02 0.128 2028 -0.34384  0.38413   0.158  1.0000
##  remember - analyze     1.15e-01 0.127 2028 -0.24821  0.47779   0.902  0.9461
##  remember - apply      -1.77e-01 0.127 2028 -0.54034  0.18566  -1.394  0.7310
##  remember - evaluate    3.72e-01 0.127 2028  0.00970  0.73349   2.929  0.0402
##  remember - create      8.24e-01 0.128 2028  0.45950  1.18764   6.453  <.0001
##  understand - analyze   9.46e-02 0.128 2028 -0.26929  0.45857   0.742  0.9767
##  understand - apply    -1.97e-01 0.128 2028 -0.56142  0.16644  -1.548  0.6331
##  understand - evaluate  3.51e-01 0.127 2028 -0.01158  0.71446   2.761  0.0642
```

```
##  understand - create    8.03e-01 0.128 2028  0.43842  1.16842   6.279  <.0001
##  analyze - apply       -2.92e-01 0.127 2027 -0.65499  0.07072  -2.296  0.1958
##  analyze - evaluate     2.57e-01 0.127 2028 -0.10514  0.61875   2.024  0.3290
##  analyze - create       7.09e-01 0.128 2028  0.34485  1.07271   5.555  <.0001
##  apply - evaluate       5.49e-01 0.127 2028  0.18699  0.91088   4.326  0.0002
##  apply - create         1.00e+00 0.128 2028  0.63698  1.36484   7.845  <.0001
##  evaluate - create      4.52e-01 0.127 2028  0.08896  0.81500   3.551  0.0053
##
## task = pbl:
##  contrast               estimate    SE   df lower.CL upper.CL t.ratio p.value
##  remember - understand -5.62e-02 0.127 2027 -0.41904  0.30668  -0.442  0.9979
##  remember - analyze    -4.16e-01 0.127 2027 -0.77859 -0.05287  -3.268  0.0140
##  remember - apply      -4.38e-01 0.127 2027 -0.80106 -0.07534  -3.445  0.0077
##  remember - evaluate   -3.26e-01 0.127 2027 -0.68870  0.03702  -2.561  0.1074
##  remember - create     -4.03e-01 0.127 2028 -0.76532 -0.04143  -3.179  0.0187
##  understand - analyze  -3.60e-01 0.127 2027 -0.72241  0.00331  -2.826  0.0538
##  understand - apply    -3.82e-01 0.127 2027 -0.74488 -0.01916  -3.003  0.0323
##  understand - evaluate -2.70e-01 0.127 2027 -0.63252  0.09320  -2.120  0.2773
##  understand - create   -3.47e-01 0.127 2028 -0.70914  0.01475  -2.736  0.0687
##  analyze - apply       -2.25e-02 0.127 2027 -0.38533  0.34039  -0.177  1.0000
##  analyze - evaluate     8.99e-02 0.127 2027 -0.27297  0.45275   0.707  0.9812
##  analyze - create       1.24e-02 0.127 2028 -0.34959  0.37430   0.097  1.0000
##  apply - evaluate       1.12e-01 0.127 2027 -0.25050  0.47522   0.883  0.9506
##  apply - create         3.48e-02 0.127 2028 -0.32712  0.39677   0.274  0.9998
##  evaluate - create     -7.75e-02 0.127 2028 -0.43948  0.28441  -0.611  0.9903
##
## Degrees-of-freedom method: satterthwaite
## Confidence level used: 0.95
## Conf-level adjustment: tukey method for comparing a family of 6 estimates
## P value adjustment: tukey method for comparing a family of 6 estimates
```

My thoughts

Very analogous results again. `pbl` in particular **definitely** doesn't look as pretty as our original output.

But, I think we can interpret each of these significant pairwise comparisons in a much more statistically sound way, which will please reviewers.

Assumptions for this mixed model

Final points: assumptions. This paper points out that even blatant violations are mostly fine, with the caveats of (1) no bimodial distributions and (2) noticeable skew and heteroscedasticity (which we have much of if I remember correctly) can lead to imprecise estimates. This is reflected in many wide confidence intervals in our data, and is an effect of both potential underlying skew/heteroscedasticity, but also more simply the Likert nature of our data.

I'll have more robust assumptions work added soon, but this is a great initial proof of concept and starting point for revisions.