Interaction Terms & Hierarchical Linear Models

Fitting Models to Data, Not Data to Models
Model Fitting Series - With Applications in R

Jesse Ghashti

October 21, 2025

Centre for Scholarly Communication
The University of British Columbia | Okanagan Campus | Syilx Okanagan Nation Territory

Welcome, feel free to get ready while we wait to start...

Install Required Packages

```r
packages <- c("dplyr", "mgcv", "ggplot2",
              "gratia", "janitor")
toInstall <- packages[!(packages %in%
                          installed.packages()[,"Package"])]
if(length(toInstall)) install.packages(toInstall)

library('dplyr')   # for data wrangling
library('mgcv')    # for modeling
library('ggplot2') # for fancy plots
library('janitor') # for data cleaning
library('gratia')  # ggplot-based graphics
```

Workshop Series Overview

Session	Topic	Date/Time
1	Simple Linear Regression	Oct 7, 9:00 AM
2	Fitting Linear Models in R	Oct 8, 10:30 AM
3	Multiple Linear Regression in R	Oct 16, 4:00 PM
4	**Interaction Terms & Hierarchical Linear Models**	**Oct 21, 11:00 AM**
5	Generalized Linear Models	Oct 23, 4:00 PM
6	Generalized Additive Models (GAMs)	Oct 28, 11:00 AM
7	Interpreting & Predicting from GAMs	Oct 29, 10:30 AM
8	Hierarchical GAMs	Nov 4, 12:00 PM
9	Penalized Models	Nov 18, 11:00 AM
10	Survival Models	Nov 25, 11:00 AM
11	Nonparametric Models	Dec 2, 11:00 AM

New Here?

<——————————-

New to R? Check out the Fundamentals of R series!

GitHub code and slides for today's workshop (and previous workshops)

——————————>

Alternatively, code/slides available at the bottom of `https://csc-ubc-okanagan.github.io/workshops/`

Last Time (Workshop 3) — Quick Recap

Key Concepts

- Extended to multiple linear regression: $Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \epsilon$
- Used DAGs to visualize causal relationships
- Learned about multicollinearity and VIF
- Interpreted partial effects (holding other variables constant)
- Saw that more predictors $\neq$ better model (use R^2_{adj})

Today: Interaction Terms & Hierarchical Models

Today we will...

- Learn when effects depend on context (interactions)
- Distinguish fixed effects vs random effects
- Build hierarchical linear models (mixed models)
- Work with `ChickWeight` data: growth curves by diet
- Visualize and interpret interaction effects

Today we require...

```r
library('dplyr')   # for data wrangling
library('mgcv')    # for modeling
library('ggplot2') # for fancy plots
library('gratia')  # for ggplot-based model graphics
library('janitor') # for cleaning names
theme_set(theme_classic(base_size = 15))
```

What Are Interaction Terms?

When Effects Depend on Context

Without interaction: $Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \epsilon$

- Effect of x_1 is always β_1, regardless of x_2

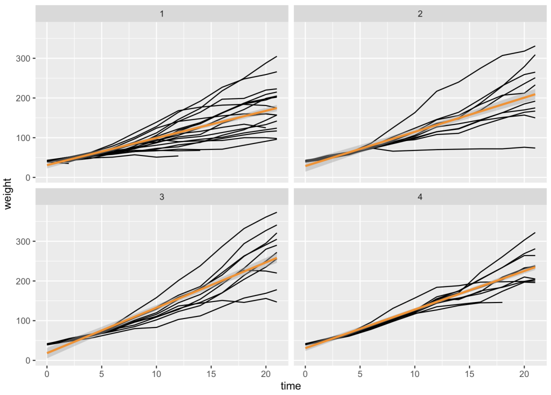
With interaction: $Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + \epsilon$

- Effect of x_1 depends on x_2: $\frac{\partial Y}{\partial x_1} = \beta_1 + \beta_3 x_2$

Real-world example: The effect of study time on grades might depend on teaching method

Exploring ChickWeight Data

```r
cw <- janitor::clean_names(ChickWeight) %>%
  as_tibble()
# Plot growth by diet
ggplot(cw, aes(time, weight)) +
  facet_wrap(~ diet) +
  geom_line(aes(group = chick)) +
  geom_smooth(method = 'lm',
              formula = y ~ x,
              color = 'darkorange')
```

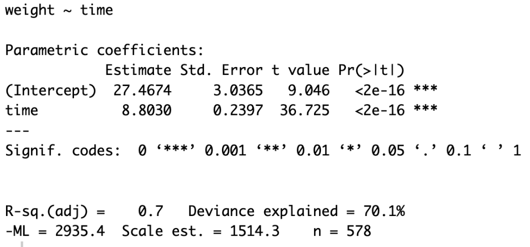


Each diet shows different growth patterns — perfect for interactions!

Model 0: Ignoring Diet Differences I

```r
# Model that only accounts for time
m_cw_0 <- gam(formula = weight ~ time,
              family = gaussian(),
              data = cw,
              method = 'ML')

summary(m_cw_0)
```

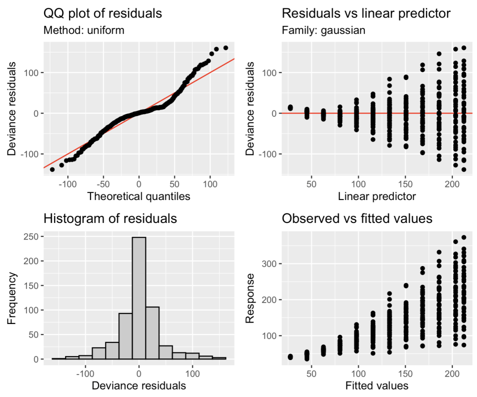
This model assumes all chicks grow at the same rate!



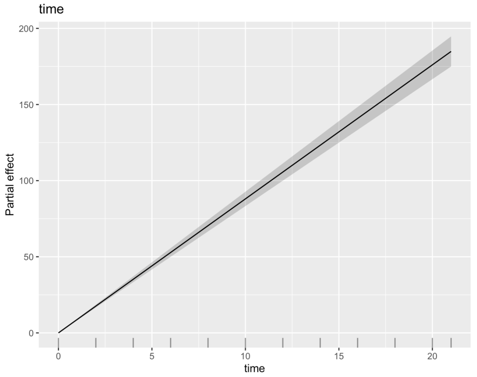
```
weight ~ time

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  27.4674     3.0365   9.046   <2e-16 ***
time          8.8030     0.2397  36.725   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1


R-sq.(adj) =    0.7   Deviance explained = 70.1%
-ML = 2935.4  Scale est. = 1514.3    n = 578
```

Model 0: Ignoring Diet Differences II

```r
appraise(m_cw_0)
```



```r
draw(m_cw_0, parametric = TRUE)
```



Fixed Effects vs Random Effects

Fixed Effects

- Independent categories
- We care about specific levels
- Examples:
  - Treatment vs Control
  - Male vs Female
  - Diet 1, 2, 3, 4
- Each gets its own parameter

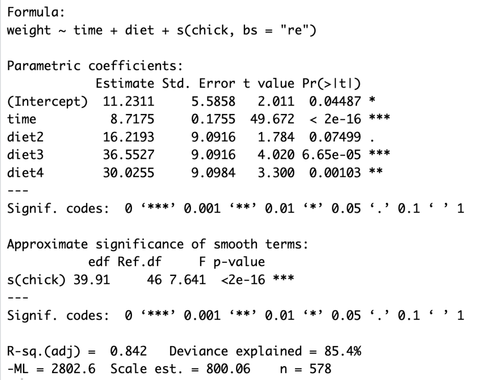
Random Effects

- Levels from a population
- We care about variation
- Examples:
  - Individual subjects
  - Schools in a district
  - Chicks in experiment
- Share information (pooling)

Random effects allow "borrowing strength" across levels

Model 1: Adding Diet & Chick Effects I

```r
m_cw_1 <- gam(
  formula = weight ~
    time +              # temporal trend
    diet +              # fixed: diet type
    s(chick, bs = 're'), # random: chicks
  family = gaussian(),
  data = cw,
  method = 'ML')
summary(m_cw_1)
# Note changes in R-sq, Dev. expl.
```



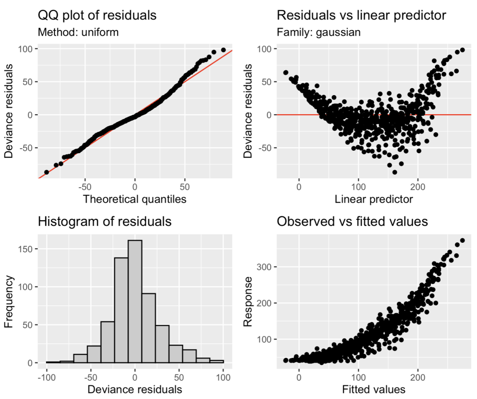
```
Formula:
weight ~ time + diet + s(chick, bs = "re")

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  11.2311     5.5858   2.011  0.04487 *
time          8.7175     0.1755  49.672  < 2e-16 ***
diet2        16.2193     9.0916   1.784  0.07499 .
diet3        36.5527     9.0916   4.020 6.65e-05 ***
diet4        30.0255     9.0984   3.300  0.00103 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Approximate significance of smooth terms:
           edf Ref.df     F p-value
s(chick) 39.91     46 7.641  <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

R-sq.(adj) =  0.842   Deviance explained = 85.4%
-ML = 2802.6  Scale est. = 800.06    n = 578
```

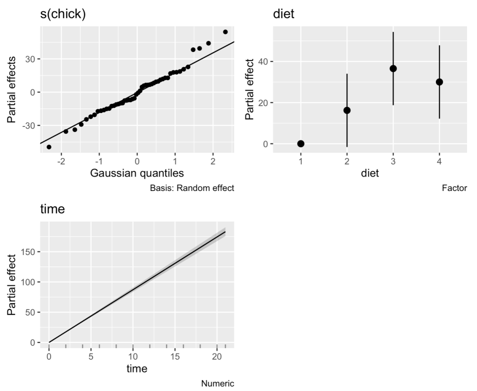
`s(chick, bs = 're')` creates random intercepts — each chick can start at a different weight

Model 1: Adding Diet & Chick Effects II

```r
appraise(m_cw_1)
```

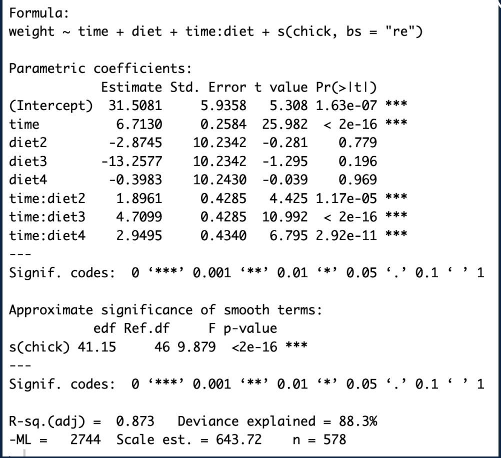
```r
draw(m_cw_1, parametric = TRUE)
```





Model 2: Diet Affects Growth Rate I

```r
m_cw_2 <- gam(formula = weight ~
                time +
                diet +
                time:diet +
                s(chick, bs = 're'),
              family = gaussian(),
              data = cw,
              method = 'ML')
summary(m_cw_2)
```



```
Formula:
weight ~ time + diet + time:diet + s(chick, bs = "re")

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  31.5081     5.9358   5.308 1.63e-07 ***
time          6.7130     0.2584  25.982  < 2e-16 ***
diet2        -2.8745    10.2342  -0.281    0.779    
diet3       -13.2577    10.2342  -1.295    0.196    
diet4        -0.3983    10.2430  -0.039    0.969    
time:diet2    1.8961     0.4285   4.425 1.17e-05 ***
time:diet3    4.7099     0.4285  10.992  < 2e-16 ***
time:diet4    2.9495     0.4340   6.795 2.92e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Approximate significance of smooth terms:
           edf Ref.df     F p-value    
s(chick) 41.15     46 9.879  <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

R-sq.(adj) =  0.873   Deviance explained = 88.3%
-ML =   2744  Scale est. = 643.72    n = 578
```

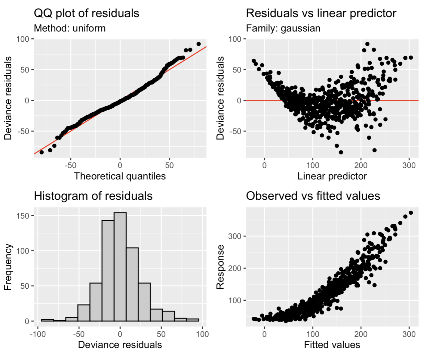
`time:diet` means each diet has its own slope — some diets lead to faster growth?

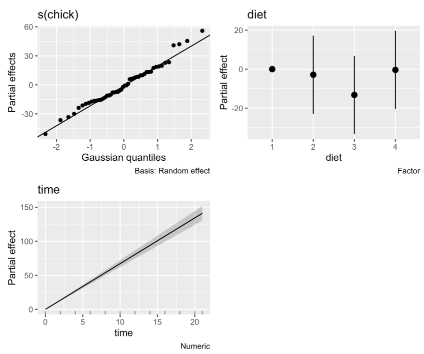
Model 2: Diet Affects Growth Rate II

Interpretability of `draw()` might be a little abstract since no interaction terms are plotted…

```r
appraise(m_cw_2)
```

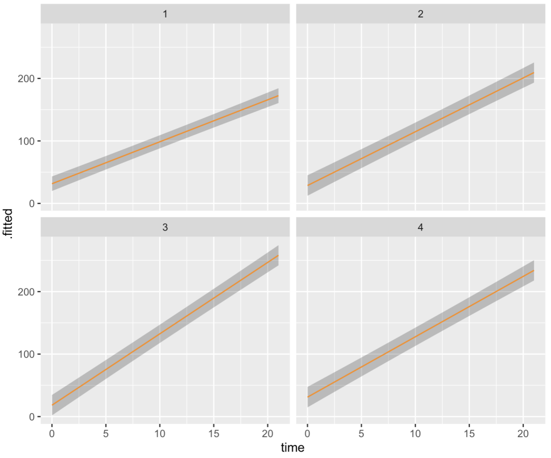
```r
draw(m_cw_2, parametric = TRUE)
```





Visualizing Interaction Effects

```r
# plot predictions for "average" chick
m_cw_2 %>%
  data_slice(time = evenly(time),
             diet = evenly(diet),
             chick = 'new') %>%
  fitted_values(object = m_cw_2,
                data = .) %>%
  ggplot() +
  facet_wrap(~ diet) +
  geom_ribbon(aes(time,
                  ymin = .lower_ci,
                  ymax = .upper_ci),
              alpha = 0.3) +
  geom_line(aes(time, .fitted),
            color = 'darkorange')
```



Continuous × Continuous Interactions

When Both Variables Are Continuous

Model: $Y = \beta_0 + \beta_1 \text{time} + \beta_2 \text{food} + \beta_3 \text{time} \times \text{food} + \epsilon$

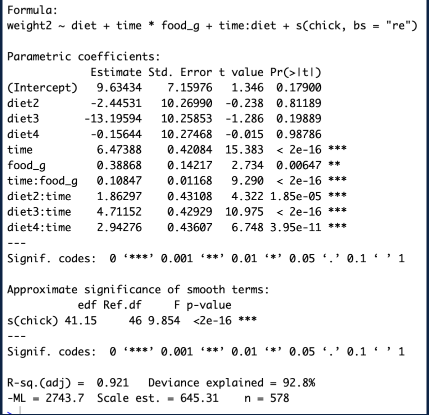
The effect of time now depends on food amount:

$$\frac{\partial \text{weight}}{\partial \text{time}} = \beta_1 + \beta_3 \times \text{food}$$

```r
# Adding hypothetical food amounts
cw <- cw %>%
  mutate(food_g = sample(c(10, 20, 30, 40, 50),
                         size = n(), replace = TRUE),
         weight2 = weight - 25 + food_g * 0.5 +
                   food_g * time * 0.1)
```

Model 3: Multiple Interactions I

```r
m_cw_3 <- gam(formula = weight2 ~
                diet +
                time * food_g +
                time:diet +
                s(chick, bs = 're'),
              family = gaussian(),
              data = cw,
              method = 'ML')
summary(m_cw_3)
```



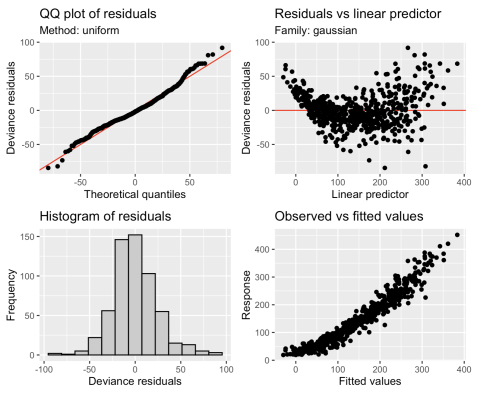
```
Formula:
weight2 ~ diet + time * food_g + time:diet + s(chick, bs = "re")

Parametric coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   9.63434    7.15976   1.346  0.17900
diet2        -2.44531   10.26990  -0.238  0.81189
diet3       -13.19594   10.25853  -1.286  0.19889
diet4        -0.15644   10.27468  -0.015  0.98786
time          6.47388    0.42084  15.383  < 2e-16 ***
food_g        0.38868    0.14217   2.734  0.00647 **
time:food_g   0.10847    0.01168   9.290  < 2e-16 ***
diet2:time    1.86297    0.43108   4.322 1.85e-05 ***
diet3:time    4.71152    0.42929  10.975  < 2e-16 ***
diet4:time    2.94276    0.43607   6.748 3.95e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Approximate significance of smooth terms:
           edf Ref.df     F p-value
s(chick) 41.15     46 9.854  <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

R-sq.(adj) =  0.921   Deviance explained = 92.8%
-ML = 2743.7  Scale est. = 645.31    n = 578
```

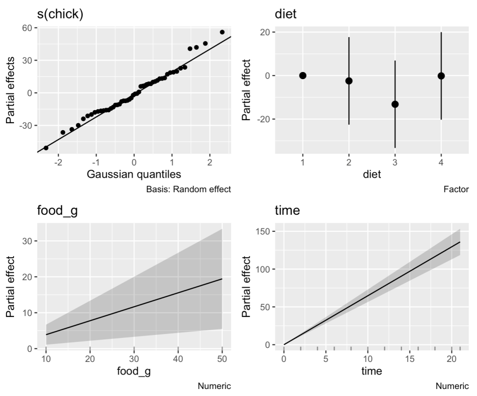
The $*$ operator includes main effects and interaction, while $:$ is just the interaction

Model 3: Multiple Interactions II

```r
appraise(m_cw_3)
```

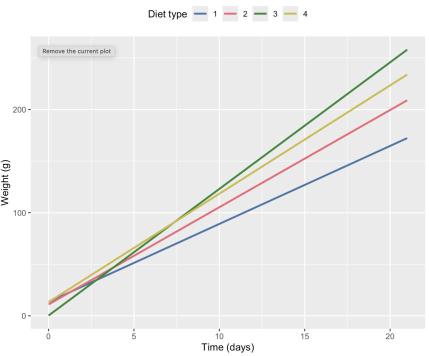
```r
draw(m_cw_3, parametric = TRUE)
```





Visualizing Continuous × Continuous Interactions I

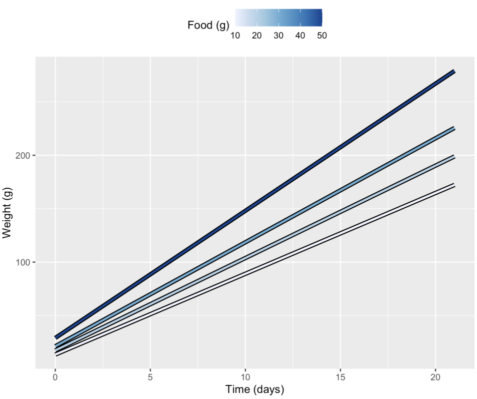
```
preds_3 <- m_cw_3 %>%
  data_slice(time = evenly(time),
             diet = evenly(diet),
             food_g = evenly(food_g),
             chick = 'new') %>%
  fitted_values(object = m_cw_3, data = .)

# plot weight over time by diet
preds_3 %>%
  filter(food_g == min(food_g)) %>%
  ggplot(aes(time, .fitted, color = diet)) +
  geom_line(linewidth = 1) +
  labs(x = 'Time (days)', y = 'Weight (g)') +
  khroma::scale_color_bright(name = 'Diet type') +
  theme(legend.position = 'top')
```



Brighter colors = higher weight; steeper gradient = stronger interaction

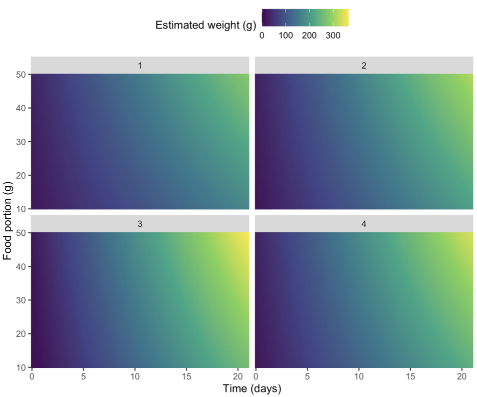
Visualizing Continuous × Continuous Interactions II

```r
# plot weight over time by food_g
m_cw_3 %>%
  data_slice(time = evenly(time),
             diet = evenly(diet),
             food_g = unique(food_g),
             chick = 'new') %>%
  fitted_values(object = m_cw_3, data = .) %>%
  filter(diet == 1) %>%
  filter(food_g %in% sample(unique(food_g), 4)) %>%
  ggplot(aes(time, .fitted, group = food_g)) +
  geom_line(linewidth = 2) +
  geom_line(aes(color = food_g), linewidth = 1) +
  labs(x = 'Time (days)', y = 'Weight (g)') +
  scale_color_distiller('Food (g)', type = 'seq',
                        palette = 1,
                        direction = 1) +
  theme(legend.position = 'top')
```



Visualizing Continuous × Continuous Interactions III

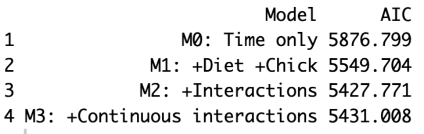
```r
#' plot `time * food_g` interaction for each diet
ggplot(preds_3, aes(time, food_g, fill = .fitted)) +
  facet_wrap(~ diet) +
  geom_raster() +
  scale_fill_viridis_c('Estimated weight (g)',
                       limits = c(0, NA)) +
  scale_x_continuous('Time (days)',
                     expand = c(0, 0)) +
  scale_y_continuous('Food portion (g)',
                     expand = c(0, 0)) +
  theme(legend.position = 'top')
```



Prediction I

So what model does the best at predicting new observations? We haven't really examined this until now.

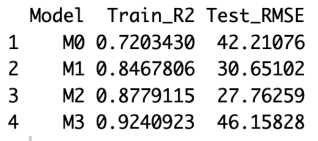
```r
compTab <- data.frame(
  Model = c("M0: Time only",
            "M1: +Diet +Chick",
            "M2: +Interactions",
            "M3: +Continuous interactions"),
  AIC = c(AIC(m_cw_0),
          AIC(m_cw_1),
          AIC(m_cw_2),
          AIC(m_cw_3))
)
compTab
```



```
                         Model      AIC
1                M0: Time only 5876.799
2             M1: +Diet +Chick 5549.704
3            M2: +Interactions 5427.771
4 M3: +Continuous interactions 5431.008
```

Prediction II

```r
set.seed(10212025)
keep <- sample(1:nrow(cw), 0.8*nrow(cw), replace = F)
trainDat <- cw[keep,]; testDat <- cw[-keep, ]
m0_train <- update(m_cw_0, data = trainDat)
m1_train <- update(m_cw_1, data = trainDat)
m2_train <- update(m_cw_2, data = trainDat)
m3_train <- update(m_cw_3, data = trainDat)
testDat <- testDat %>% mutate(
    pred_m0 = predict(m0_train, newdata = testDat),
    pred_m1 = predict(m1_train, newdata = testDat),
    pred_m2 = predict(m2_train, newdata = testDat),
    pred_m3 = predict(m3_train, newdata = testDat))
rmse <- function(actual, predicted) sqrt(mean((actual - predicted)^2))
rmseSummary <- data.frame(Model = c("M0", "M1", "M2", "M3"),
  Train_R2 = c(summary(m0_train)$r.sq, summary(m1_train)$r.sq,
               summary(m2_train)$r.sq, summary(m3_train)$r.sq),
  Test_RMSE = c(rmse(testDat$weight, testDat$pred_m0),
                rmse(testDat$weight, testDat$pred_m1),
                rmse(testDat$weight, testDat$pred_m2),
                rmse(testDat$weight, testDat$pred_m3)))
rmseSummary
```



```
  Model  Train_R2 Test_RMSE
1    M0 0.7203430  42.21076
2    M1 0.8467806  30.65102
3    M2 0.8779115  27.76259
4    M3 0.9240923  46.15828
```

Interpreting Interaction Terms

Key Points for Interpretation

- **Main effects**: Average effect when other variable = 0
- **Interaction coefficient**: How much the slope changes
- **Centering helps**: Consider centering continuous variables
- **Plot, plot, plot**: Visualizations are very helpful

Good Practice

- Include main effects with interactions
- Check if interaction improves fit
- Visualize at multiple levels

Common Mistakes

- Interaction without main effects
- Over-interpreting small interactions
- Ignoring multicollinearity

You Try: Practice with Interactions

Exercise: Explore the sleep study data

```r
# load the sleep study data which is a part of the package 'lme4'
library('lme4')
?sleepstudy  # reaction times with sleep deprivation

# Your tasks:
# 1. Plot reaction time vs days for each subject
# 2. Fit a model with random intercepts for subjects
# 3. Add random slopes (do subjects differ in how sleep loss affects them?)
# 4. Compare models and determine a 'best' model using diagnostics

# you might wish to start with
ggplot(sleepstudy, aes(Days, Reaction)) +
  geom_line(aes(group = Subject)) +
  facet_wrap(~ Subject)
```

Takeaways

- Interactions allow effects to vary by context: $Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2$
- Fixed effects: specific levels we care about
- Random effects: sample from population, allow pooling
- Hierarchical models handle grouped/nested data
- Usually you want to include main effects with interactions
- Visualization is important for understanding interactions

Remember that with interactions, "it depends" becomes part of your model!

What's Next?

Additional Questions?
Book an Appointment!

Next Workshop:

Generalized Linear Models October 23, 4:00 PM
– Beyond normal distributions
– Link functions

Additional Resources:

- *Data Analysis Using Regression and Multilevel/Hierarchical Models* (Gelman & Hill)