Hierarchical Linear Models

Mixed Models, and Multilevel Models, GLMMs, etc...

AUTHORS
Camila Pacheco
Katrín Björnsdóttir

Today

- What are hierarchical linear models
- Identify situations in which the use of mixed effects is appropriate
- Implement basic linear mixed models (LMM) with `R`
- Break ☻
- Practice

Required Material

You are required to have downloaded and installed

```
        install.packages(c("lme4",
                           "ggeffects",
                           "stargazer"
                           ))
```

Required Material

Do not hesitate to ask questions!

Ecological and biological data can be complex!

- Hierarchical structure in the data
- Many covariates and grouping factors
- Unbalanced study/experimental design

What is Independence Assumption?

A linear regression model assumes that :

Any other data point does not influence each data point in a dataset

- We are **NOT** referring to your independent and responses variable
- *Ideally* we want them to be correlated it

We are referring withing the variable

Diagnosis of dependence?

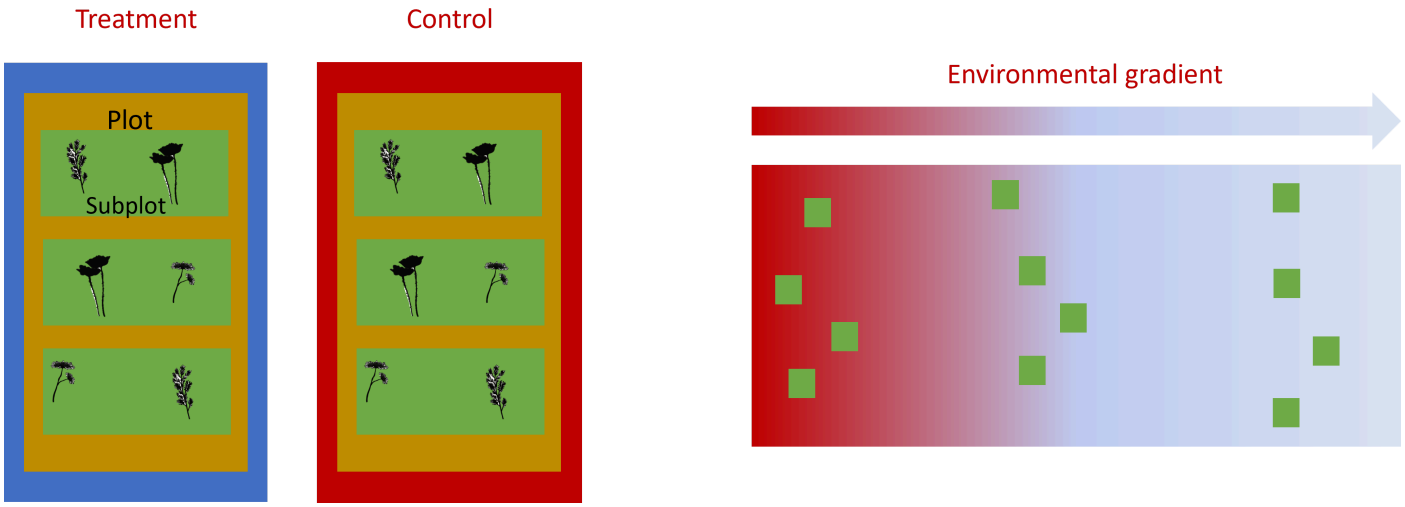
Think about how the data were collected.

Are there clusters in my data?

For example:

Are there data from individual species clustered within sampling area?

Hierarchical structres example



Hierarchical structres in ecology

- Grouping factors: populations, species, sites
- Sample sizes: Species area relationship
- Time: Might take repeated measurements of the same plant in time
- Space: the closer the similar

How could we analyze this data?

We need to explicitly account for the correlated nature of the data

The *random effects* structure will aid correct inference about *fixed effects*, depending on which level of the system's hierarchy is being manipulated.

- What is random effects ?
- What is fixed effects?
- When to used them ?

When to used them ?

You will need to used random and fixed effects every time your data has a **Hierarchical** structure.

- It's important to not that this difference has little to do with the variables themselves, and a lot to do with your research question!
- What is just variation ("**noise**") that you need to control for?

Fixed effects : deterministic processes

- They are like drivers, categories or groups that you think might directly influence the outcome you're studying.
- They're called "*fixed*" because you're interested in the specific levels of these categories.
- `"I want to know how these different things (fixed effects) affect the outcome."`
- levels of a factor (qualitative variable)
- a predictor (quantitative variable)

Random effects : stochastic processes

- Name random doesn't have much to do with mathematical randomness
- They are like grouping factors for which we are trying to control
- You're not interested in the specific effect of these factors themselves
- but we know that they might be influencing the patterns we see
- `"I want to know how these factors (random effects) contribute to the overall variation in the outcome."`
- They are always categorical

Data for this part of the session

Coding Club Mixed models

Explore the data

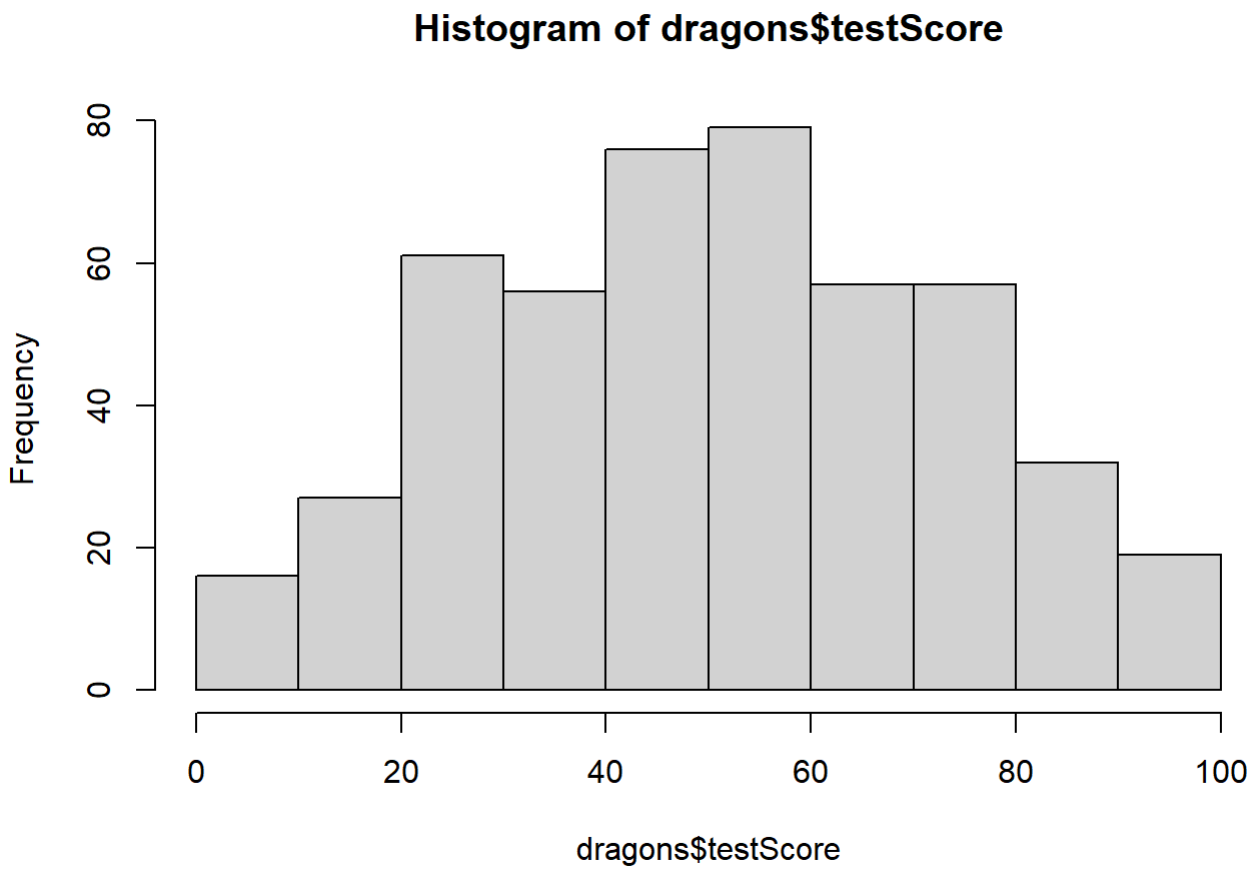
```
library(tidyverse)
```

```
── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
✓ dplyr     1.1.4     ✓ readr     2.1.5
✓ forcats   1.0.0     ✓ stringr   1.5.1
✓ ggplot2   3.5.0     ✓ tibble    3.2.1
✓ lubridate 1.9.3     ✓ tidyr     1.3.1
✓ purrr     1.0.2
── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become
errors
```

```r
load("CC-Linear-mixed-models/dragons.RData")
head(dragons)
```

```
  testScore bodyLength mountainRange site
1 16.147309   165.5485      Bavarian    a
2 33.886183   167.5593      Bavarian    a
3  6.038333   165.8830      Bavarian    a
4 18.838821   167.6855      Bavarian    a
5 33.862328   169.9597      Bavarian    a
6 47.043246   168.6887      Bavarian    a
```

```r
hist(dragons$testScore)
```



Scaling the data

It is good practice to standardize your explanatory variables before proceeding so that they have a mean of zero (“centering”) and standard deviation of one (“scaling”).

If two variables in the same model have very different scales, the mixed model will likely return a `convergence error` when trying to compute the parameters.

```
dragons<- dragons |>
  mutate(bodyLength2=scale(bodyLength,center = TRUE, scale = TRUE))
```

Research question

Is the test score affected by body length?

Linear model

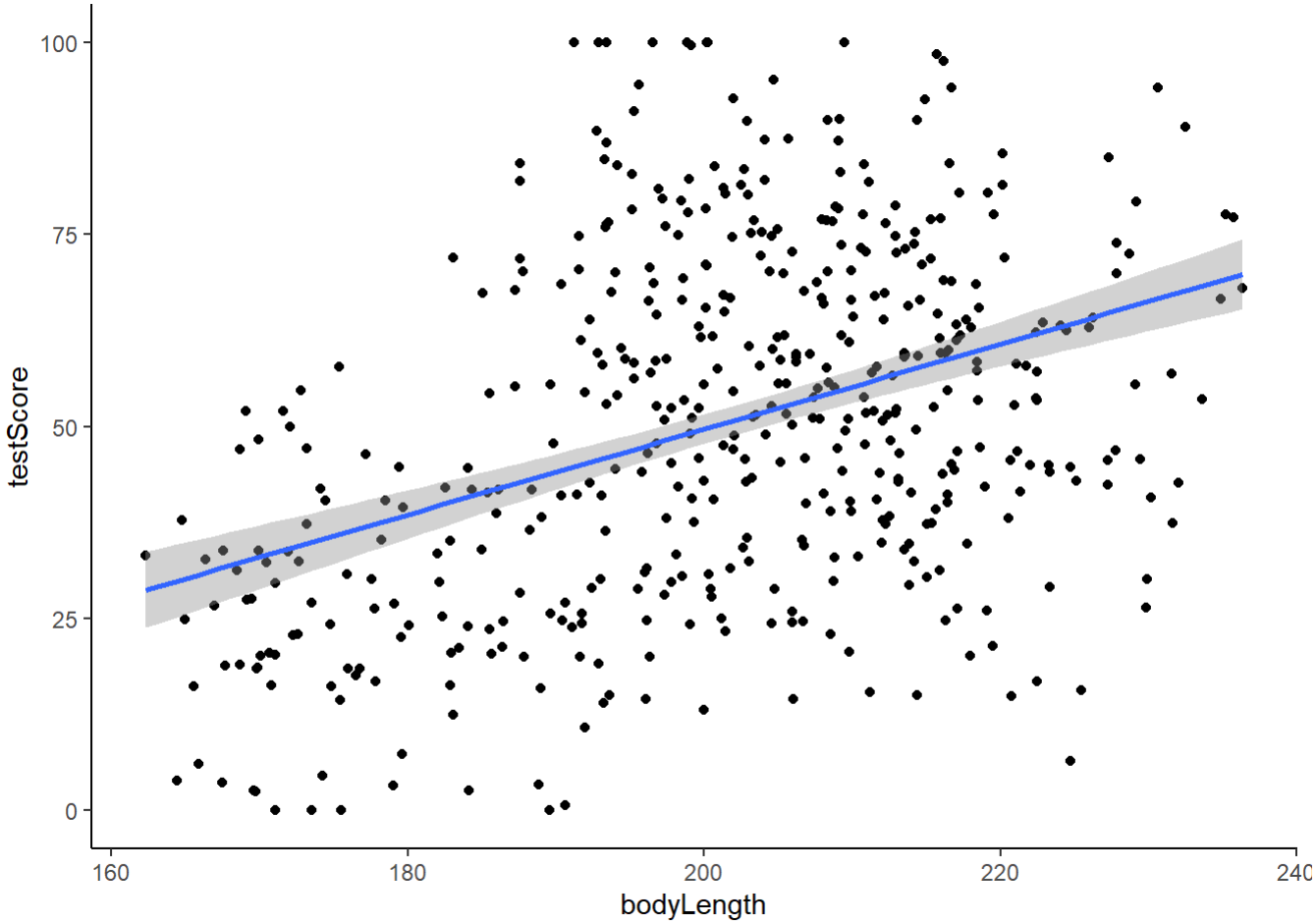
```
basic.lm <- lm(testScore ~ bodyLength2, data = dragons)
summary(basic.lm)
```

```
Call:
lm(formula = testScore ~ bodyLength2, data = dragons)

Residuals:
    Min      1Q  Median      3Q     Max
-56.962 -16.411  -0.783  15.193  55.200

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  50.3860     0.9676  52.072   <2e-16 ***
bodyLength2   8.9956     0.9686   9.287   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 21.2 on 478 degrees of freedom
Multiple R-squared:  0.1529,    Adjusted R-squared:  0.1511
F-statistic: 86.25 on 1 and 478 DF,  p-value: < 2.2e-16
```

Linear model

```
(prelim_plot <- ggplot(dragons, aes(x = bodyLength, y = testScore)) +
  geom_point() +
  geom_smooth(method = "lm"))+
  theme_classic()
```

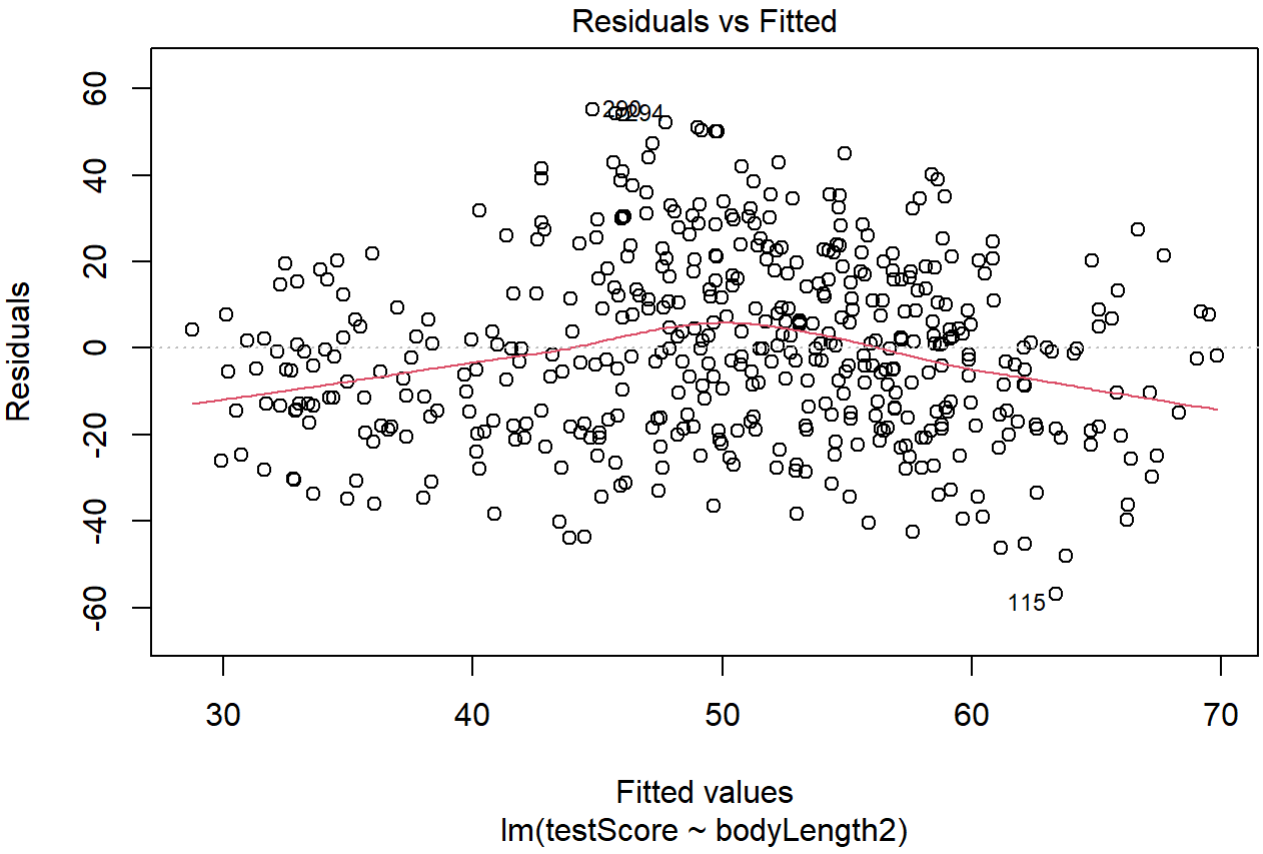
```
`geom_smooth()` using formula = 'y ~ x'
```

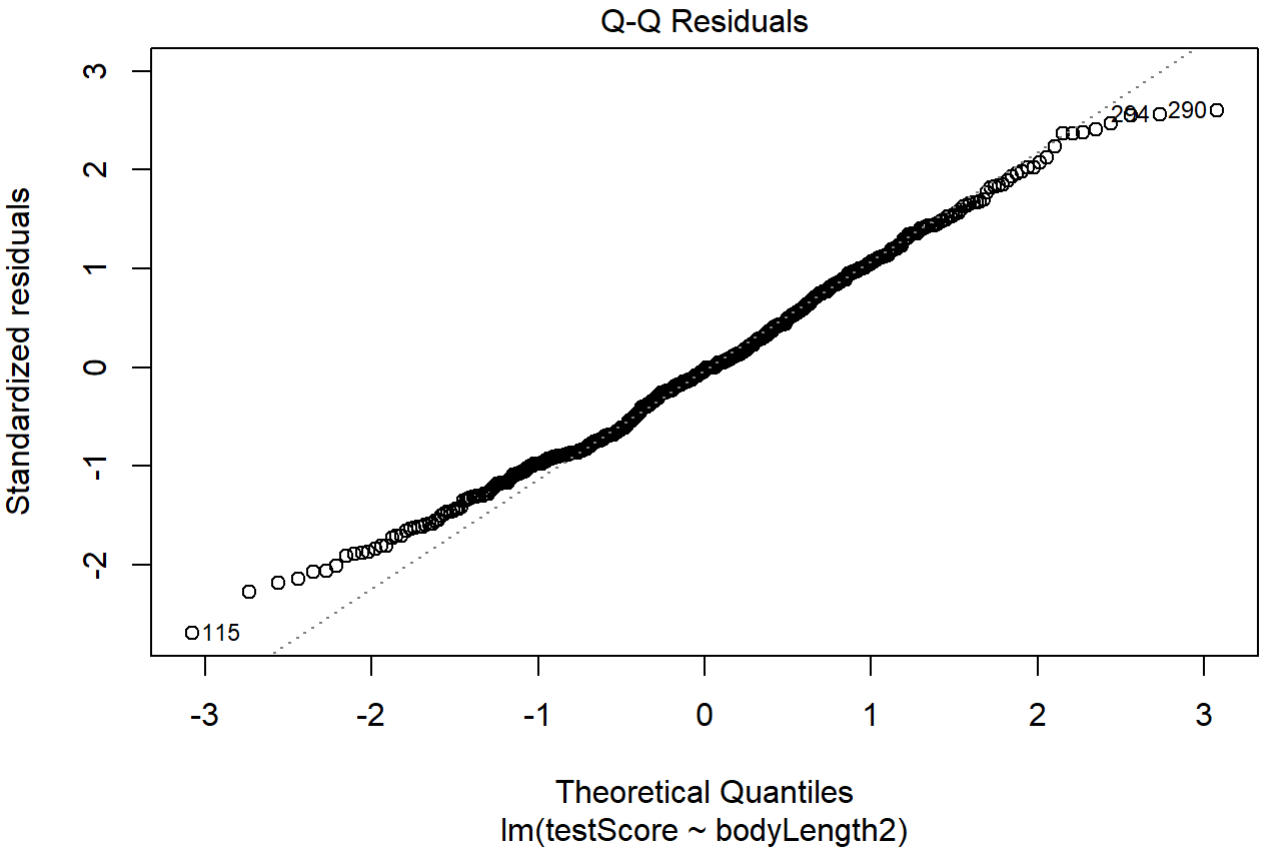


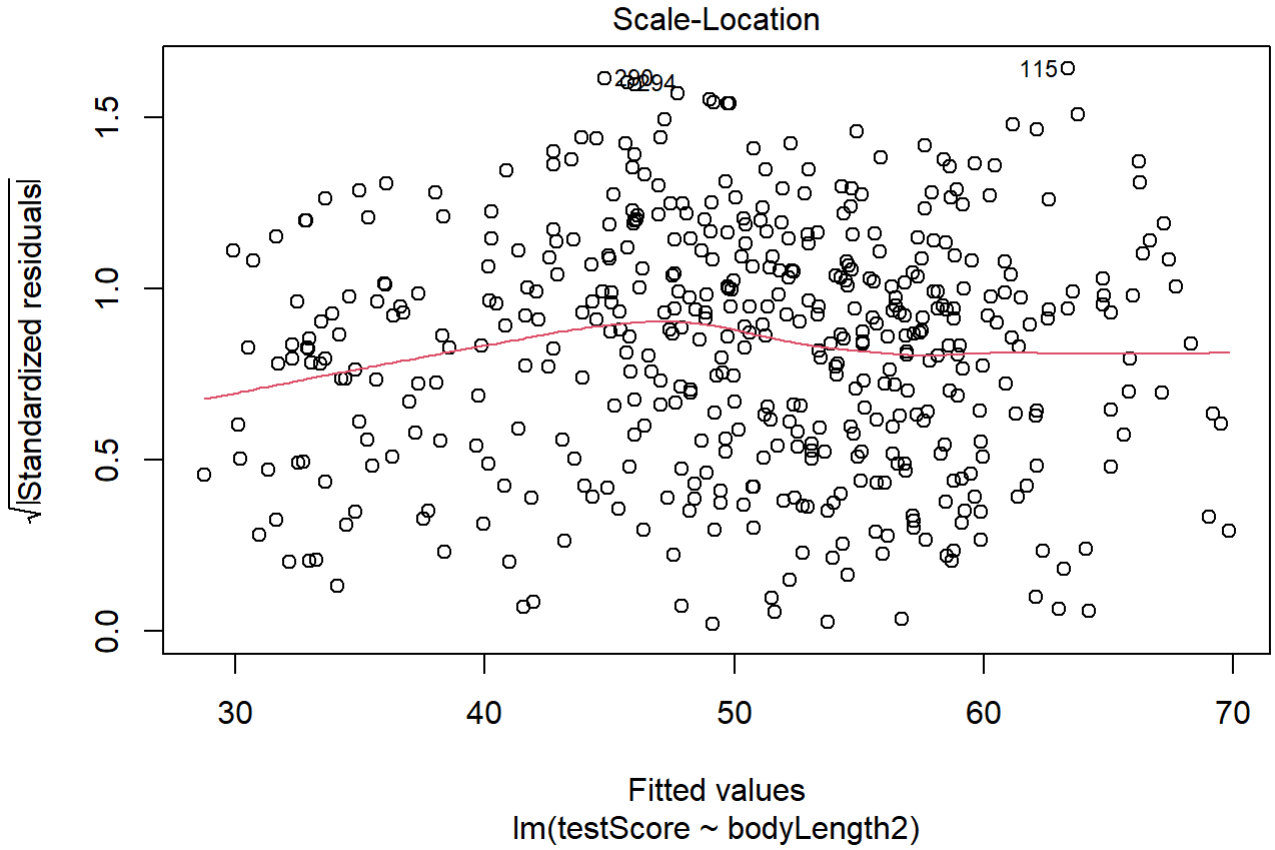
- it seems like bigger dragons do better in our intelligence test. That seems a bit odd: size shouldn't really affect the test scores.

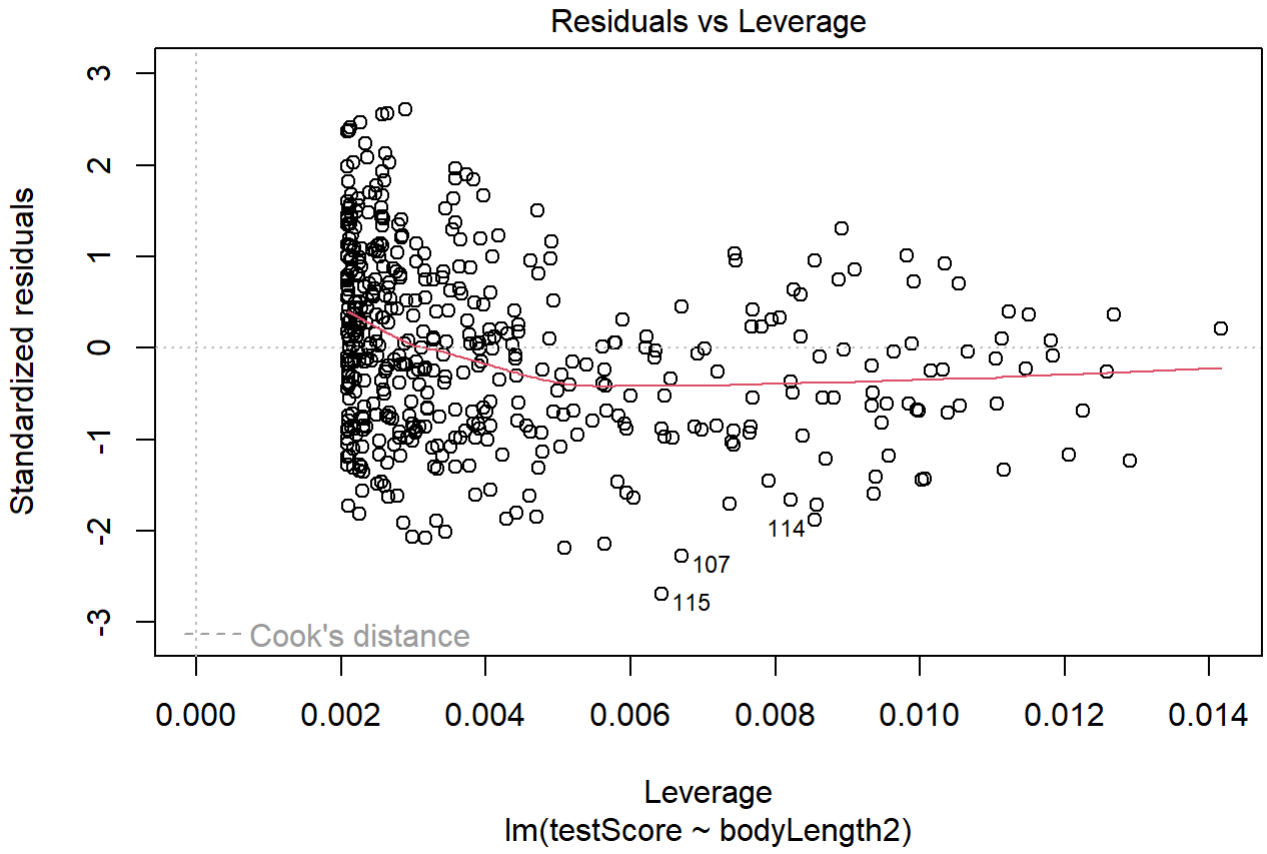
Assumptions check

```
plot(basic.lm)
```


Residuals vs Fitted
Residuals
60
40
20
0
-20
-40
-60
30
40
50
60
70
Fitted values
lm(testScore ~ bodyLength2)
115


Q-Q Residuals
Standardized residuals
3
2
1
0
-1
-2
115
294
290
-3
-2
-1
0
1
2
3
Theoretical Quantiles
lm(testScore ~ bodyLength2)


Scale-Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
30
40
50
60
70
Fitted values
lm(testScore ~ bodyLength2)



Assumptions check

Are our data independent?

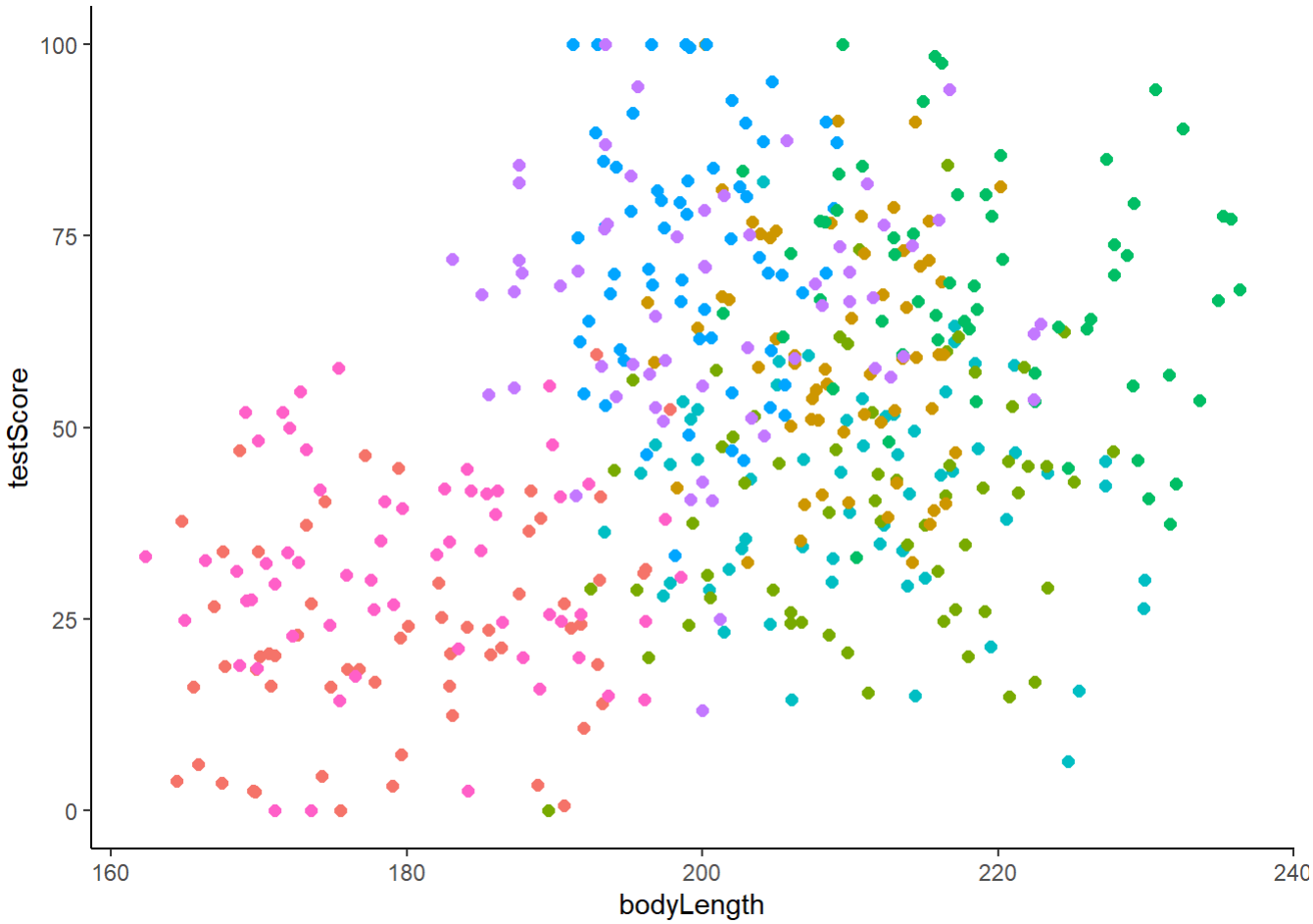
Assumptions check

`Data description:` The data were collected from multiple samples from eight mountain ranges.

- It's perfectly plausible that the data from within each mountain range are more similar to each other than the data from different mountain ranges
- they are Hierarchical!

Assumptions check

```
(colour_plot <- ggplot(dragons, aes(x = bodyLength, y = testScore, colour = mountainRange
  geom_point(size = 2) +
  theme_classic() +
  theme(legend.position = "none"))
```



How to implement mixed models in R?

- Step 1: Model building
- Step 2: Model validation
- Step 3: Model interpretation
- Step 4: Model visualization

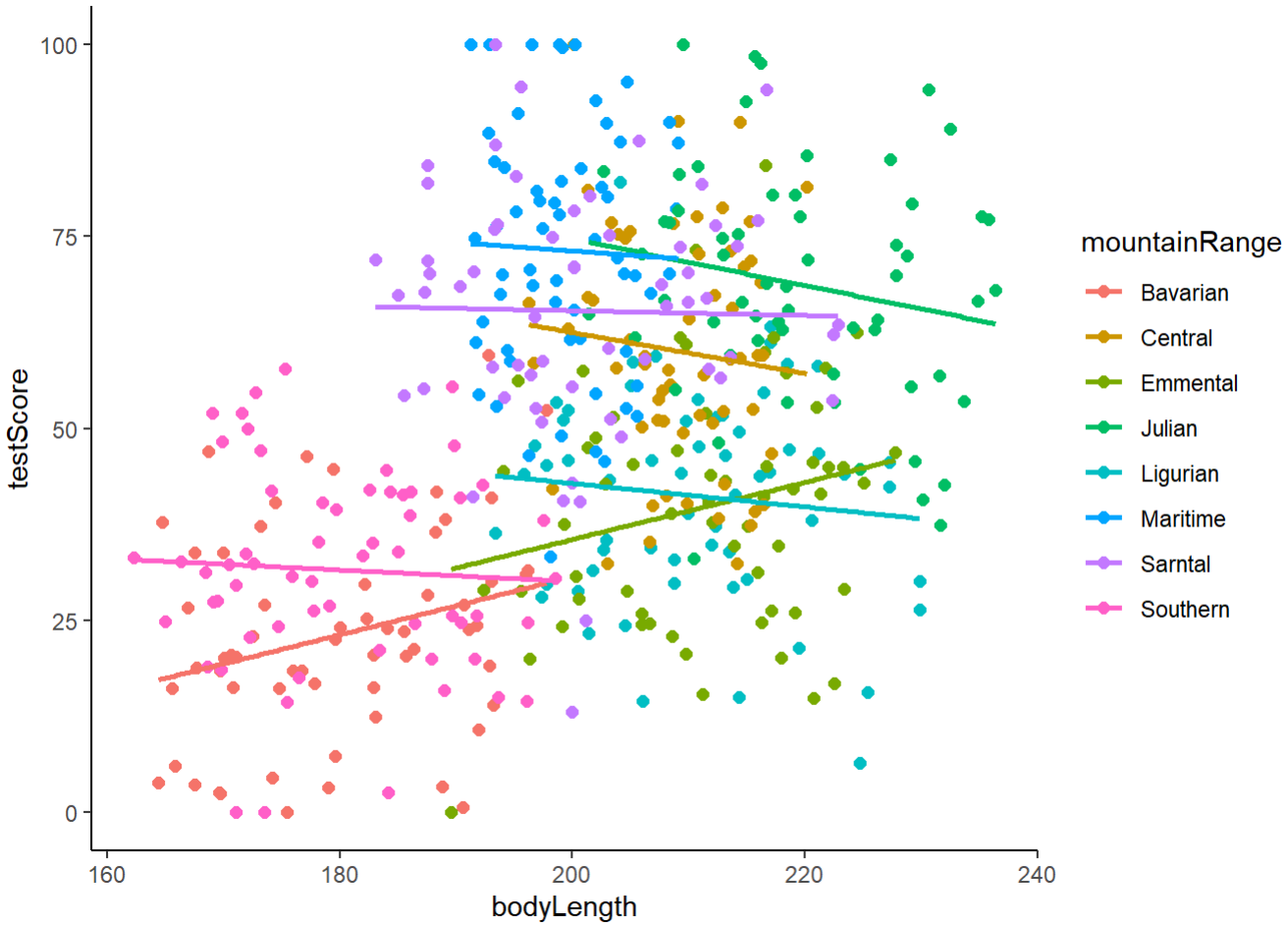
Step 1: Model building

Hierarchical linear models do is they essentially fit a separate regression line for each and every cluster.

Step 1: Model building Dragons

```
(colour_plot <- ggplot(dragons, aes(x = bodyLength, y = testScore, colour = mountainRange
  geom_point(size = 2) +
  geom_smooth(method = "lm",se=FALSE)+
  theme_classic() )
```

```
`geom_smooth()` using formula = 'y ~ x'
```



Step 1: Model building

Hierarchical linear models do is they essentially fit a separate regression line for each and every cluster. And then estimates what we call the *Fixed slope*. Average slope between x and y across my clusters.

Mathematically speaking it is more complicated than that.

Step 1: Model building Dragons

```
1  library(lme4) #"linear mixed model" function from lme4 package
```

```
Loading required package: Matrix


Attaching package: 'Matrix'

The following objects are masked from 'package:tidyr':

    expand, pack, unpack
```

```
mixed.lmer <- lmer(testScore ~ bodyLength2 +
                     (1|mountainRange), #random effect
                   data = dragons,
                   REML = TRUE #estimation method other method ML but it has a bias
                   )
summary(mixed.lmer)
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: testScore ~ bodyLength2 + (1 | mountainRange)
   Data: dragons

REML criterion at convergence: 3985.6

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.4815 -0.6513  0.0066  0.6685  2.9583

Random effects:
 Groups        Name        Variance Std.Dev.
 mountainRange (Intercept) 339.7    18.43
 Residual                  223.8    14.96
Number of obs: 480, groups:  mountainRange, 8

Fixed effects:
            Estimate Std. Error t value
(Intercept)  50.3860     6.5517   7.690
bodyLength2   0.5377     1.2750   0.422

Correlation of Fixed Effects:
            (Intr)
bodyLength2 0.000
```

Step 1: Model building Dragons

Mountain ranges are clearly important: they explain a lot of variation:

Factor	Variance
Mountain range	339.7
Residuals	223.8
Body length	?

```
(339.7/(339.7 + 223.8) )* 100
```

```
[1] 60.28394
```

Variance Body length

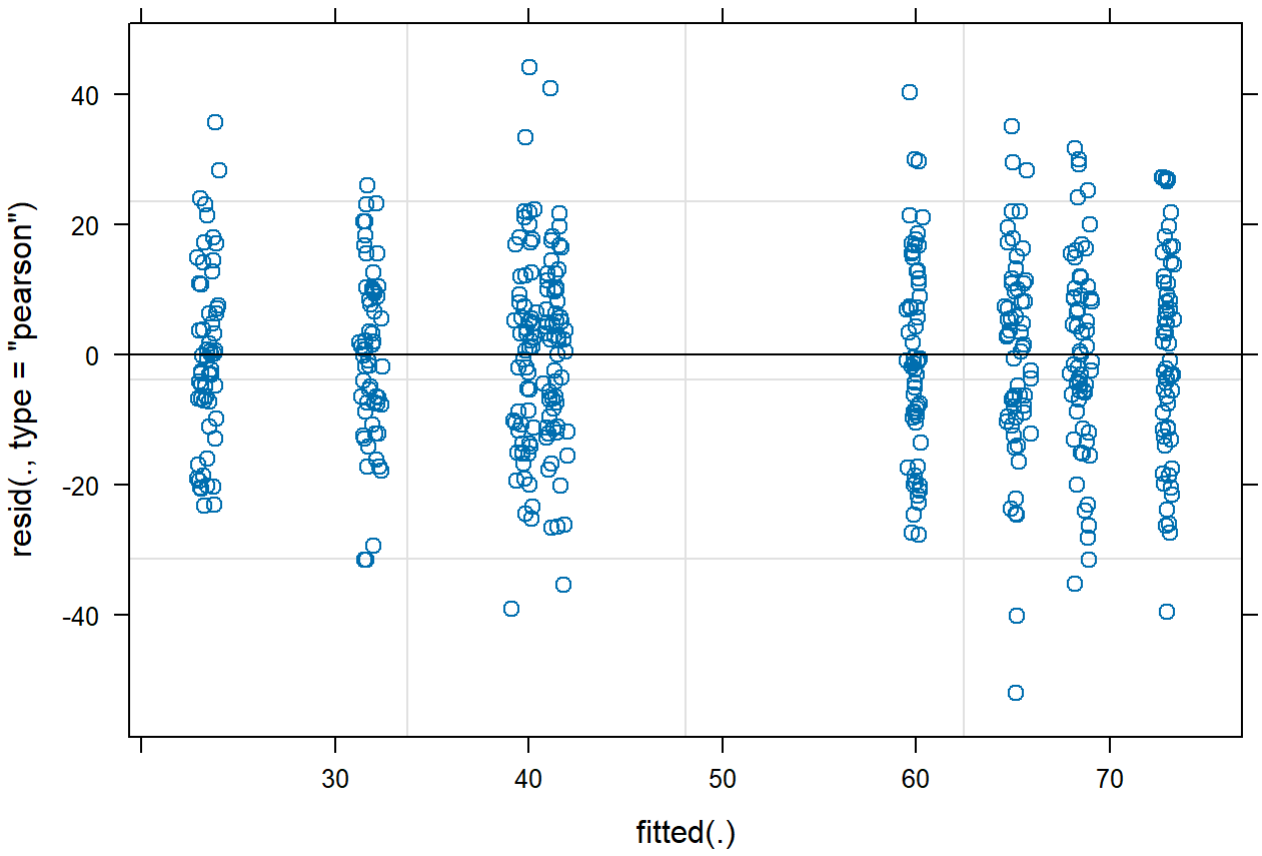
```
100-60.28
```

```
[1] 39.72
```

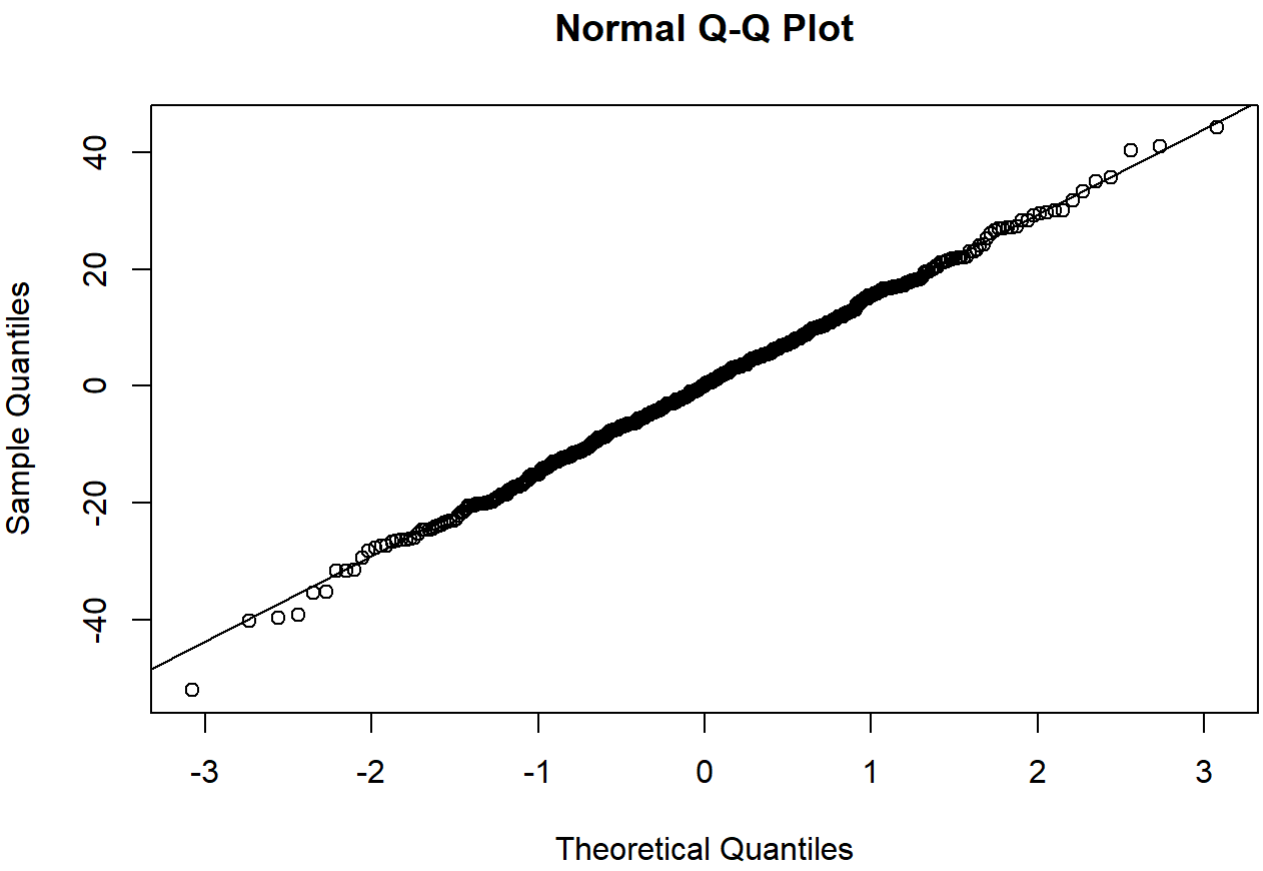
Step 2: Model validation Dragons

Assumptions check

```
plot(mixed.lmer)
```




```
qqnorm(resid(mixed.lmer))
qqline(resid(mixed.lmer))
```



Step 1: Model building Nesting

Example:

10 control | 10 experimental

- 3 years
- Each season
- 20 beds
- 50 seedlings
- 5 leaves
- 5 leaves x 50 seedlings x 20 beds x 4 seasons x 3 years = 60 000 measurements per treatment

Step 1: Model building Nesting

Effect of treatment in leaf length

```
leafLength ~ treatment
```

- *Pseudoreplication*
- Massively increasing sampling size

Better model

```
leafLength ~ treatment + (1|Bed/Plant/Leaf)
```

What about the crossed effects ?

- Crossed (or partially crossed) random factors that do not represent levels in a hierarchy.
- This account for the fact that all plants in the experiment, regardless of the fixed (treatment) effect, may have experienced a very hot summer in the second year.

```
leafLength ~ treatment + (1|Bed/Plant/Leaf) + (1|Season)
```

Step 1: Model building Dragons Nesting

```
mixed.lmer2 <- lmer(testScore ~ bodyLength2+
                      (1|mountainRange/site),
                    data = dragons)
summary(mixed.lmer2)
```

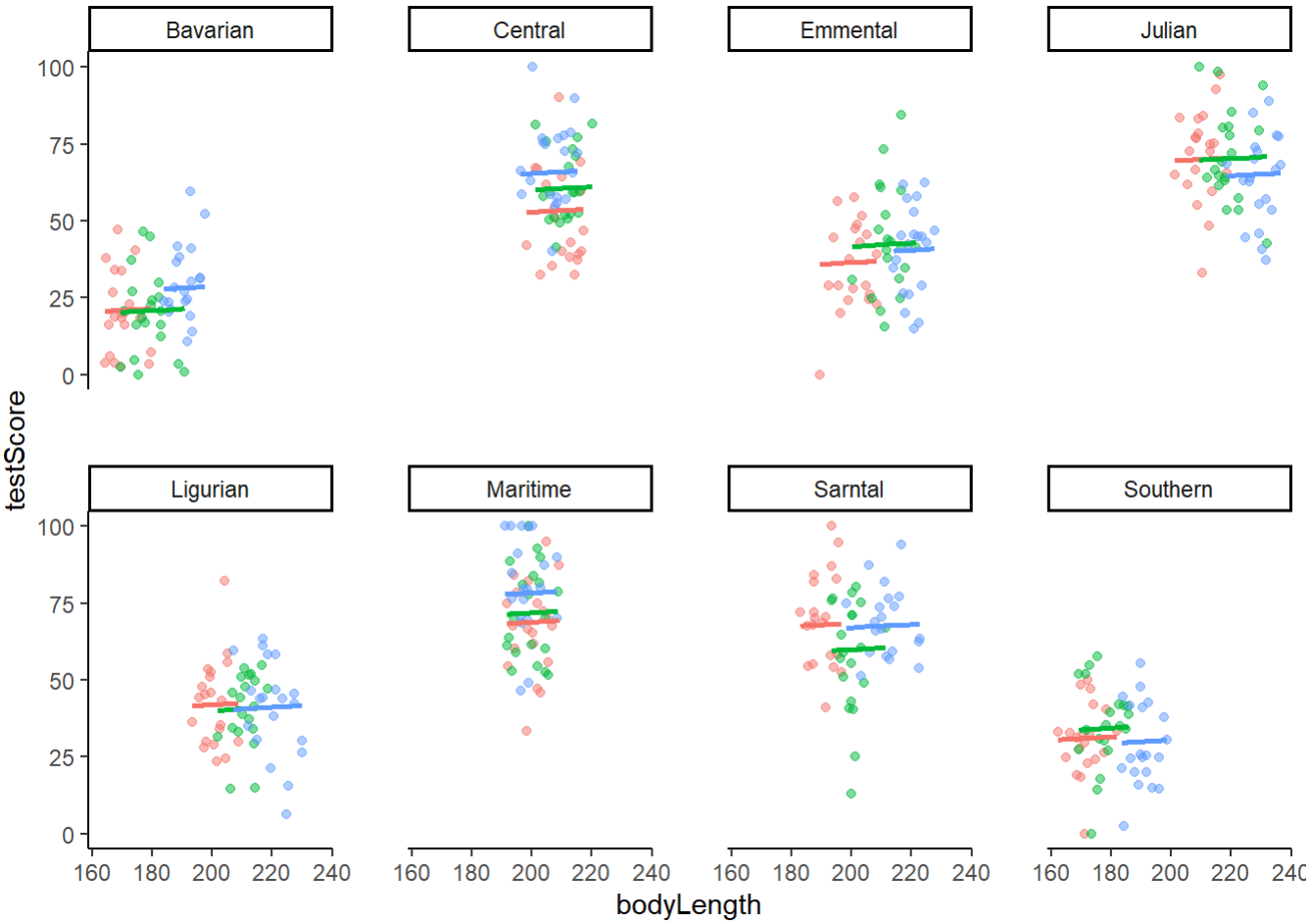
```
Linear mixed model fit by REML ['lmerMod']
Formula: testScore ~ bodyLength2 + (1 | mountainRange/site)
   Data: dragons

REML criterion at convergence: 3970.4

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.2425 -0.6752 -0.0117  0.6974  2.8812

Random effects:
 Groups              Name        Variance Std.Dev.
 site:mountainRange (Intercept)  23.09    4.805
 mountainRange      (Intercept) 327.56   18.099
 Residual                       208.58   14.442
Number of obs: 480, groups:  site:mountainRange, 24; mountainRange, 8

Fixed effects:
            Estimate Std. Error t value
(Intercept)   50.386      6.507   7.743
bodyLength2    0.831      1.681   0.494

Correlation of Fixed Effects:
            (Intr)
bodyLength2 0.000
```

```r
mixed.lmer3 <- lmer(testScore ~ bodyLength2
                    +(1|mountainRange) + (1|mountainRange:site),
                    data = dragons)
summary(mixed.lmer3)
```

```
Linear mixed model fit by REML ['lmerMod']
Formula:
testScore ~ bodyLength2 + (1 | mountainRange) + (1 | mountainRange:site)
   Data: dragons

REML criterion at convergence: 3970.4

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.2425 -0.6752 -0.0117  0.6974  2.8812

Random effects:
 Groups             Name        Variance Std.Dev.
 mountainRange:site (Intercept)  23.09    4.805
 mountainRange      (Intercept) 327.56   18.099
 Residual                       208.58   14.442
Number of obs: 480, groups:  mountainRange:site, 24; mountainRange, 8

Fixed effects:
            Estimate Std. Error t value
(Intercept)   50.386      6.507   7.743
bodyLength2    0.831      1.681   0.494

Correlation of Fixed Effects:
            (Intr)
bodyLength2 0.000
```

Step 1: Model building Dragons Nesting

```
Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
ℹ Please use `linewidth` instead.
```



Random slopes and Random intercept

A *random-intercept* model recognizes that each cluster might have its own starting point (intercept), but keeps the slope constant among them. So in our example we acknowledge that some populations may be smarter or dumber to begin with.

Lets say we expect that dragons in all mountain ranges do not exhibit the same relationship between body length and intelligence (rando, slope)

Random slopes and Random intercept

We only need to make one change to our model to allow for random slopes as well as intercept, and that's adding the fixed variable into the random effect brackets:

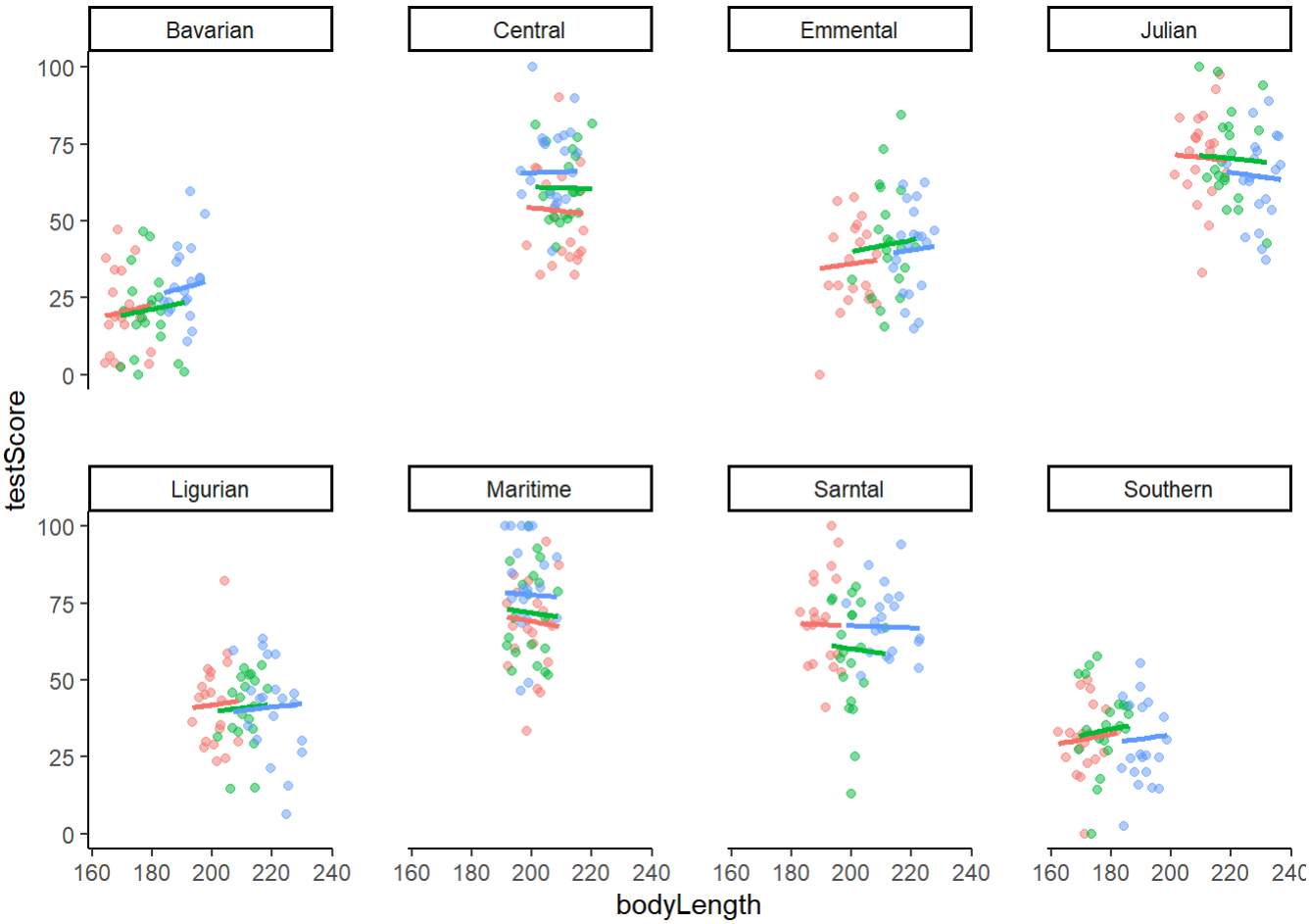
```
mixed.ranslope <- lmer(testScore ~ bodyLength2 +
                         (1 + bodyLength2|mountainRange/site),
                       data = dragons)
```

```
boundary (singular) fit: see help('isSingular')
```

```
summary(mixed.ranslope)
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: testScore ~ bodyLength2 + (1 + bodyLength2 | mountainRange/site)
   Data: dragons

REML criterion at convergence: 3968.4

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.2654 -0.6737 -0.0200  0.6931  2.8432

Random effects:
 Groups              Name        Variance Std.Dev. Corr
 site:mountainRange (Intercept)  19.8156  4.4515
                    bodyLength2   0.7178  0.8472  1.00
 mountainRange      (Intercept) 310.9691 17.6343
                    bodyLength2   6.1119  2.4722  -1.00
 Residual                       208.5025 14.4396
Number of obs: 480, groups:  site:mountainRange, 24; mountainRange, 8

Fixed effects:
            Estimate Std. Error t value
(Intercept)  51.4263     6.3408   8.110
bodyLength2   0.6691     1.8729   0.357

Correlation of Fixed Effects:
            (Intr)
bodyLength2 -0.461
optimizer (nloptwrap) convergence code: 0 (OK)
boundary (singular) fit: see help('isSingular')
```

Random slopes and Random intercept



Step 3: Model interpretation

```
library(stargazer)
```

```
Please cite as:

 Hlavac, Marek (2022). stargazer: Well-Formatted Regression and Summary Statistics Tables.

 R package version 5.2.3. https://CRAN.R-project.org/package=stargazer
```

```
stargazer(mixed.lmer2,
          digits = 3,
          type="text",
          star.cutoffs = c(0.05, 0.01, 0.001),
          digit.separator = "")
```

```
================================================
                        Dependent variable:
                    ----------------------------
```

```
                            testScore
-------------------------------------------------
bodyLength2                   0.831
                             (1.681)

Constant                     50.386***
                             (6.507)

-------------------------------------------------
Observations                   480
Log Likelihood               -1985.195
Akaike Inf. Crit.            3980.389
Bayesian Inf. Crit.          4001.258
=================================================
Note:               *p<0.05; **p<0.01; ***p<0.001
```

Step 4. Visualization

```r
library(ggeffects)

# Extract the prediction data frame
pred.mm <- ggpredict(mixed.lmer2, terms = c("bodyLength2"))  # this gives overall p
head(pred.mm)
```

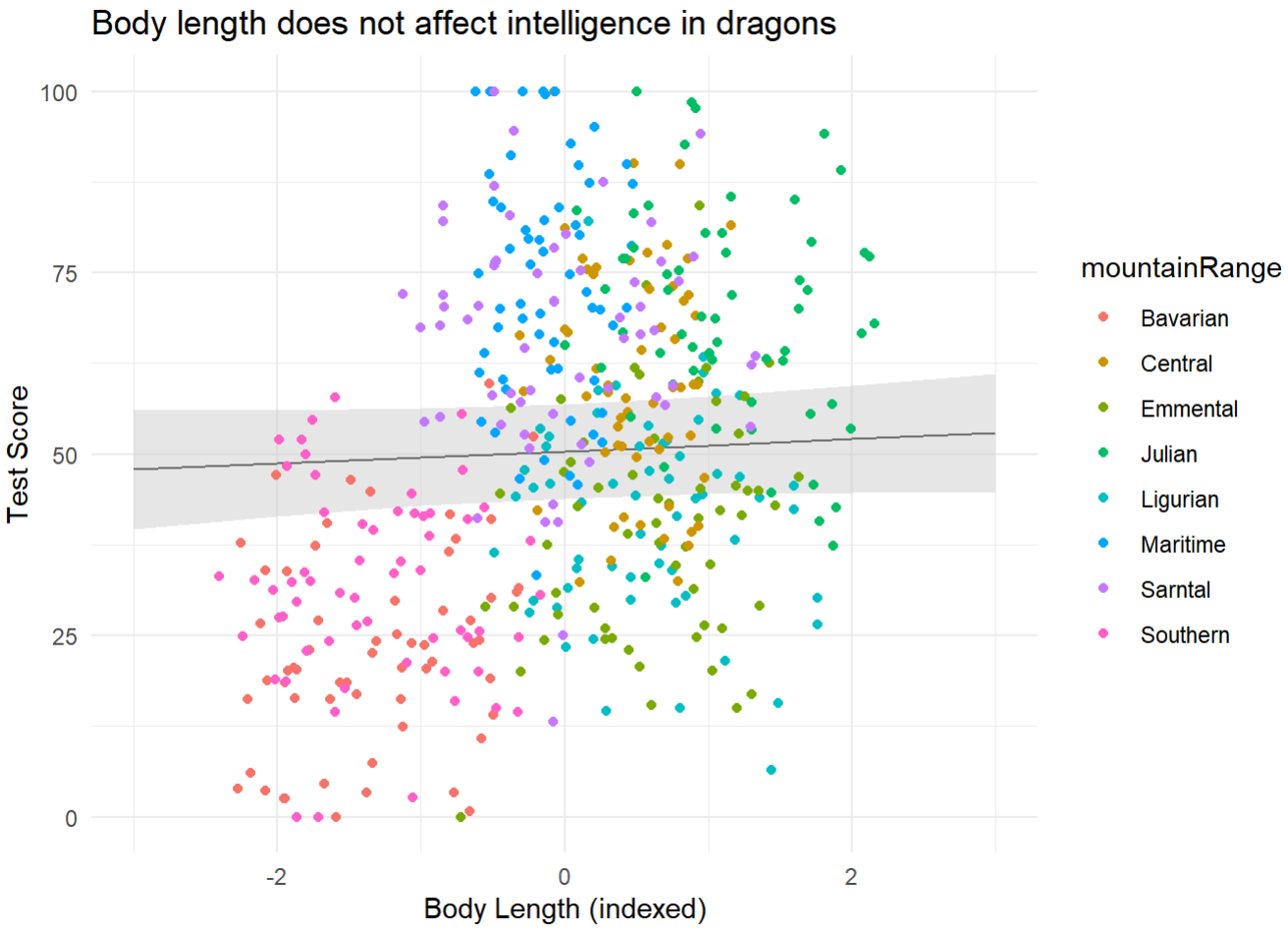
```
# Predicted values of testScore

bodyLength2 | Predicted |       95% CI
--------------------------------------
         -3 |     47.89 | 31.72, 64.07
         -2 |     48.72 | 34.33, 63.12
         -1 |     49.56 | 36.35, 62.76
          0 |     50.39 | 37.60, 63.17
          1 |     51.22 | 38.01, 64.42
          2 |     52.05 | 37.66, 66.44

Adjusted for:
*          site = 0 (population-level)
* mountainRange = 0 (population-level)
```

Step 4. Visualization

```r
# Plot the predictions
p<-ggplot(pred.mm) +
    # slope
    geom_line(aes(x = x, y = predicted)) +
    # error band
    geom_ribbon(
```

```
      aes(
        x = x,
        ymin = predicted - std.error,
        ymax = predicted + std.error
      ),
      fill = "lightgrey",
      alpha = 0.5
    ) +
    # adding the raw data (scaled values)
    geom_point(data = dragons,
               aes(x = bodyLength2, y = testScore, colour = mountainRange)) +
    labs(x = "Body Length (indexed)",
         y = "Test Score",
         title = "Body length does not affect intelligence in dragons") +
    theme_minimal()
```

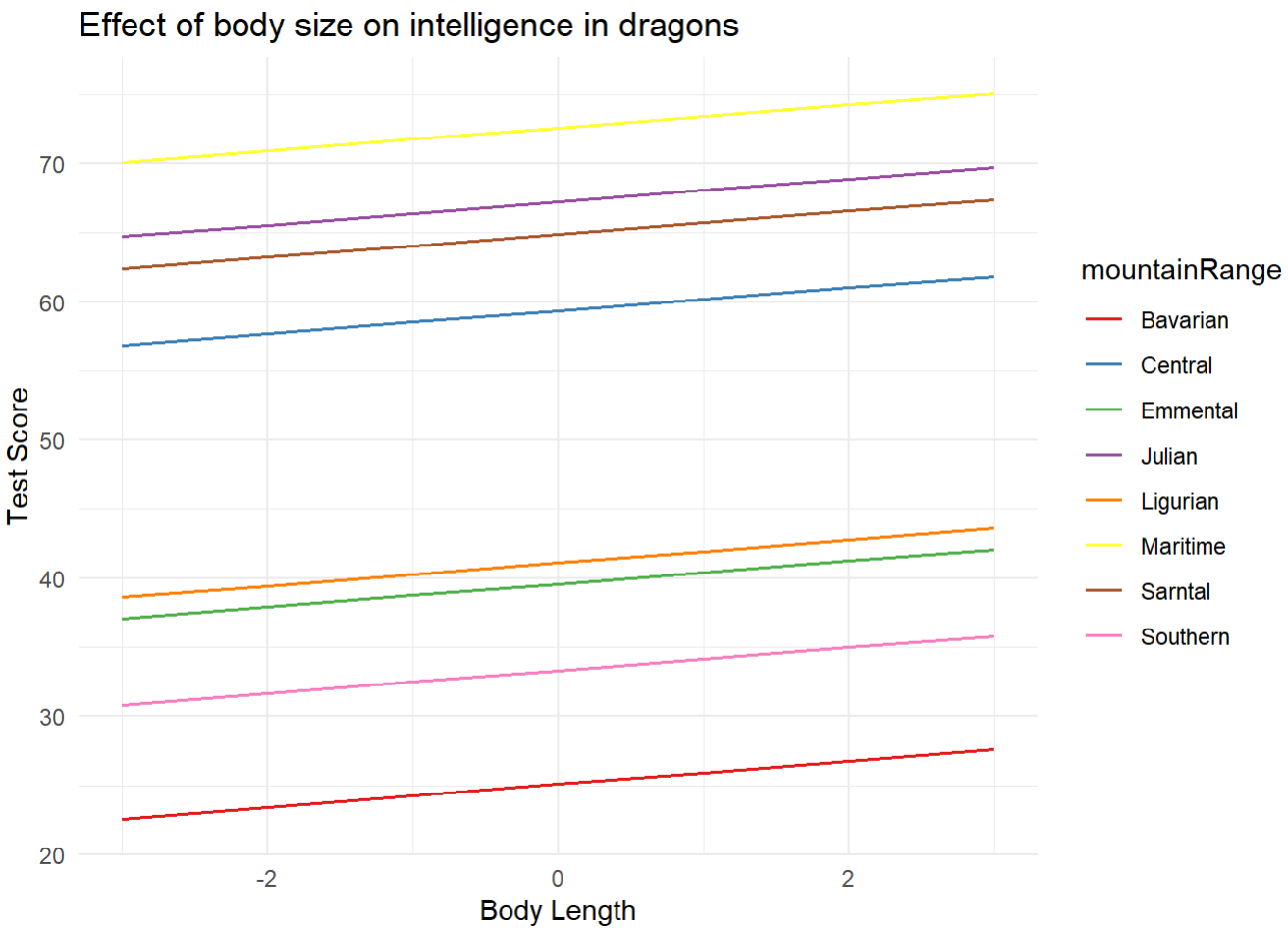
Step 4. Visualization



Step 4. Visualization

```r
p<-
  ggpredict(mixed.lmer2, terms = c("bodyLength2", "mountainRange"),
            type = "random") %>%
   plot(show_ci = FALSE) +
   labs(x = "Body Length", y = "Test Score",
        title = "Effect of body size on intelligence in dragons") +
   theme_minimal()
```

Step 4. Visualization



Additional ressources

Popular libraries for (G)LMMs:

- Frequentist : `nlme`, `lme4`, `glmmTMB`
- Bayesian : `brms`, `rstan`, `rstanarm`, `MCMCglmm`
- Nice visualization : link
- Visit: Coding Club

BREAK