Even More Interaction Models

2017-11-26

Introduction and Research Question

In this set of notes, we will focus on interaction effects between two categorical predictors. We will use the *iron-deficiency.csv* data in this set of notes. In particular, we are going to examine the question of whether cooking pots made of different materials explains variation in iron content

In less-developed countries, malnutrition is an ongoing problem. Iron-deficiency aneemia, one of the most common forms of malnutrition, affects about 50% of women and children and about 25% of men. The effects of anemia have obvious implications for the health and welfare of a country's populace and can also lead to economic reprecussions. As such, many countries are interested in cost-effective and convenient strategies to alleviate this problem.

Adish et al. hypothesized that the use of iron cooking pots might be one potential solution, since cooking in an iron pot would leach iron into foods prepared in them. (Iron cooking pots were used routinely in many developing countries, but were generally abandoned in favor of cheaper and lighter aluminium pots.). What is the effect of cooking pot material on the iron content of cooked food? To study this, researchers cooked each of three typical Ethiopian meals—Shiro Wet' (Legumes), Yesiga Wet' (Meat), and Ye-atkilt Allych'a (Vegetables)—in an aluminium pot, a clay pot, and an iron pot. All dishes were prepared according to specified recipes from the Ethiopian Nutritional Institute. The meals were then analysed for the total iron leftover in the cooked food for absorption by the human body.

The dataset, *iron-deficiency.csv*, includes the following variables for 200 4th graders:

- `id`: Laboratory sample ID
- `iron`: The total iron leftover for absorption by the human body after cooking (as a percentage of total iron)
- `pot`: Categorical variable representing the type of cooking pot used (`Aluminum`; `Clay`; or `Iron`)
- `food`: Categorical variable representing the type of food (`Legumes` = Shiro Wet'; `Meat` = Yesiga Wet'; and `Vegetables` = Ye-atkilt Allych'a)

The data come from Adish, A. A., Esrey, S. A., Gyorkos, T. W., Jean-Baptiste, J., & Rojhani, A. (1999). Effect of consumption of food cooked in iron pots on iron status and growth of young children: A randomised trial. *The Lancet, 353*, 712–716.

Preparation

```
# Load libraries
library(broom)
library(dplyr)
library(ggplot2)
library(readr)
library(sm)

# Read in data
anemia = read_csv(file = "~/Dropbox/epsy-8251/data/iron-deficiency.csv")
head(anemia)
```

```
# A tibble: 6 x 4
     id  iron    food      pot
```

```
  <int> <dbl>   <chr>    <chr>
1     1  2.40 Legumes Aluminum
2     2  2.17 Legumes Aluminum
3     3  2.41 Legumes Aluminum
4     4  2.34 Legumes Aluminum
5     5  2.41 Legumes     Clay
6     6  2.43 Legumes     Clay
```

Cross-Classification: Tabulating the Data

When you have categorical predictors, it is common to organize the data into a table. Here we cross-categorize the data by pot type and food type.

	Food Type		
Pot Type	Legumes	Meat	Vegetables
Aluminum	2.40 2.17 2.41 2.34	1.77 2.36 1.96 2.14	1.03 1.53 1.07 1.30
Clay	2.41 2.43 2.57 2.48	2.27 1.28 2.48 2.68	1.55 0.79 1.68 1.82
Iron	3.69 3.43 3.84 3.72	5.27 5.17 4.06 4.22	2.45 2.99 2.80 2.92

Examine Potential Main Effect of Pot Type

A quick examination of the data suggest that there are differences in the mean available iron content between pot types in the sample.

```
## Examine numeric evidence for race differences
pot_anemia = anemia %>%
  group_by(pot) %>%
  summarize( M = mean(iron), SD = sd(iron), N = n() )

pot_anemia
```

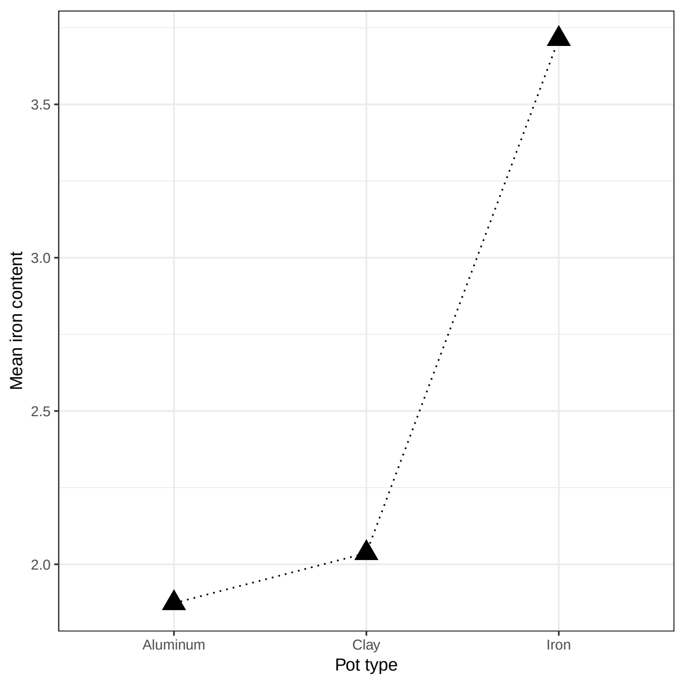
```
# A tibble: 3 x 4
       pot        M        SD     N
     <chr>    <dbl>     <dbl> <int>
1 Aluminum 1.873333 0.5222736    12
2     Clay 2.036667 0.6013369    12
3     Iron 3.713333 0.8842905    12
```

This is akin to collapsing the cross-classified table over food type.

Pot Type				Mean
Aluminum	2.40	1.77	1.03	1.87
	2.17	2.36	1.53	
	2.41	1.96	1.07	
	2.34	2.14	1.30	
Clay	2.41	2.27	1.55	2.04
	2.43	1.28	0.79	
	2.57	2.48	1.68	
	2.48	2.68	1.82	
Iron	3.69	5.27	2.45	3.71
	3.43	5.17	2.99	
	3.84	4.06	2.80	
	3.72	4.22	2.92	

We can also visualize these mean differences.

```r
# Create plot to examine race differences
ggplot(data = pot_anemia, aes(x = pot, y = M)) +
  geom_line(aes(group = 1), color = "black", linetype = "dotted") +
  geom_point(size = 5, pch = 17) +
    theme_bw() +
    xlab("Pot type") +
    ylab("Mean iron content")
```



To examine whether the sample differences are simply due to chance, we will fit the simple regression model. Because there are more than two levels for the `pot` variable, to examine whether pot type has a significant effect on iron content, we will fit an intercept-only model and the model that includes the effect of pot type, and use the nested F-test to compare the models. (Note that the intercept-only model is nested in every regression model.)

```
# Create pot type dummy variables
anemia = anemia %>%
  mutate(
    al_pot = if_else(pot == "Aluminum", 1, 0),
    cl_pot = if_else(pot == "Clay", 1, 0),
    ir_pot = if_else(pot == "Iron", 1, 0) #Do not name this iron..it is the DV
  )

# Fit intercept-only model
lm.0 = lm(iron ~ 1, data = anemia)

# Fit model with pot type effect
lm.1 = lm(iron ~ 1 + cl_pot + ir_pot, data = anemia)

# Nested F-test
anova(lm.0, lm.1)
```

```
Analysis of Variance Table

Model 1: iron ~ 1
Model 2: iron ~ 1 + cl_pot + ir_pot
  Res.Df    RSS Df Sum of Sq      F       Pr(>F)
1     35 40.474
2     33 15.580  2    24.894 26.364 0.0000001442 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The test suggests a statistically significant effect of pot type on iron content, $F(2,33) = 26.36$, $p < .001$. (There are differences in the mean iron content in food cooked in the various types of pots.) Examining the summary output (not shown) we find that differences in pot type explain 61.5% of the variation in iron content available in food.

Examine Potential Main Effect of Food Type

Although we are primarily interested in the effect of pot type, it behooves us to also examine the main effect of food type. A quick examination of the data suggest that there are differences in the mean available iron content between food types in the sample.

```
## Examine numeric evidence for race differences
food_anemia = anemia %>%
  group_by(food) %>%
  summarize( M = mean(iron), SD = sd(iron), N = n() )

food_anemia
```

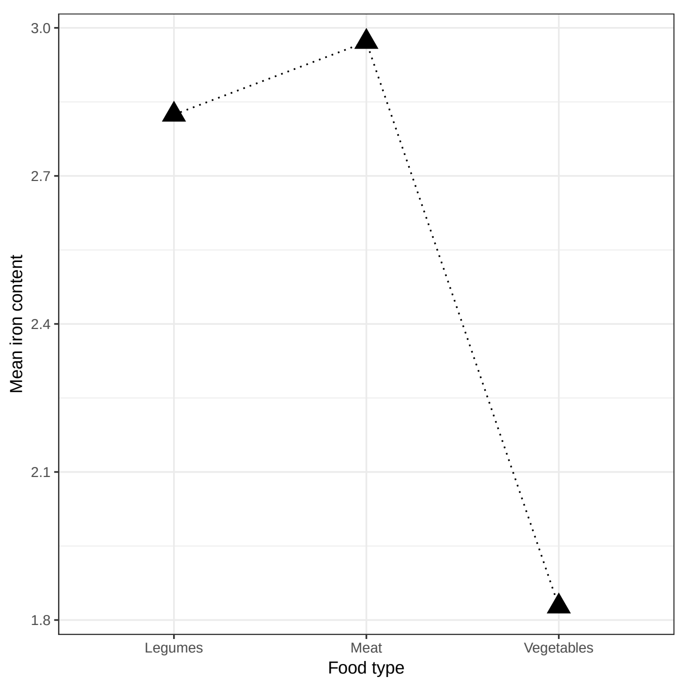
```
# A tibble: 3 x 4
        food        M        SD     N
       <chr>    <dbl>     <dbl> <int>
1    Legumes 2.824167 0.6378153    12
2       Meat 2.971667 1.3508235    12
3 Vegetables 1.827500 0.7763566    12
```

This is akin to collapsing the cross-classified table over pot type.

	Food Type		
	Legumes	Meat	Vegetables
	2.40	1.77	1.03
	2.17	2.36	1.53
	2.41	1.96	1.07
	2.34	2.14	1.30
	2.41	2.27	1.55
	2.43	1.28	0.79
	2.57	2.48	1.68
	2.48	2.68	1.82
	3.69	5.27	2.45
	3.43	5.17	2.99
	3.84	4.06	2.80
	3.72	4.22	2.92
Mean	2.82	2.97	1.83

We can also visualize these mean differences.

```
# Create plot to examine race differences
ggplot(data = food_anemia, aes(x = food, y = M)) +
  geom_line(aes(group = 1), color = "black", linetype = "dotted") +
  geom_point(size = 5, pch = 17) +
    theme_bw() +
    xlab("Food type") +
    ylab("Mean iron content")
```



To examine whether the sample differences are simply due to chance, we will fit an intercept-only model and the model that includes the effect of food type, and use the nested F-test to compare the models. (Note that the intercept-only model is nested in every regression model.)

```r
# Create pot type dummy variables
anemia = anemia %>%
  mutate(
    legume = if_else(food == "Legumes", 1, 0),
    meat = if_else(food == "Meat", 1, 0),
    vegetable = if_else(food == "Vegetables", 1, 0) #Do not name this iron..it is the DV
  )

# Fit model with pot type effect
lm.2 = lm(iron ~ 1 + meat + vegetable, data = anemia)

# Nested F-test
anova(lm.0, lm.2)
```

```
Analysis of Variance Table

Model 1: iron ~ 1
Model 2: iron ~ 1 + meat + vegetable
  Res.Df    RSS Df Sum of Sq      F  Pr(>F)
1     35 40.474
2     33 31.177  2    9.2969 4.9203 0.01349 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The test suggests a statistically significant effect of pot type on iron content, $F(2, 33) = 4.92$, $p = .013$. (There are differences in the mean iron content in the various types of food cooked.) Examining the summary output (from `lm.2` not shown) we find that differences in food type explain 23.0% of the variation in iron content available in food.

Controlled Effect of Pot Type

One explanation for the differences we saw earlier in the iron content between pot types is the type of food cooked in the pot. To see if there is an effect of pot type, after controlling for differences in food type, we (1) fit a model that includes the effects of food type, and then compare it to (2) a model that includes the effects of food type and pot type.

```r
lm.2 = lm(iron ~ 1 + meat + vegetable, data = anemia)
lm.3 = lm(iron ~ 1 + meat + vegetable + cl_pot + ir_pot, data = anemia)
anova(lm.2, lm.3)
```

```
Analysis of Variance Table

Model 1: iron ~ 1 + meat + vegetable
Model 2: iron ~ 1 + meat + vegetable + cl_pot + ir_pot
  Res.Df     RSS Df Sum of Sq      F    Pr(>F)
1     33 31.1769
2     31  6.2829  2    24.894 61.413 1.649e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The test suggests a statistically significant effect of pot type on iron content, even after conrolling for differences in food type $F(2, 31) = 24.89$, $p < .001$. Examining the summary output (not shown) we find that differences in pot type and food type explain 84.48% of the variation in iron content available in food.

Plotting the Main-Effects Model

To understand the main-effects model, we plot the results. To do this when we have categorical predictors, it is easier if we fit the model using the original categorical predictors rather than the dummy variables.

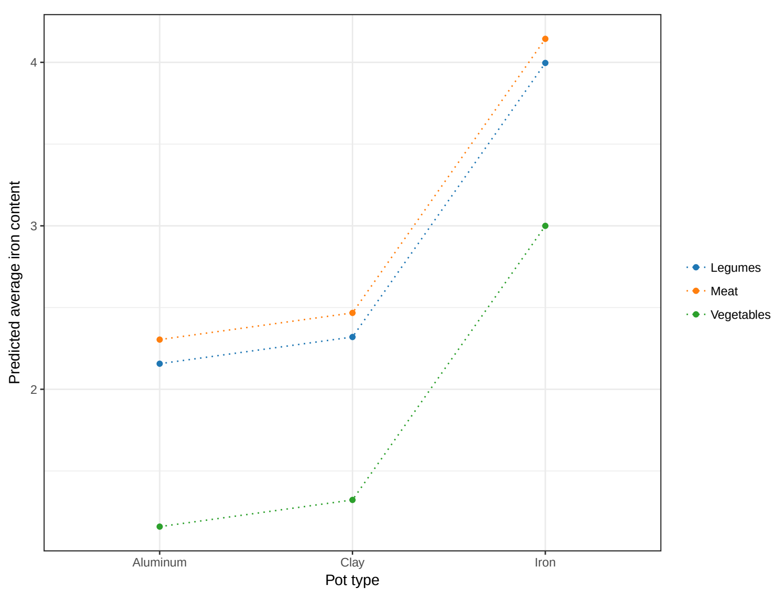
```
lm.4 = lm(iron ~ 1 + pot + food, data = anemia)
```

Then we can create our data using the categorical predictors and predict from the `lm.4` model.

```
# Create new data set
plotData = expand.grid(
  pot = c("Aluminum", "Clay", "Iron"),
  food = c("Legumes", "Meat", "Vegetables")
)

# Use fitted model to compute fitted values for the data
plotData$yhat = predict(lm.4, newdata = plotData)
head(plotData)
```

```
       pot    food     yhat
1 Aluminum Legumes 2.156389
2     Clay Legumes 2.319722
3     Iron Legumes 3.996389
4 Aluminum    Meat 2.303889
5     Clay    Meat 2.467222
6     Iron    Meat 4.143889
```

```
# Plot the fitted model
ggplot(data = plotData, aes(x = pot, y = yhat, color = food)) +
  geom_line(aes(group = food), linetype = "dotted") +
  geom_point() +
  theme_bw() +
  xlab("Pot type") +
  ylab("Predicted average iron content") +
  ggsci::scale_color_d3(name = "")
```



First note that the line segments are parallel. This suggests that the effect of pot type is exactly the same for each type of food. In general, the mean iron content available for consumption for food cooked in an iron pot is highest, followed by clay and aluminum pots, which seem to have similar percentages of iron, on average. (Note that since the two predictors are categorical, the only thing that matters are the points in this plot, because the only y-hat values that really exist are for a particular pot and food type. The lines are added only to help us interpret the effects.)

Examining the Potential Interaction Between Pot Type and Food Type

Another question a researcher might ask is whether the effect of pot type IS DIFFERENT depending on the type of food being cooked. Asking about whether there is an intereaction between two categorical variables is akin to asking whether the mean Y is different for the different combinations of the two predictors. In our example, we are asking: is the mean iron content different for the different combinations of food type and pot type.

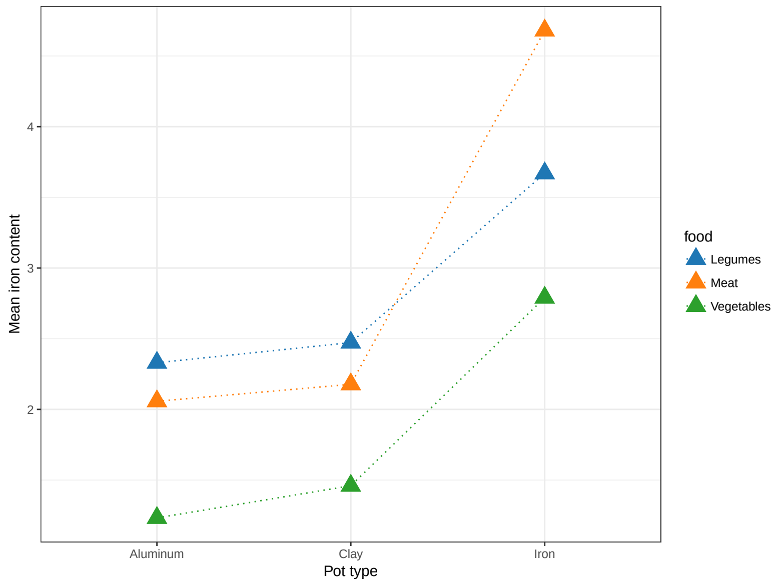
```
## Examine numerical evidence for interaction
pot_food_anemia = anemia %>%
  group_by(pot, food) %>%
  summarize( M = mean(iron), SD = sd(iron), N = n() )

pot_food_anemia
```

```
# A tibble: 9 x 5
# Groups:   pot [?]
       pot       food      M         SD     N
     <chr>      <chr>  <dbl>      <dbl> <int>
1 Aluminum    Legumes 2.3300 0.11105554     4
2 Aluminum       Meat 2.0575 0.25197553     4
3 Aluminum Vegetables 1.2325 0.23128266     4
4     Clay    Legumes 2.4725 0.07135592     4
5     Clay       Meat 2.1775 0.62130910     4
6     Clay Vegetables 1.4600 0.46007246     4
7     Iron    Legumes 3.6700 0.17262677     4
8     Iron       Meat 4.6800 0.62827807     4
9     Iron Vegetables 2.7900 0.23986107     4
```

	Food Type		
Pot Type	Legumes	Meat	Vegetables
Aluminum	2.33	2.06	1.23
Clay	2.47	2.18	1.46
Iron	3.67	4.68	2.79

We can also look at this visually.

```
# Create plot to examine the potential interaction
ggplot(data = pot_food_anemia, aes(x = pot, y = M, color = food)) +
  geom_line(aes(group = food), linetype = "dotted") +
  geom_point(size = 5, pch = 17) +
    theme_bw() +
    xlab("Pot type") +
    ylab("Mean iron content") +
  ggsci::scale_color_d3()
```



The sample data suggests a potential interaction between pot type and food type on iron content. There is variation in the nine sample means (they are not all the same). To examine whether these differences are more than we would expect because of sampling error, we need to fit an interaction model and examine the different effects.

Fitting an Interaction Model

To test the potential interaction effect between sex and race, we will fit a main-effects model (already fitted using the categorical predictors in `lm.4`) and the interaction model, and then use the nested F-test.

```
# Fit interaction model
lm.5 = lm(iron ~ 1 + food + pot + food:pot, data = anemia)

# Nested F-test
anova(lm.4, lm.5)
```

```
Analysis of Variance Table

Model 1: iron ~ 1 + pot + food
Model 2: iron ~ 1 + food + pot + food:pot
  Res.Df    RSS Df Sum of Sq     F   Pr(>F)
1     31 6.2829
2     27 3.6425  4    2.6404 4.893 0.004247 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The test suggests a statistically significant interaction effect between pot type and food type on iron content, $F(4, 27) = 4.89$, $p = .004$.

If we wanted to fit the interaction model using the dummy variables we would include the necessary dummy variables for each main effect and all possible cross-products.

```
lm.6 = lm(iron ~ 1 + cl_pot + ir_pot + meat + vegetable +
            cl_pot:meat + cl_pot:vegetable +
            ir_pot:meat + ir_pot:vegetable, data = anemia)
```

```r
# Check that results are the same (compare to main effects using dummies)
anova(lm.3, lm.6)
```

```
Analysis of Variance Table

Model 1: iron ~ 1 + meat + vegetable + cl_pot + ir_pot
Model 2: iron ~ 1 + cl_pot + ir_pot + meat + vegetable + cl_pot:meat +
    cl_pot:vegetable + ir_pot:meat + ir_pot:vegetable
  Res.Df    RSS Df Sum of Sq     F   Pr(>F)
1     31 6.2829
2     27 3.6425  4    2.6404 4.893 0.004247 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Interpreting the Interaction Effect

Researchers may want to dig deeper into which pot type/food type combinations statistically differ in their mean iron content. All we currently know is that there is an interaction effect, which means that at least one pot type/food type differs. To figure out which are different, we need to examine the `summary()` output from the model.

```r
summary(lm.5)
```

```

Call:
lm(formula = iron ~ 1 + food + pot + food:pot, data = anemia)

Residuals:
     Min       1Q   Median       3Q      Max
-0.89750 -0.16062  0.05875  0.17750  0.59000

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)
(Intercept)               2.3300     0.1837  12.687 6.87e-13 ***
foodMeat                 -0.2725     0.2597  -1.049 0.303382
foodVegetables           -1.0975     0.2597  -4.226 0.000243 ***
potClay                   0.1425     0.2597   0.549 0.587740
potIron                   1.3400     0.2597   5.159 1.98e-05 ***
foodMeat:potClay         -0.0225     0.3673  -0.061 0.951605
foodVegetables:potClay    0.0850     0.3673   0.231 0.818733
foodMeat:potIron          1.2825     0.3673   3.492 0.001669 **
foodVegetables:potIron    0.2175     0.3673   0.592 0.558668
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3673 on 27 degrees of freedom
Multiple R-squared:   0.91, Adjusted R-squared:  0.8833
F-statistic: 34.13 on 8 and 27 DF,  p-value: 3.623e-12
```

The fitted equation is,

$$
\begin{aligned}
\text{iron }\hat{\text{content}} =& 2.3 - 0.3(\text{meat}) - 1.1(\text{vegetables}) + 0.1(\text{clay pot}) + 1.3(\text{iron pot}) - 0.02(\text{meat})(\text{clay pot}) + \\
& 0.09(\text{vegetables})(\text{clay pot}) + 1.3(\text{meat})(\text{iron pot}) + 0.2(\text{vegetables})(\text{iron pot}).
\end{aligned}
$$

The reference group in the fitted model is legumes cooked in aluminum pots. All of the variables in the model are dummy coded 0 or 1, so to find the mean iron content for a particular food/pot type, we substitute in the appropriate set of 0s and 1s.

The intercept gives the mean iron content for legumes cooked in an aluminum pot ($\hat{\beta}_0 = 2.3$). What about for meat cooked in an luminum pot? Substituting in a 1 for `meat` and a 0 for `vegetables`, `iron pot`, and `clay pot`, we find that the mean iron content for meat cooked in an aluminum pot is 2.0, a mean difference of -0.3. We can continue making substitutions to compute the mean iron content for each of the pot type/food type combinations.

Pot Type	Food Type	Computation	Mean Iron Content
Aluminum	Legume	2.3	2.3
Aluminum	Meat	2.3 - 0.3	2.0
Aluminum	Vegetable	2.3 - 1.1	1.2
Clay	Legume	2.3 + 0.1	2.4
Clay	Meat	2.3 - 0.3 + 0.1 - 0.02	2.08
Clay	Vegetable	2.3 - 1.1 + 0.1 + 0.09	1.39
Iron	Legume	2.3 + 1.3	3.6
Iron	Meat	2.3 - 0.3 + 1.3 + 1.3	4.6
Iron	Vegetable	2.3 - 1.1 + 1.3 + 0.2	2.7

Looking at the table, we can see that the coefficients in the fitted model for the main-effects represent the difference between the reference group and combinations with at least one of the food type or pot type in common to the reference group:

- Intercept: The mean iron content for legumes cooked in an aluminum pot (reference group).
- Main-effect of `meat`: Difference in iron content between the reference group and meat cooked in an aluminum pot.
- Main-effect of `vegetables`: Difference in iron content between the reference group and vegetables cooked in an aluminum pot.
- Main-effect of `clay pot`: Difference in iron content between the reference group and legumes cooked in a clay pot.
- Main-effect of `iron pot`: Difference in iron content between the reference group and legumes cooked in an iron pot.

The interaction effects are less directly interpretable. This is because the mean iron content is computed using the main effects and the interaction effect for those combinations that differ from the reference group in both food and pot type. For example, for meat cooked in a clay pot we have to add the main-effect for meat, the main-effect for clay pot, and the interaction effect:

$$
\begin{aligned}
\text{iron } \hat{\text{content}} &= 2.3 - 0.3(1) + 0.1(1) - 0.02(1)(1) + \\
&= 2.08
\end{aligned}
$$

The interaction effect represents the additional combination effect beyond the separate main effects. Interaction terms are difficult to interpret without considering the constituent main-effects. This is why we always leave the constituent main-effects in the model, regardless if they are statistically significant.

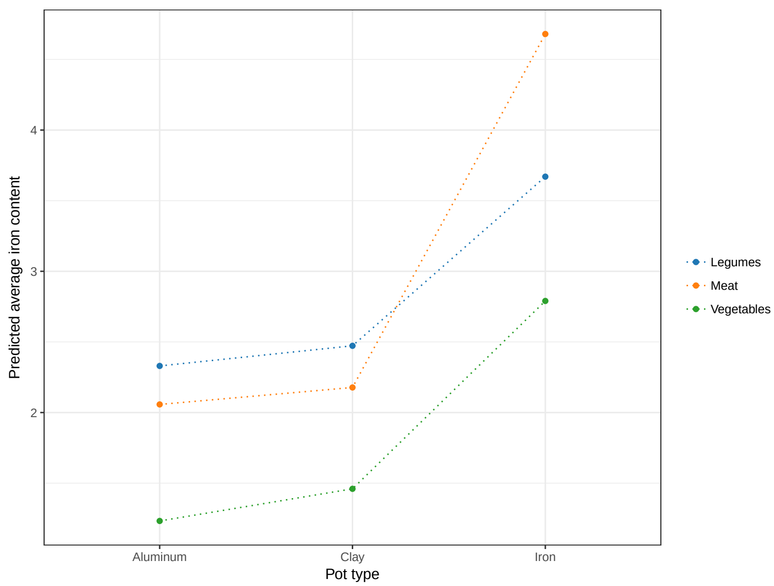
To understand this interaction better we can plot the model results.

```r
# Create new data set with main effects
plotData = expand.grid(
  pot = c("Aluminum", "Clay", "Iron"),
  food = c("Legumes", "Meat", "Vegetables")
```

```
)

# Use fitted model to compute fitted values for the data
plotData$yhat = predict(lm.5, newdata = plotData)
head(plotData)
```

```
       pot    food   yhat
1 Aluminum Legumes 2.3300
2     Clay Legumes 2.4725
3     Iron Legumes 3.6700
4 Aluminum    Meat 2.0575
5     Clay    Meat 2.1775
6     Iron    Meat 4.6800
```

```
# Plot the fitted model
ggplot(data = plotData, aes(x = pot, y = yhat, color = food)) +
  geom_line(aes(group = food), linetype = "dotted") +
  geom_point() +
  theme_bw() +
  xlab("Pot type") +
  ylab("Predicted average iron content") +
  ggsci::scale_color_d3(name = "")
```



Cooking food in an iron pot leaves the highest percentage of iron content leftover for absorption by the human body, regardless of food type. This effect is even higher when cooking meat.

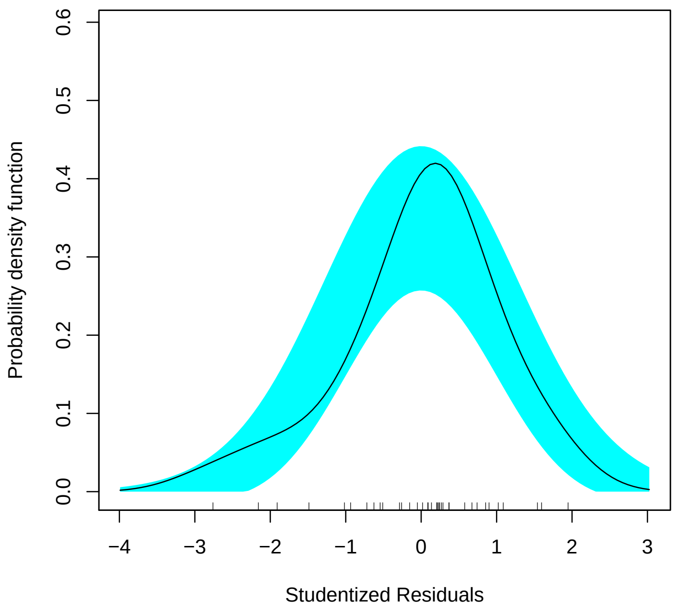
Model Assumptions

Finally, we should again examine all of the model assumptions. Based on the plots (shown below) it seems that the assumptions seem reasonably satisfied.

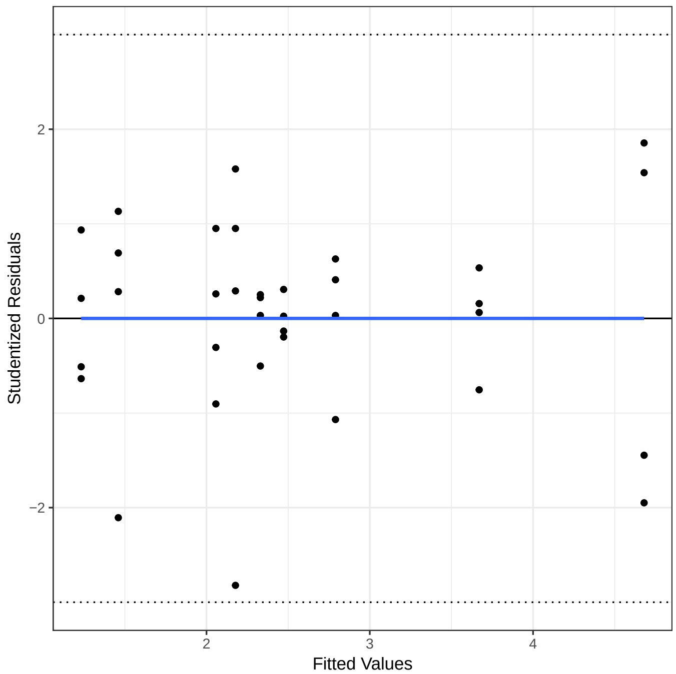
```
# Create fortified data
out5 = augment(lm.5)
head(out5)
```

```
  iron    food      pot .fitted   .se.fit  .resid .hat    .sigma
1 2.40 Legumes Aluminum  2.3300 0.1836487  0.0700 0.25 0.3739584
2 2.17 Legumes Aluminum  2.3300 0.1836487 -0.1600 0.25 0.3725364
3 2.41 Legumes Aluminum  2.3300 0.1836487  0.0800 0.25 0.3738555
4 2.34 Legumes Aluminum  2.3300 0.1836487  0.0100 0.25 0.3742874
5 2.41 Legumes     Clay  2.4725 0.1836487 -0.0625 0.25 0.3740265
6 2.43 Legumes     Clay  2.4725 0.1836487 -0.0425 0.25 0.3741704
       .cooksd  .std.resid
1 0.0017936399  0.22006426
2 0.0093708534 -0.50300402
3 0.0023427133  0.25150201
4 0.0000366049  0.03143775
5 0.0014298787 -0.19648594
6 0.0006611759 -0.13361044
```

```r
# Examine normality assumption
sm.density(out5$.std.resid, model = "normal", xlab = "Studentized Residuals")
```




```r
# Examine other assumptions
ggplot(data = out5, aes(x = .fitted, y = .std.resid)) +
  geom_point() +
  geom_hline(yintercept = 0) +
  geom_hline(yintercept = c(-3, 3), linetype = "dotted") +
  geom_smooth(se = FALSE) +
  theme_bw() +
  xlab("Fitted Values") +
  ylab("Studentized Residuals")
```



Pairwise Differences between the Nine Pot/Food Combinations

One way to explore the interaction is to look at the pairwise differences between pot/food type combinations. If you are interested in these pairwise differences, you need to fit several different models (with varying reference groups) and collect the appropriate p-values (which are unadjusted), and then adjust them based on the number of comparisons.

For example, here, there are 36 different p-values you would need to collect and adjust. Based on the model we just fitted, we have the following four comparisons.

Pot/Food	Pot/Food	p
Aluminum/Legume	Aluminum/Meat	0.303382
Aluminum/Legume	Aluminum/Vegetable	0.000243
Aluminum/Legume	Clay/Legume	0.587740
Aluminum/Legume	Iron/Legume	0.0000198

We would have to collect the remaining 32 different comparisons, and then we could put them in a vector and adjust using the Benjamani-Hochberg adjustment. Since all the predictors are categorical, we can use the `pairwise.t.test()` function to quickly do this.

```r
## Create pot x food combination
anemia = anemia %>%
  mutate(food_pot = paste0(food, "_", pot))
head(anemia)
```

```
# A tibble: 6 x 11
     id  iron    food      pot al_pot cl_pot ir_pot legume  meat vegetable
  <int> <dbl>   <chr>    <chr>  <dbl>  <dbl>  <dbl>  <dbl> <dbl>     <dbl>
1     1  2.40 Legumes Aluminum      1      0      0      1     0         0
2     2  2.17 Legumes Aluminum      1      0      0      1     0         0
3     3  2.41 Legumes Aluminum      1      0      0      1     0         0
```

```
4     4  2.34 Legumes Aluminum      1      0      0      1     0          0
5     5  2.41 Legumes     Clay      0      1      0      1     0          0
6     6  2.43 Legumes     Clay      0      1      0      1     0          0
# ... with 1 more variables: food_pot <chr>
```

```r
pairwise.t.test(x = anemia$iron, g = anemia$food_pot, p.adjust.method = "BH")
```

```

	Pairwise comparisons using t tests with pooled SD 

data:  anemia$iron and anemia$food_pot 

                    Legumes_Aluminum Legumes_Clay Legumes_Iron
Legumes_Clay        0.60453          -            -           
Legumes_Iron        5.5e-05          0.00020      -           
Meat_Aluminum       0.34130          0.15109      4.4e-06     
Meat_Clay           0.59502          0.30891      1.2e-05     
Meat_Iron           7.0e-09          1.8e-08      0.00112     
Vegetables_Aluminum 0.00051          0.00013      3.9e-09     
Vegetables_Clay     0.00392          0.00112      1.8e-08     
Vegetables_Iron     0.11292          0.27851      0.00373     
                    Meat_Aluminum Meat_Clay Meat_Iron Vegetables_Aluminum
Legumes_Clay        -             -         -         -                  
Legumes_Iron        -             -         -         -                  
Meat_Aluminum       -             -         -         -                  
Meat_Clay           0.64776       -         -         -                  
Meat_Iron           1.4e-09       2.8e-09   -         -                  
Vegetables_Aluminum 0.00581       0.00206   8.6e-12   -                  
Vegetables_Clay     0.03916       0.01468   2.1e-11   0.42413            
Vegetables_Iron     0.01332       0.03580   3.2e-07   7.0e-06            
                    Vegetables_Clay
Legumes_Clay        -              
Legumes_Iron        -              
Meat_Aluminum       -              
Meat_Clay           -              
Meat_Iron           -              
Vegetables_Aluminum -              
Vegetables_Clay     -              
Vegetables_Iron     5.7e-05        

P value adjustment method: BH 
```

The argument `x=` takes the outcome, and the `g=` argument takes the groups. We use the colon notation to define the groups (every pot/food combination). Finally we use the argument `p.adjust.method=` to specify the p-value adjustment method as Benjamani-Hochberg. This method will not work when any of the predictors are quantitative!