STA303H1S/1002H Module 3

Class code demo

Prof. Liza Bolton

Solutions released February 16, 2022

Contents

I grow rows and rows of roses
Flor de mayo, by the mile

Useful shortcuts in RStudio

Link

Rose farm

Note: I have updated the 'story' a bit, based on what we did in the Team Up! activity. You can see the asynch version of the activity here

There are 5 greenhouses on your rose farm, North, East, South, West and Central. They are at different levels of modernization as it can be very expensive to replace and update irrigation systems and other fixtures. You hope to be making some general improvements.

There are 4 varieties of red rose that your farm specializes in: Devotion, Checkmate, Wanted, and Hearts.

Within each greenhouse, there are 32 plots. In preparation for the 2021 Valentine's Day, you planted equal plots of each of the special varieties in each greenhouse. Then the flowers were cut and sold and it was recorded how much net profit was made per plot.

Now, you want to understand your data from last year and make predictions about your 2022 profits. (Ignoring inflation, which probably shouldn't do, eh? Yikes. The economy, amirite?)

Grow roses

This setup chunk load the function that will 'grow' our roses (and profit data) for us. For this class demo, we will all have the same rose data, but you will have different data to your peers in the portfolio.

We don't really need to do the seed set up and everything here, but I wanted to give you a useful example for your portfolio. Be very careful about relative paths to files. MarkUs autograding will be set up based on the same structure as the JupyterHub and will require relative paths.

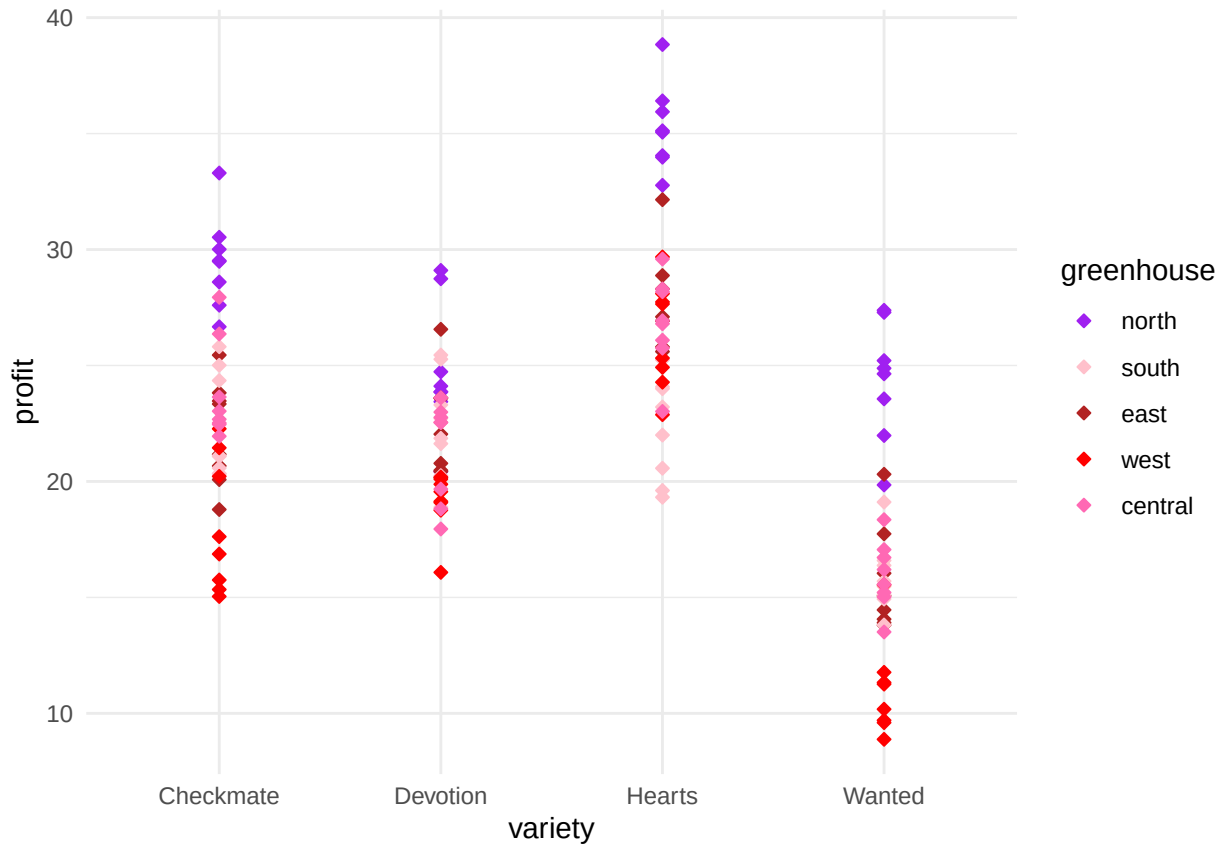
```
# This chunk is named 'setup'

# I have set message=FALSE so I don't get all of tidyverses message on load
library(tidyverse)


# Suppose the last three digits of my ID number were 788
last3digplus <- 100 + 788

# Runs an R file for us so we can access out functtion.
# This file path is short, just the file name itself, as it's relative to our RMD
source("grow-roses.R")

# Function we loaded
roses <- grow_roses(seed = last3digplus)

# Glimpse data
glimpse(roses)
```

```
## Rows: 160
## Columns: 3
## $ greenhouse <chr> "north", "north", "north", "north", "north", "north", "nort~
## $ variety    <chr> "Devotion", "Devotion", "Devotion", "Devotion", "Devotion",~
## $ profit     <dbl> 23.44, 20.42, 23.86, 28.74, 29.10, 24.73, 23.86, 24.11, 26.~
```

Visualize

- This task is a mix of what was asked in the Team Up! and what I demoed in class. The "Valentine's Day colour scheme" I added would not be one I recommend for future data viz! It was just to have some fun and show you how you can tell R colours by names it knows, as well as using hex codes as you've seen elsewhere.
- Note: `fct_relevel` doesn't change the order of your raw data, but it does give R information about how you would like to treat that variable in future plots/models. I know a few folks were trying to check the facotr ordering worked by looking at their data, but that is not the right place to look, if the order is correct in your plot (or the right reference level is used in a model), then you know you've got it right.

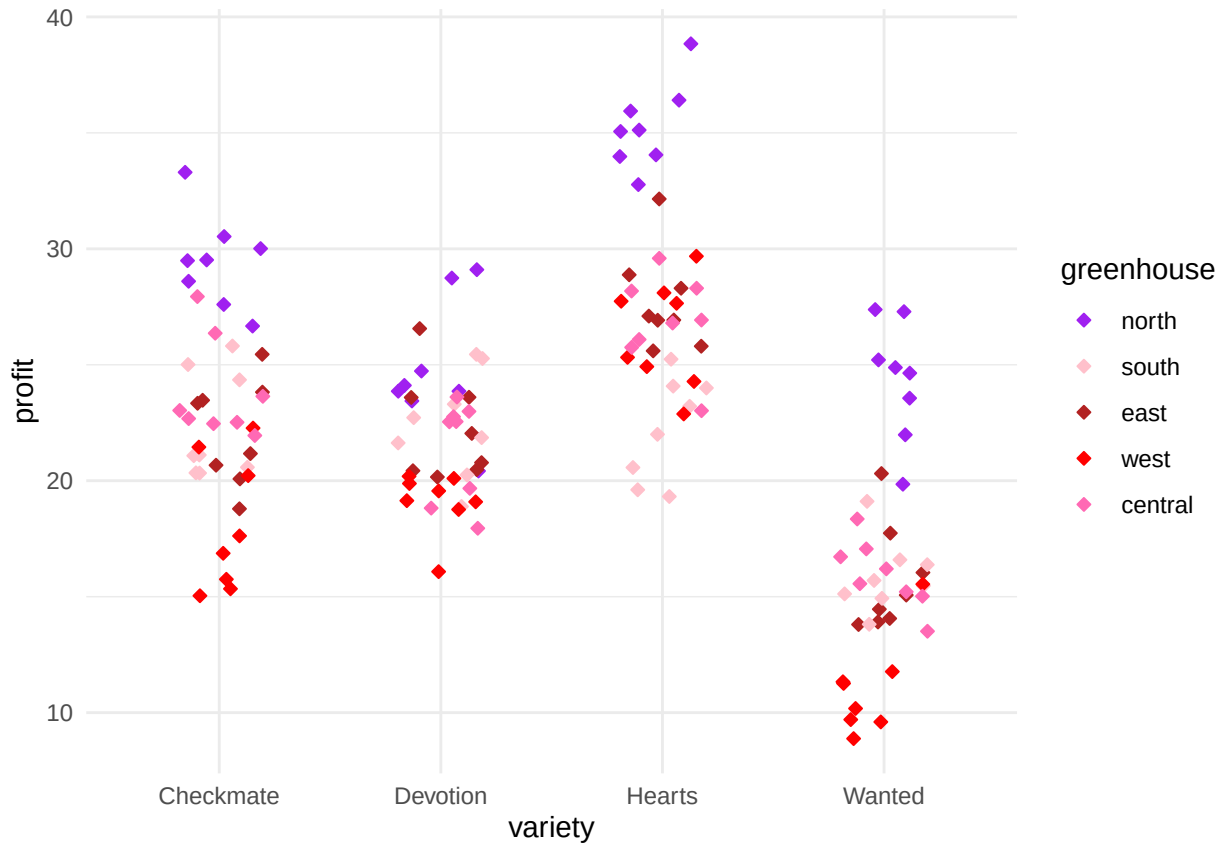
```
# Plot roses
roses %>%
  # This will change the order
  mutate(greenhouse = fct_relevel(greenhouse, "north", "south", "east", "west", "central" )) %>%
  ggplot(aes(x = variety, y = profit, color = greenhouse, fill = greenhouse)) +
  scale_color_manual(values = c(central = "hotpink", east = "firebrick",
                                north = "purple", south = "pink", west = "red")) +
  scale_fill_manual(values = c(central = "hotpink", east = "firebrick",
                                north = "purple", south = "pink", west = "red")) +
  # Make the points a girl's best friend
  geom_point(pch = 23) +
  theme_minimal()
```



Plot roses AND do the jitterbug

- In class we also looked a replacing `geom_point()` with `geom_jitter()`.
  - If you wanted to spread your points out horizontally, but not add any vertical noise (vertical is the only meaningful diffference as long as we don't overlap the other categorical levels), you could do as follows.

```
# Plot roses
roses %>%
  # This will change the order
  mutate(greenhouse = fct_relevel(greenhouse, "north", "south", "east", "west", "central" )) %>%
  ggplot(aes(x = variety, y = profit, color = greenhouse, fill = greenhouse)) +
  scale_color_manual(values = c(central = "hotpink", east = "firebrick",
                                north = "purple", south = "pink", west = "red")) +
  scale_fill_manual(values = c(central = "hotpink", east = "firebrick",
                                north = "purple", south = "pink", west = "red")) +
  # height = 0 means no vetical jitter, width = 0.2 allows that much horizontal jitter +/-
  geom_jitter(pch = 23, width = 0.2, height = 0) +
  theme_minimal()
```



Teaching and learning world

This part of the demo is like Statdew Valley

- Uses ONLY `lm`
- We can ONLY do this for balanced designs

IJK, it's easy as 123, as simple as do re mi

Suppose an appropriate model for this data is:

$$y_{ijk} = \mu + \alpha_i + b_j + (\alpha b)_{ij} + \epsilon_{ijk}$$

where:

- μ is the grand mean profit of strawberries, in thousands of dollars, across all the data.
- α_i is the fixed effect of variety.
- b_j is the random effect of greenhouse, $N(0, \sigma_b^2)$.
- $(\alpha b)_{ij}$ is in interaction between variety and greenhouse, $N(0, \sigma_{\alpha b}^2)$.
- ϵ_{ijk} is the error term, $N(0, \sigma^2)$.

Alternative title: abcdefgh____lmnopqrstuvwxyz ;)

I: How many varieties? You should know this from the story already, but you can verify with your data that nothing is missing/has been added.

```
# There are LOTS of ways to do this!
# Here is one:
length(unique(roses$variety))
```

```
## [1] 4
```

J: How many greenhouses? You should know this from the story already, but you can verify with your data that nothing is missing/has been added.

```
# Here is another way
levels(factor(roses$greenhouse))
```

```
## [1] "central" "east"    "north"   "south"   "west"
```

```
# or to get the number exactly

length(levels(factor(roses$greenhouse)))
```

```
## [1] 5
```

```
roses %>%
  group_by(greenhouse, variety) %>%
  count()
```

K: How many observations in each combination of greenhouse and variety?

```
## # A tibble: 20 x 3
## # Groups:   greenhouse, variety [20]
##    greenhouse variety       n
##    <chr>      <chr>     <int>
##  1 central    Checkmate     8
##  2 central    Devotion      8
##  3 central    Hearts        8
##  4 central    Wanted        8
##  5 east       Checkmate     8
##  6 east       Devotion      8
##  7 east       Hearts        8
##  8 east       Wanted        8
##  9 north      Checkmate     8
## 10 north      Devotion      8
## 11 north      Hearts        8
## 12 north      Wanted        8
## 13 south      Checkmate     8
## 14 south      Devotion      8
## 15 south      Hearts        8
## 16 south      Wanted        8
## 17 west       Checkmate     8
## 18 west       Devotion      8
## 19 west       Hearts        8
## 20 west       Wanted        8
```

```
# or in even shorter code
count(roses, greenhouse, variety)
```

```
## # A tibble: 20 x 3
##    greenhouse variety       n
##    <chr>      <chr>     <int>
##  1 central    Checkmate     8
##  2 central    Devotion      8
##  3 central    Hearts        8
##  4 central    Wanted        8
##  5 east       Checkmate     8
##  6 east       Devotion      8
##  7 east       Hearts        8
##  8 east       Wanted        8
##  9 north      Checkmate     8
## 10 north      Devotion      8
## 11 north      Hearts        8
## 12 north      Wanted        8
## 13 south      Checkmate     8
## 14 south      Devotion      8
```

```
## 15 south      Hearts        8
## 16 south      Wanted        8
## 17 west       Checkmate     8
## 18 west       Devotion      8
## 19 west       Hearts        8
## 20 west       Wanted        8
```

There are 8 plots (our unit of observation), in each combination of greenhouse and variety.

Yeah, I work in modelling

```
# Fit a model with profit as the response and the main effects
# and interaction of variety and greenhouse as the explanatory variables
roses_int_mod <- lm(profit ~ variety*greenhouse, data = roses)

# This is the code from question 3 that
# the data to get the average profit for each variety and greenhouse combination.
agg_int <- roses %>%
  group_by(greenhouse, variety) %>%
  summarize(profit_avg_int = mean(profit), .groups = "drop")
agg_int
```

```
## # A tibble: 20 x 3
##    greenhouse variety   profit_avg_int
##    <chr>      <chr>              <dbl>
##  1 central    Checkmate           23.8
##  2 central    Devotion            21.4
##  3 central    Hearts              26.8
##  4 central    Wanted              16.0
##  5 east       Checkmate           22.1
##  6 east       Devotion            22.2
##  7 east       Hearts              27.7
##  8 east       Wanted              15.7
##  9 north      Checkmate           29.5
## 10 north      Devotion            24.8
## 11 north      Hearts              35.3
## 12 north      Wanted              24.3
## 13 south      Checkmate           22.3
## 14 south      Devotion            22.4
## 15 south      Hearts              22.3
## 16 south      Wanted              15.9
## 17 west       Checkmate           18.1
## 18 west       Devotion            19.1
## 19 west       Hearts              26.3
## 20 west       Wanted              11.0
```

```
# Fit a linear model model using the agg_int data that has
# profit_avg_int as the response and variety and greenhouse as main effects.
# No interaction.
roses_agg_mod<- lm(profit_avg_int ~ variety + greenhouse, data = agg_int)
```

```r
# This code aggregates the data to find the average profit for each greenhouse
agg_greenhouse <- roses %>%
  group_by(greenhouse) %>%
  summarize(profit_avg_greenhouse = mean(profit), .groups = "drop")
agg_greenhouse
```

```
## # A tibble: 5 x 2
##   greenhouse profit_avg_greenhouse
##   <chr>                      <dbl>
## 1 central                     22.0
## 2 east                        21.9
## 3 north                       28.5
## 4 south                       20.7
## 5 west                        18.6
```

```r
# Fit a INTERCEPT ONLY linear model using the agg_greenhouse data
# that has profit_avg_greenhouse as the response.
roses_greenhouse_mod <- lm(profit_avg_greenhouse ~ 1, data = agg_greenhouse)


var_combo <- summary(roses_agg_mod)$sigma^2 - (summary(roses_int_mod)$sigma^2)/8

var_greenhouse <- summary(roses_greenhouse_mod)$sigma^2 - (summary(roses_agg_mod)$sigma^2)/4

var_resid <- summary(roses_int_mod)$sigma^2

# I = 4 (varieties)
# J = 5 (greenhouses)
# K = 8 (obs of variety/greenhouse)
```

Sources of variance table (I've fancied up this table a little, just for fun)

After accounting for our fixed effects, our remaining variability in profit can be explained by:

```
tibble(Source = c("Interaction between greenhouse and variety",
                  "Greenhouse-to-greenhouse differences",
                  "Unexplained/residual"),
       `Variance of random effect` = round(c(var_combo, var_greenhouse, var_resid),2),
  `Variance explained` = c(
    var_combo / (var_combo + var_greenhouse + var_resid),
    var_greenhouse / (var_combo + var_greenhouse + var_resid),
    var_resid / (var_combo + var_greenhouse + var_resid))) %>%
  # Write as percentages
  mutate( `Variance explained (%)` = round( `Variance explained`*100, 0)) %>%
  # remove column we don't need any more
  select(-`Variance explained`) %>%
  # changing the percentages to be right aligned
  knitr::kable(align = c("l", "r"))
```

Source	Variance of random effect	Variance explained (%)
Interaction between greenhouse and variety	3.60	17
Greenhouse-to-greenhouse differences	12.50	60
Unexplained/residual	4.75	23

Real world

One model that does all the work from above for us!

```
mod <- lme4::lmer(profit ~ variety + (1|greenhouse) + (1|greenhouse:variety), data = roses)
summary(mod)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: profit ~ variety + (1 | greenhouse) + (1 | greenhouse:variety)
##    Data: roses
##
## REML criterion at convergence: 742.1
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.18587 -0.74947 -0.08427  0.58329  2.09372
##
## Random effects:
##  Groups              Name        Variance Std.Dev.
##  greenhouse:variety (Intercept)  3.600   1.897
##  greenhouse         (Intercept) 12.504   3.536
##  Residual                        4.751   2.180
## Number of obs: 160, groups:  greenhouse:variety, 20; greenhouse, 5
##
## Fixed effects:
##                 Estimate Std. Error t value
## (Intercept)       23.156      1.827  12.672
## varietyDevotion   -1.182      1.295  -0.913
## varietyHearts      4.521      1.295   3.491
## varietyWanted     -6.577      1.295  -5.078
##
## Correlation of Fixed Effects:
##             (Intr) vrtyDv vrtyHr
## varietyDvtn -0.354
## varietyHrts -0.354  0.500
## varietyWntd -0.354  0.500  0.500
```

Interpretation

Let's actually start by asking for confidence intervals for all the random effect variances and the coefficients for the fixed effects. (This might be the first time you're seeing `confint`).

```
confint(mod)
```

```
## Computing profile confidence intervals ...

##                    2.5 %    97.5 %
## .sig01           1.079833  2.639315
## .sig02           1.724281  7.115840
## .sigma           1.947260  2.462163
## (Intercept)     19.423481 26.889517
```

```
## varietyDevotion -3.606423  1.241423
## varietyHearts    2.097327  6.945173
## varietyWanted   -9.000923 -4.153077
```

We could additionally do some post-hoc comparisons to make some sensible claims about out fixed effects, but from a quick look, it seems like Hearts is the variety producing the most profit. The reference level is Checkmate, so we might say something like, with 95% confidence, we can claim that the true profit from a plot of Hearts variety roses is about $2,100 to $6,900 dollars higher than for the Checkmate variety. (No idea how appropriate these numbers are for the real rose market, remember that before you make roses the next stonks craze).

Additionally, we can see that the estimated stanard devitions (note that we've mostly been working with sigma squared, but these are plain sigmas in the same order as the model output above) for all our random effects have 95% CIs that are not super close to 0, so that makes us think they are all sensible to include. BUT we can check this with some likelihood ratio tests, too.

LRTs

```
# Simplest linear model, no random effects
simple_lm <- lm(profit ~ variety, roses)

# Random slope but no interactions
# Note, REML = TRUE is the default setting if we don't say anything else
main_lmm <- lme4::lmer(profit ~ variety + (1|greenhouse), data = roses)

# Full model, most complicated we can fit with our data (same as mod above)
full_lmm <- lme4::lmer(profit ~ variety +
                         # random intercept for each greenhouse
                         (1|greenhouse) +
                         # random intercept for each combination of greenhouse and variety
                         (1|greenhouse:variety), data = roses)


# This will give me a message that I am comparing  a lm and lmer model
# That is okay, because that is what I want to do as my lm is NESTED in my lmer
lmtest::lrtest(simple_lm, main_lmm)
```

```
## Warning in modelUpdate(objects[[i - 1]], objects[[i]]): original model was of
## class "lm", updated model is of class "lmerMod"
```

```
## Likelihood ratio test
##
## Model 1: profit ~ variety
## Model 2: profit ~ variety + (1 | greenhouse)
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   5 -456.07
## 2   6 -389.03  1 134.09  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Compare the two LMMS
lmtest::lrtest(main_lmm, full_lmm)
```

```
## Likelihood ratio test
##
## Model 1: profit ~ variety + (1 | greenhouse)
## Model 2: profit ~ variety + (1 | greenhouse) + (1 | greenhouse:variety)
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   6 -389.03
## 2   7 -371.04  1 35.987  1.986e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Put all three models in the function?
lmtest::lrtest(simple_lm, main_lmm, full_lmm)
```

Power move: all three at once?

```
## Warning in modelUpdate(objects[[i - 1]], objects[[i]]): original model was of
## class "lm", updated model is of class "lmerMod"

## Warning in modelUpdate(objects[[i - 1]], objects[[i]]): original model was of
## class "lm", updated model is of class "lmerMod"
```

```
## Likelihood ratio test
##
## Model 1: profit ~ variety
## Model 2: profit ~ variety + (1 | greenhouse)
## Model 3: profit ~ variety + (1 | greenhouse) + (1 | greenhouse:variety)
##   #Df  LogLik Df   Chisq Pr(>Chisq)
## 1   5 -456.07
## 2   6 -389.03  1 134.086  < 2.2e-16 ***
## 3   7 -371.04  1  35.987  1.986e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Here we can see that we get two comparisons, Model 1 and Model 2, and then Model 2 and Model 3.

In each case we have a very small p value (p <.001), and so can reject the null hypothesis that the simpler model explains the data as well as the more complicated model. The simpler model is the one nested in the other model. Make sure you understand the idea of nesting models! It will be useful going forwards.

So, our overall conclusion? Of the models we've explored, the one with the random intercepts for the interaction of greenhouse and variety is the most appropriate model. This makes sense because we've seen the interaction does explain a good chunk of our variaiblity in profit after fitting fixed effects. That said, the differences in our greenhouses themselves explain the most of this variability in profit. Our story tells us some of the greenhouses are kidn of run down while others are in better repair, and this would be consistent with this outcome, assuming repair/quality of the greenhouse does have an impact on the roses grown in it.