LONGITUDINAL ANALYSIS IN R

Adding Predictors

Brandon LeBeau
Assistant Professor

Add a categorical predictor

- Predictors are commonly added to the fixed portion of the model

```
library(nlme) # This is used to get the BodyWeight Data

BodyWeight <- BodyWeight %>%
    mutate(Time = Time - 1,
           diet_f = paste("Diet", Diet, sep = " "))

body_weight <- lmer(weight ~ 1 + Time + diet_f +
    (1 + Time  | Rat), data = BodyWeight)

summary(body_weight)
```

Summary output

```
Linear mixed model fit by REML ['lmerMod']
Formula: weight ~ 1 + Time + diet_f + (1 + Time | Rat)
   Data: BodyWeight

REML criterion at convergence: 1159.3

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.2254 -0.4297  0.0563  0.5707  2.8872

Random effects:
 Groups   Name        Variance Std.Dev. Corr
 Rat      (Intercept) 1389.96  37.2822
          Time           0.12   0.3463  -0.20
 Residual               19.75   4.4436
Number of obs: 176, groups:  Rat, 16

Fixed effects:
              Estimate Std. Error t value
(Intercept)  247.04292   13.07029  18.901
Time           0.58568    0.08828   6.634
diet_fDiet 2 214.58745   22.38754   9.585
diet_fDiet 3 258.92736   22.38754  11.566
```

Fixed effects output

```
Fixed effects:
               Estimate Std. Error t value
(Intercept)   247.04292   13.07029  18.901
Time            0.58568    0.08828   6.634
diet_fDiet 2 214.58745   22.38754   9.585
diet_fDiet 3 258.92736   22.38754  11.566
```

- Average starting weight diet 1:
  - 247.0
- Average starting weight diet 2:
  - 247.0 + 214.6 = 461.6
- Average starting weight diet 3:
  - 247.0 + 258.9 = 505.9

Random effects output

With diet:

```
Random effects:
 Groups   Name        Variance Std.Dev. Corr
 Rat      (Intercept) 1389.96  37.2822
          Time           0.12   0.3463  -0.20
 Residual               19.75   4.4436
Number of obs: 176, groups:  Rat, 16
```

Without diet predictors:

```
Random effects:
 Groups   Name        Variance Std.Dev. Corr
 Rat      (Intercept) 15246.80 123.4779
          Time            0.12   0.3463 0.56
 Residual                19.75   4.4436
Number of obs: 176, groups:  Rat, 16
```

Visualize predictor

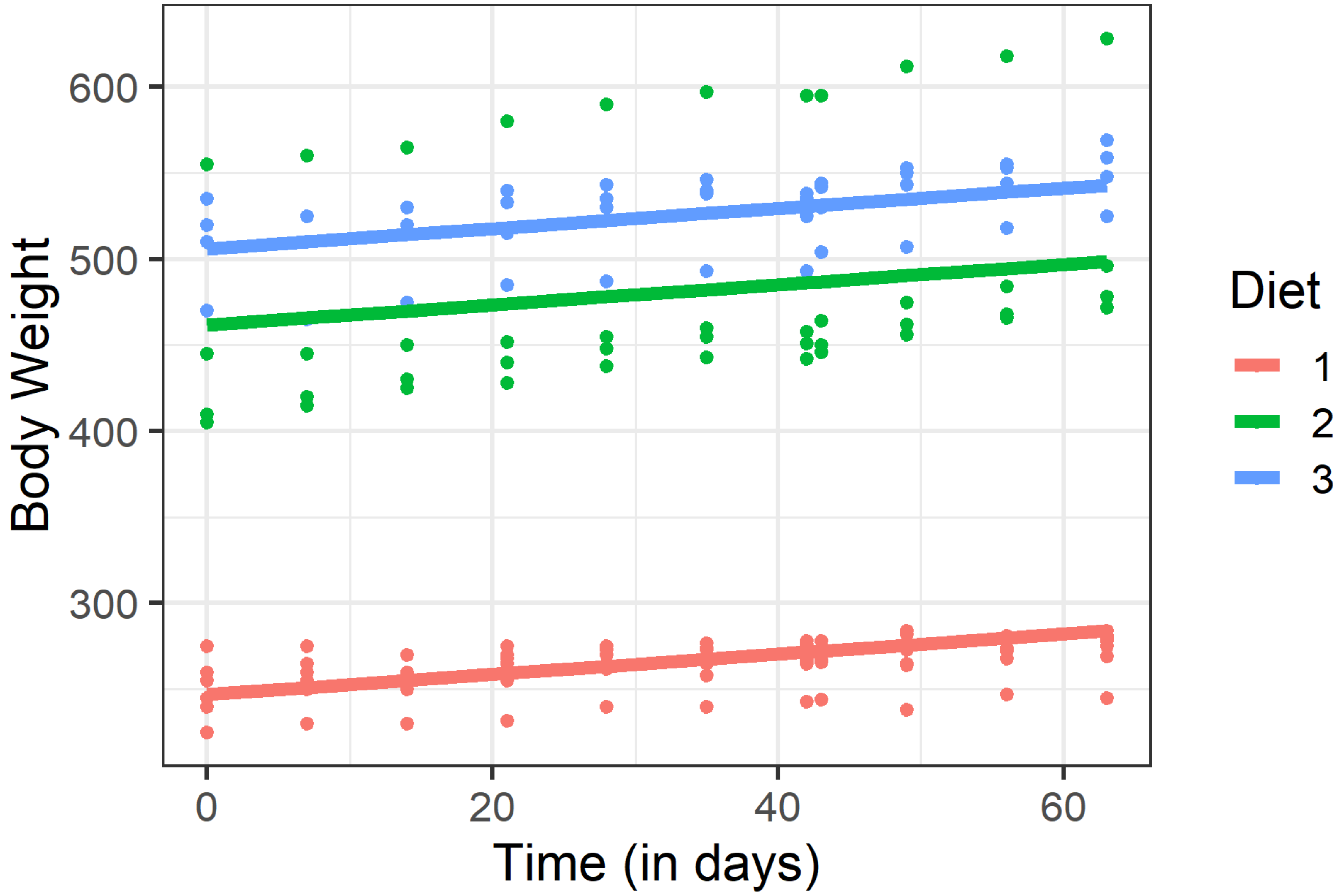
- Visualizations are particularly helpful to interpret predictor effects

```
BodyWeight <- BodyWeight %>%
    mutate(Time = Time - 1,
           diet_f = paste("Diet", Diet, sep = " "))

body_weight <- lmer(weight ~ 1 + Time + diet_f +
    (1 + Time  | Rat), data = BodyWeight)

bodyweight_agg <- BodyWeight %>%
   mutate(pred_values = predict(body_weight, re.form = NA)) %>%
   group_by(Time, Diet) %>%
   summarize(mean_diet_pred = mean(pred_values))
```

```
# visualize predicted values by each Diet Condition
ggplot(bodyweight_agg, aes(x = Time, y = mean_diet_pred, color = Diet)) +
   geom_point(data = BodyWeight, aes(x = Time, y = weight)) +
   geom_line(size = 2) +
   ylab("Body Weight") +
   xlab("Time (in days)") +
   theme_bw(base_size = 16)
```


Body Weight
600
500
400
300
0
20
40
60
Time (in days)
Diet
1
2
3

LONGITUDINAL ANALYSIS IN R

Time to practice!

LONGITUDINAL ANALYSIS IN R

Adding Predictors - Interactions

Brandon LeBeau
Assistant Professor

Adding additional predictor variables

```
library(nlme) # This is used to get the BodyWeight Data

BodyWeight <- BodyWeight %>%
    mutate(Time = Time - 1,
           diet_f = paste("Diet", Diet, sep = " "))

body_weight <- lmer(weight ~ 1 + Time + diet_f + diet_f:Time +
     (1 + Time  | Rat), data = BodyWeight)

summary(body_weight)
```

Model summary

```
Linear mixed model fit by REML ['lmerMod']
Formula: weight ~ 1 + Time + diet_f + diet_f:Time + (1 + Time | Rat)
   Data: BodyWeight

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.2556 -0.4220  0.0823  0.5993  2.7899

Random effects:
 Groups   Name        Variance  Std.Dev. Corr
 Rat      (Intercept) 1.362e+03 36.9028
          Time        6.171e-02  0.2484  -0.14
 Residual             1.975e+01  4.4436
Number of obs: 176, groups:  Rat, 16

Fixed effects:
                   Estimate Std. Error t value
(Intercept)       252.01129   13.07972  19.267
Time                0.35964    0.09114   3.946
diet_fDiet 2      201.27133   22.65474   8.884
diet_fDiet 3      252.37002   22.65474  11.140
Time:diet_fDiet 2   0.60584    0.15786   3.838
Time:diet_fDiet 3   0.29834    0.15786   1.890
```

Fixed effects output

```
Fixed effects:
                     Estimate Std. Error t value
(Intercept)         252.01129   13.07972  19.267
Time                  0.35964    0.09114   3.946
diet_fDiet 2        201.27133   22.65474   8.884
diet_fDiet 3        252.37002   22.65474  11.140
Time:diet_fDiet 2     0.60584    0.15786   3.838
Time:diet_fDiet 3     0.29834    0.15786   1.890
```

- Linear slope diet 1:
  - 0.36
- Linear slope diet 2:
  - 0.36 + 0.61 = 0.97
- Linear slope diet 3:
  - 0.36 + 0.30 = 0.66

Random effects output

- Random effects output

```
Random effects:
Groups   Name        Variance  Std.Dev. Corr
Rat      (Intercept) 1.362e+03 36.9028
         Time        6.171e-02  0.2484  -0.14
Residual             1.975e+01  4.4436
Number of obs: 176, groups:  Rat, 16
```

- Random effects output no predictors:

```
Random effects:
Groups   Name        Variance Std.Dev. Corr
Rat      (Intercept) 15246.80 123.4779
        Time            0.12   0.3463 0.56
Residual                19.75   4.4436
Number of obs: 176, groups:  Rat, 16
```

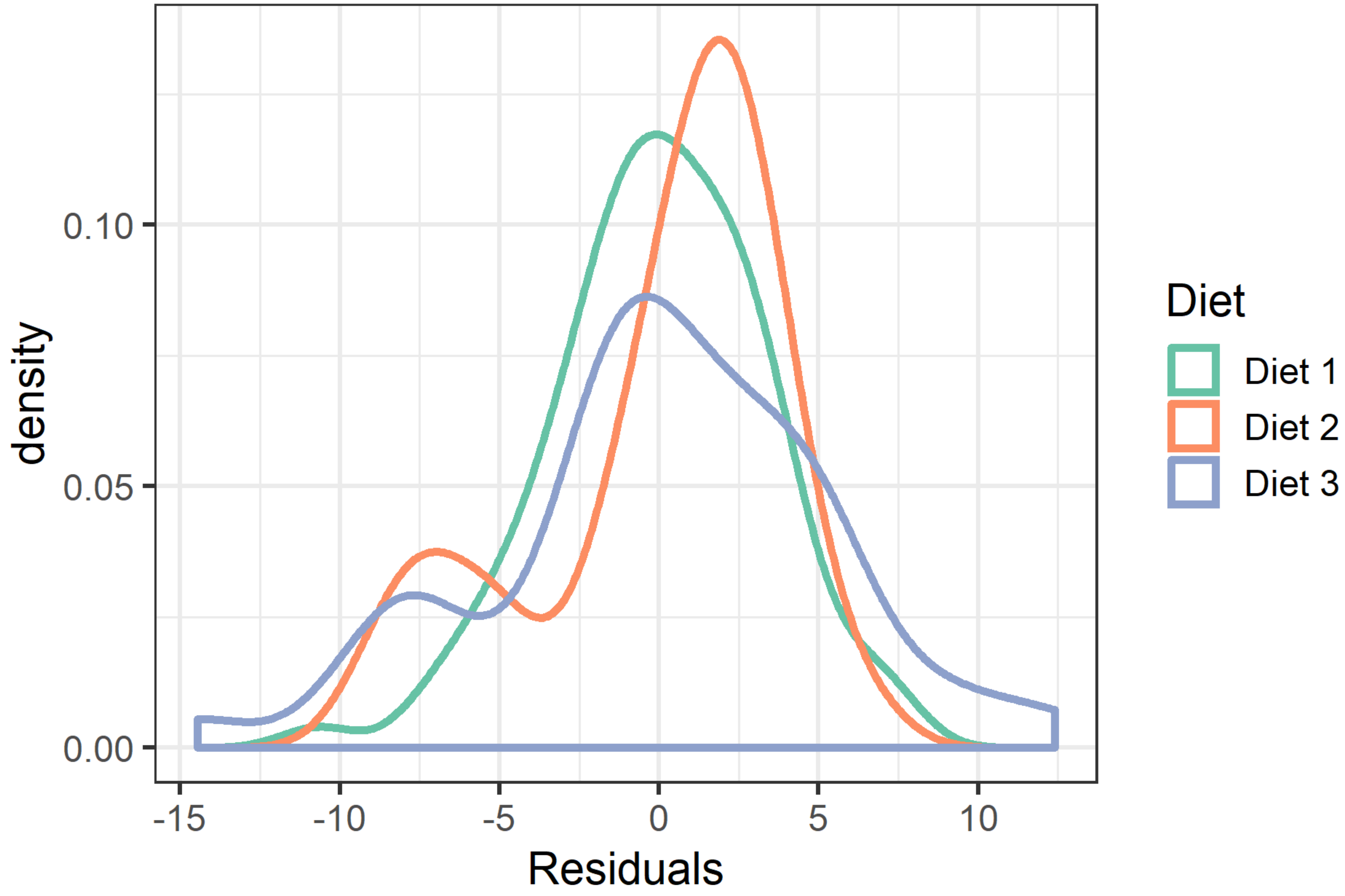
Checking assumptions

- Ensure model results are trustworthy
- Explore the distribution of residuals and random effects
- Residuals

```
body_weight <- lmer(weight ~ 1 + Time + diet_f + diet_f:Time +
    (1 + Time  | Rat), data = BodyWeight)

BodyWeight <- BodyWeight %>%
    mutate(model_residuals = residuals(body_weight))

ggplot(BodyWeight, aes(x = model_residuals)) +
    geom_density(aes(color = diet_f), size = 1.25) +
    xlab("Residuals") +
    theme_bw(base_size = 14) +
    scale_color_brewer(palette = "Set2")
```


density
0.10
0.05
0.00
-15
-10
-5
0
5
10
Residuals
Diet
Diet 1
Diet 2
Diet 3

Distribution of random effects

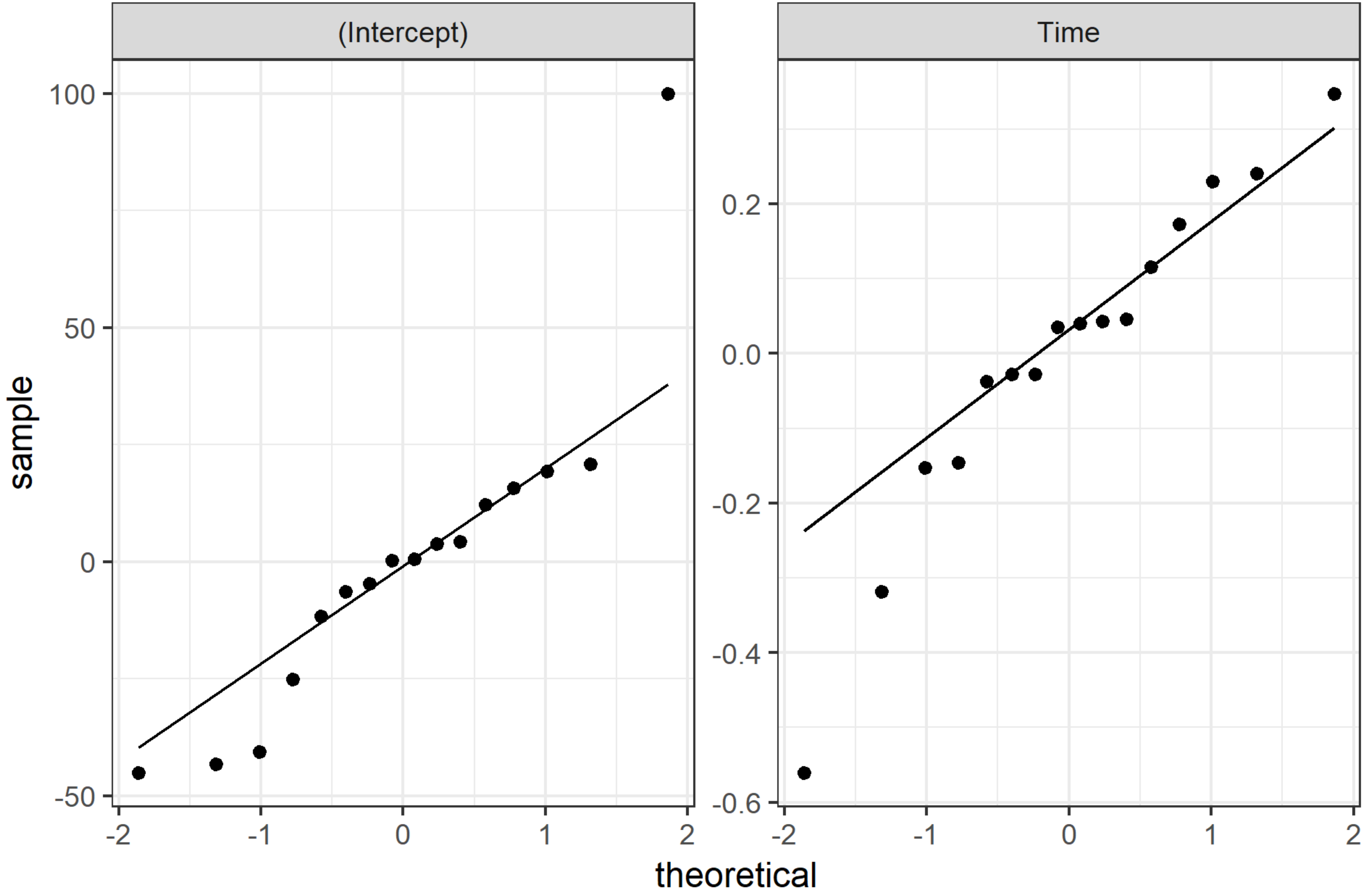
- The `ranef()` function can extract random effects

```
body_weight <- lmer(weight ~ 1 + Time + diet_f + diet_f:Time +
     (1 + Time  | Rat), data = BodyWeight)

random_effects <- ranef(body_weight)$Rat %>%
    mutate(id = 1:n()) %>%
    gather("variable", "value", -id)
```

- Quantile-quantile plots to evaluate normality assumption

```
ggplot(random_effects, aes(sample = value)) +
    geom_qq() +
    geom_qq_line() +
    facet_wrap(~variable, scales = 'free_y') +
    theme_bw(base_size = 14)
```


(Intercept)
Time
sample
theoretical
100
50
0
-50
0.2
0.0
-0.2
-0.4
-0.6
-2
-1
0
1
2
-2
-1
0
1
2

LONGITUDINAL ANALYSIS IN R

Let's practice!

LONGITUDINAL ANALYSIS IN R

Model Comparisons and Explained Variance

Brandon LeBeau
Assistant Professor

Model comparisons

- `anova()` function was used in chapter 2
  - AIC was recommended for model comparisons
  - AIC may lead to overfitting in small samples
- The corrected Akaike's information criterion (AICc) corrects for small samples
  - AICc converges to AIC with large samples

Model for comparison

```
library(nlme)
library(dplyr)
library(lme4)

BodyWeight <- BodyWeight %>%
    mutate(Time = Time - 1,
           diet_f = paste("Diet", Diet, sep = " "))

body_weight_rs <- lmer(weight ~ 1 + Time +
    (1 + Time  | Rat), data = BodyWeight, REML = FALSE)

body_weight_diet <- lmer(weight ~ 1 + Time + diet_f +
    (1 + Time  | Rat), data = BodyWeight, REML = FALSE)

body_weight_diet_int <- lmer(weight ~ 1 + Time + diet_f + diet_f:Time +
    (1 + Time  | Rat), data = BodyWeight, REML = FALSE)
```

AICc comparisons

```
library(AICcmodavg)

aictab(list(body_weight_rs, body_weight_diet, body_weight_diet_int),
  modnames = c('random slope', 'diet intercept', 'diet interaction'))
```

- Output

```
Model selection based on AICc:

                  K    AICc Delta_AICc AICcWt Cum.Wt      LL
diet interaction 10 1187.19       0.00   0.98   0.98 -582.93
diet intercept    8 1195.08       7.89   0.02   1.00 -589.11
random slope      6 1226.20      39.01   0.00   1.00 -606.85
```

Explained variance

- Explained variance can help evaluate the model
- Represents the ratio of explained variance to total variance
  - Larger values are better
- `MuMIn` package and `r.squaredGLMM()` function can be used for calculation

Explained variance example

- No predictors

```
library(MuMIn)

r.squaredGLMM(body_weight_rs)
             R2m      R2c
[1,] 0.007616649 0.998848
```

- Diet intercept only

```
r.squaredGLMM(body_weight_diet)
           R2m      R2c
[1,] 0.9109981 0.998757
```

- Diet interaction

```
r.squaredGLMM(body_weight_diet_int)
          R2m       R2c
[1,] 0.915638 0.9987975
```

LONGITUDINAL ANALYSIS IN R

Practice time!