

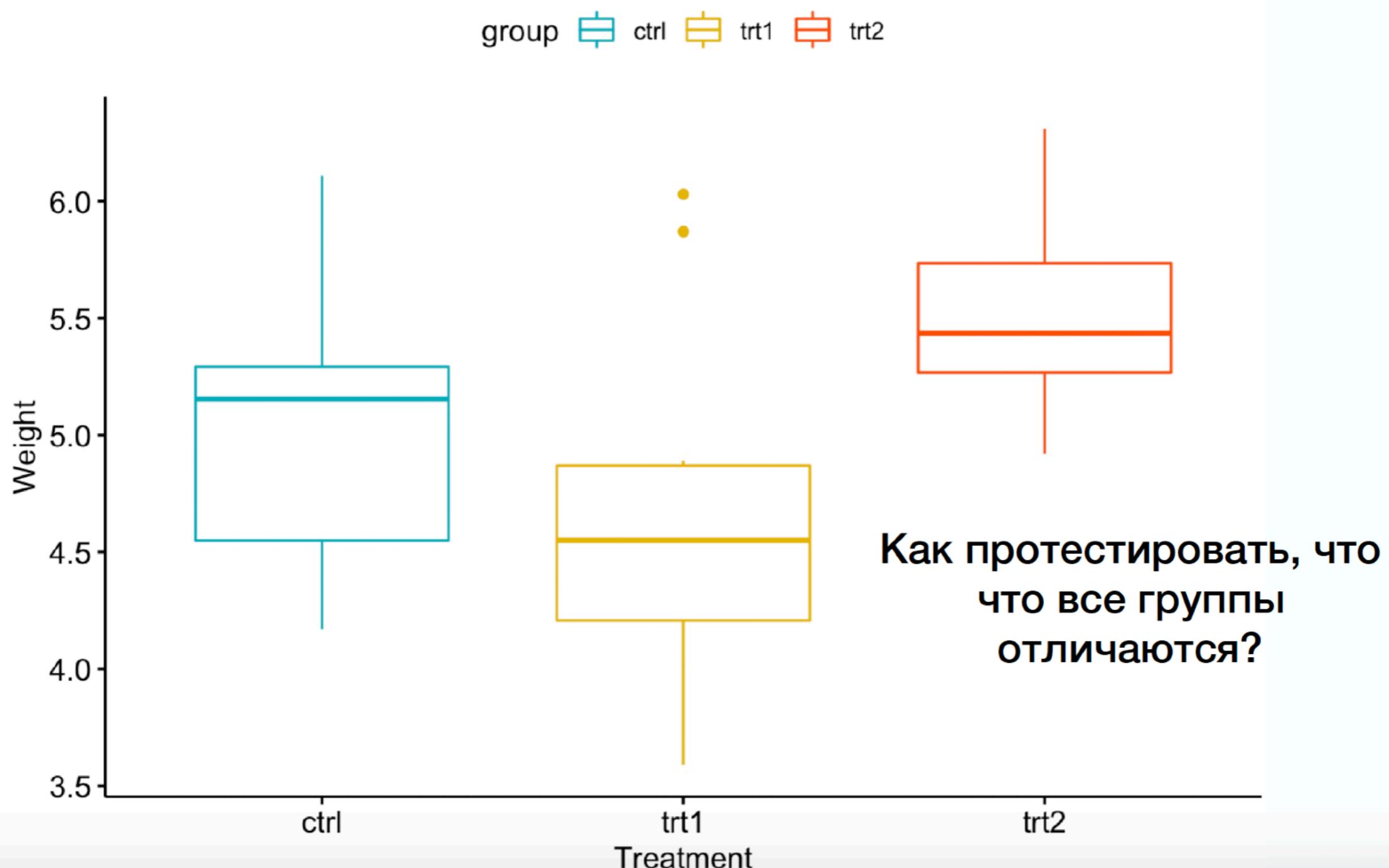
Anova.

Fixed effects model.

Random effects model

One-way ANOVA

```
library("ggpubr")
ggboxplot(PlantGrowth, x = "group", y = "weight",
          color = "group", palette = c("#00AFBB", "#E7B800", "#FC4E07"),
          order = c("ctrl", "trt1", "trt2"),
          ylab = "Weight", xlab = "Treatment")
```



1-way ANOVA

FactorA	A1	A2	A3
	y_{11}	y_{12}	y_{13}
	y_{21}

	y_{n1}		y_{nm}

1-way ANOVA

$$SST = SSX + SSE$$

$$SST = SSA + SSE$$

$$SST = SS_{among} + SS_{within}$$

$$SST = \sum_i \sum_j (y_{ij} - \bar{y})^2$$

$$SSA = \sum_i \sum_j (\bar{y}_{-j} - \bar{y})^2 = n \cdot \sum_j (\bar{y}_{-j} - \bar{y})^2$$

$$SSE = \sum_i \sum_j (y_{ij} - \bar{y}_{-j})^2$$

1-way ANOVA

$$SST = SSX + SSE$$

$$SST = SSA + SSE$$

$$SST = SS_{among} + SS_{within}$$

$$SST = \sum_i \sum_j (y_{ij} - \bar{y})^2 \text{ Сумма квадратов Y}$$

$$SSA = n \cdot \sum_j (\bar{y}_{-j} - \bar{y})^2 \text{ Сумма квадратов Y, объясняемая фактором A}$$

Фактически - сколько мы объясним, если будем предсказывать у только по j (значению фактора A)

$$SSE = \sum_i \sum_j (y_{ij} - \bar{y}_{-j})^2 \text{ Сумма квадратов Y, не объясняемая фактором A}$$

FactorA	A1	A2	A3
	\bar{y}_1	\bar{y}_2	\bar{y}_3
	\bar{y}_1

	\bar{y}_1	\bar{y}_2	\bar{y}_3

1-way ANOVA

$$H_0 : \mu_{A_1} = \mu_{A_2} \dots = \mu_{A_n} = \mu$$

H_1 : Не все средние равны

FactorA	A1	A2	A3
	y_{11}	y_{12}	y_{13}
	y_{21}

	y_{n1}		y_{nm}

1-way ANOVA

$$H_0 : \mu_{A_1} = \mu_{A_2} \dots = \mu_{A_n} = \mu$$

Если H_0 верна, то фактически, у нас имеется n выборок (по 1 выборке на значение фактора A).

В каждой мы можем оценить дисперсию. Что мы можем потом сделать?

1-way ANOVA

$$H_0 : \mu_{A_1} = \mu_{A_2} \dots = \mu_{A_n} = \mu$$

1) Если H_0 верна, то фактически, у нас имеется n выборок (по 1 выборке на значение фактора A).

В каждой мы можем оценить дисперсию. Что мы можем потом сделать?

Подсчитать pooled variance

$$s_{pooled}^2 = \frac{\sum_j (n_j - 1)s_j^2}{\sum_j (n_j - 1)} = \frac{SSE}{N - a} = MSE$$

$$s_j^2 = \frac{1}{n_j - 1} \sum_i (y_{ij} - \bar{y}_{-j})^2 \quad SSE = \sum_i \sum_j (y_{ij} - \bar{y}_{-j})^2$$

1-way ANOVA

$$H_0 : \mu_{A_1} = \mu_{A_2} \dots = \mu_{A_n} = \mu$$

1) Если H_0 верна, то фактически, у нас имеется n выборок (по 1 выборке на значение фактора A)

В каждой мы можем оценить дисперсию. Что мы можем потом сделать?

Подсчитать pooled variance

$$\sigma^2 \approx s_{pooled}^2 = \frac{\sum_j (n_j - 1)s_j^2}{\sum_j (n_j - 1)} = \frac{SSE}{N - a} = MSE$$

$$s_j^2 = \frac{1}{n_j - 1} \sum_i (y_{ij} - \bar{y}_{-j})^2 \quad SSE = \sum_i \sum_j (y_{ij} - \bar{y}_{-j})^2$$

1-way ANOVA

$$H_0 : \mu_{A_1} = \mu_{A_2} \dots = \mu_{A_n} = \mu$$

2) Если H_0 верна, то средние для значений фактора распределены следующим образом

$$\bar{y}_{-j} \sim N\left(\mu, \frac{\sigma^2}{n}\right)$$

Мы опять можем оценить наши дисперсию, но уже через эти средние!

$$\frac{\sigma^2}{n} \approx \frac{1}{a-1} \sum_{i=1}^a (\bar{y}_{-j} - \bar{y})^2$$

$$\sigma^2 \approx \frac{1}{a-1} \cdot n \cdot \sum_{i=1}^a (\bar{y}_{-j} - \bar{y})^2$$

1-way ANOVA

3) Если H_0 верна, то

$$\sigma^2 \approx \frac{1}{a-1} \cdot n \cdot \sum_{i=1}^a (\bar{y}_{-j} - \bar{y}) = MSA$$

а наблюдений
(наших средних),
считали общее
среднее - $df = a - 1$

$$\sigma^2 \approx s_{pooled}^2 = \frac{\sum_j (n_j - 1)s_j^2}{\sum_j (n_j - 1)} = \frac{SSE}{N - a} = MSE$$

п наблюдений (наших
средних), считали а
средних - потому $df = n$
- a

$$\frac{MSA}{MSE} \sim F(a - 1, n - a)$$

One-way ANOVA

```
res.aov <- aov(weight ~ group, data = PlantGrowth)
# Summary of the analysis
summary(res.aov)
```

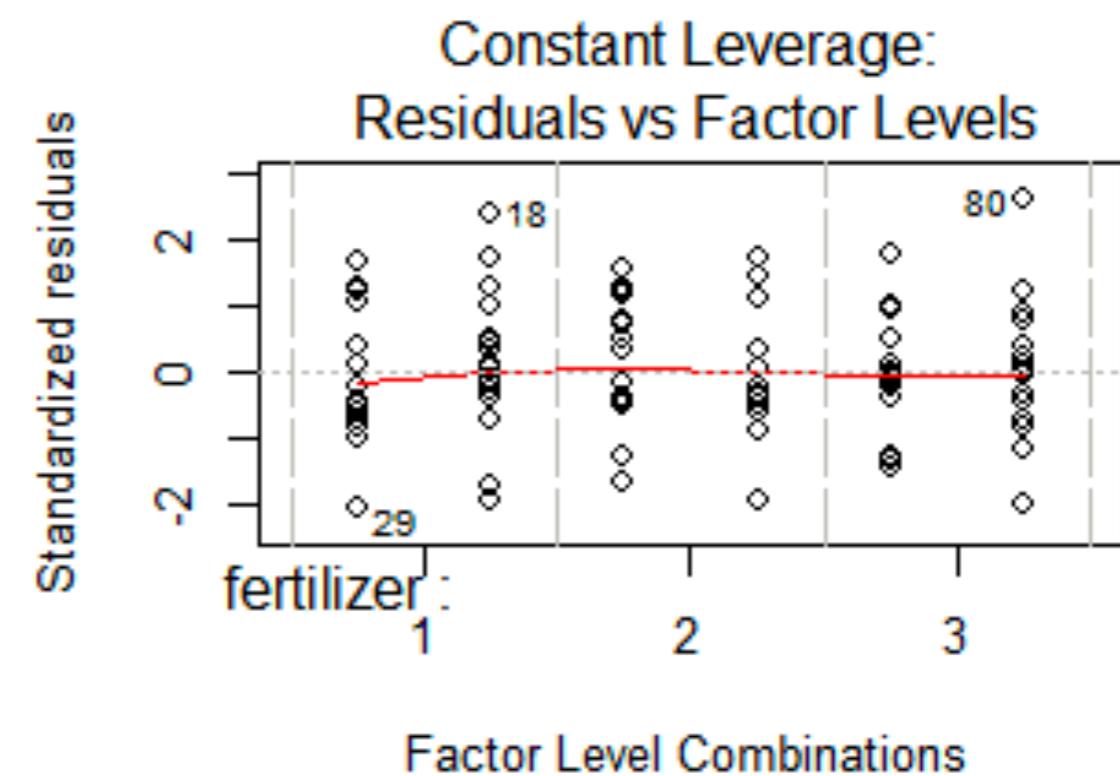
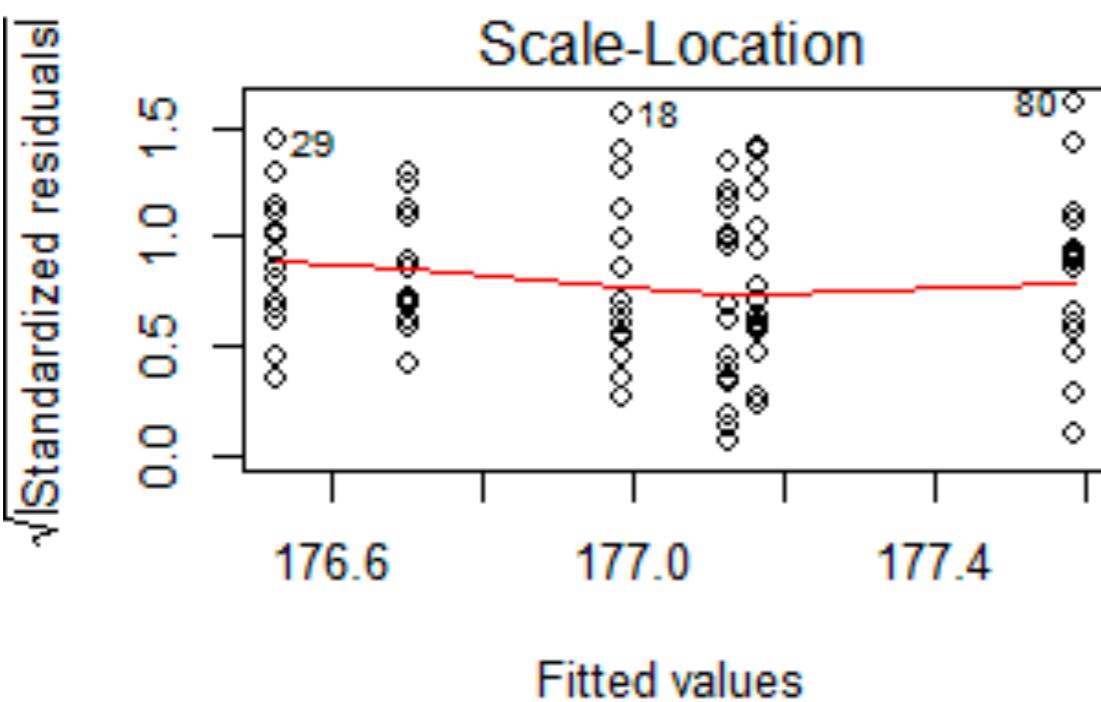
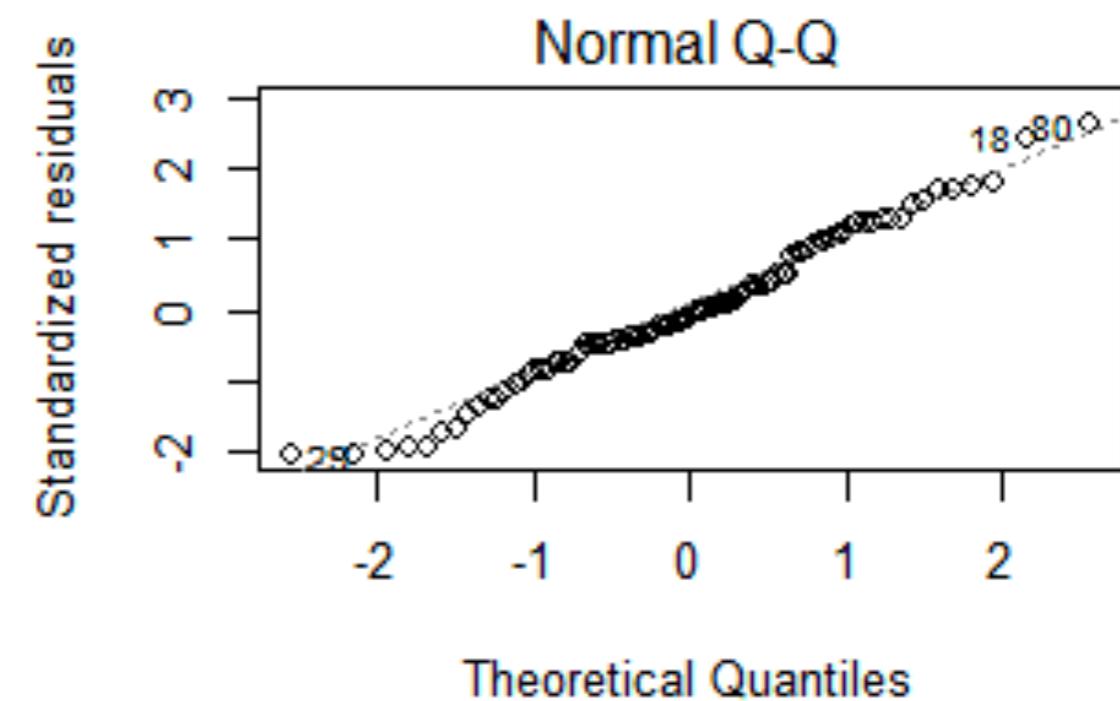
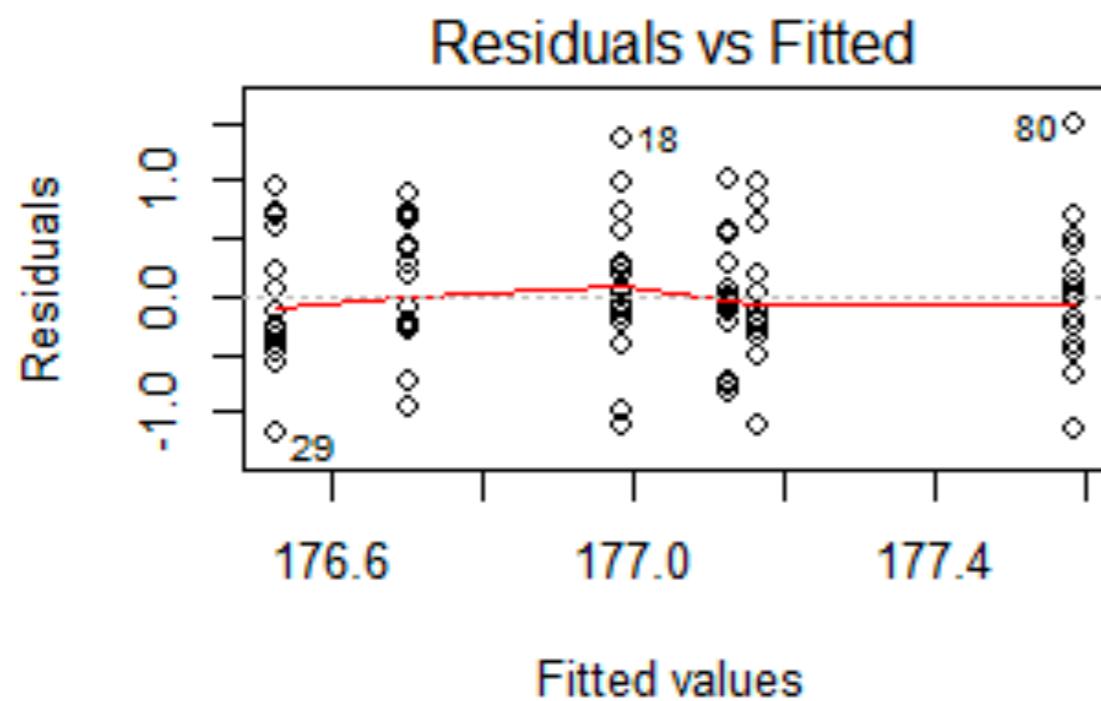
```
##                                Df Sum Sq Mean Sq F value Pr(>F)
## group                  2  3.766  1.8832   4.846 0.0159 *
## Residuals            27 10.492  0.3886
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

H₀ - нет разницы между группами ($u_1 = u_2 = \dots = u_n$)

Предположения one-way ANOVA

- Независимость наблюдений
- Нормальность остатков
- Гомоскедастичность (однородность дисперсий)

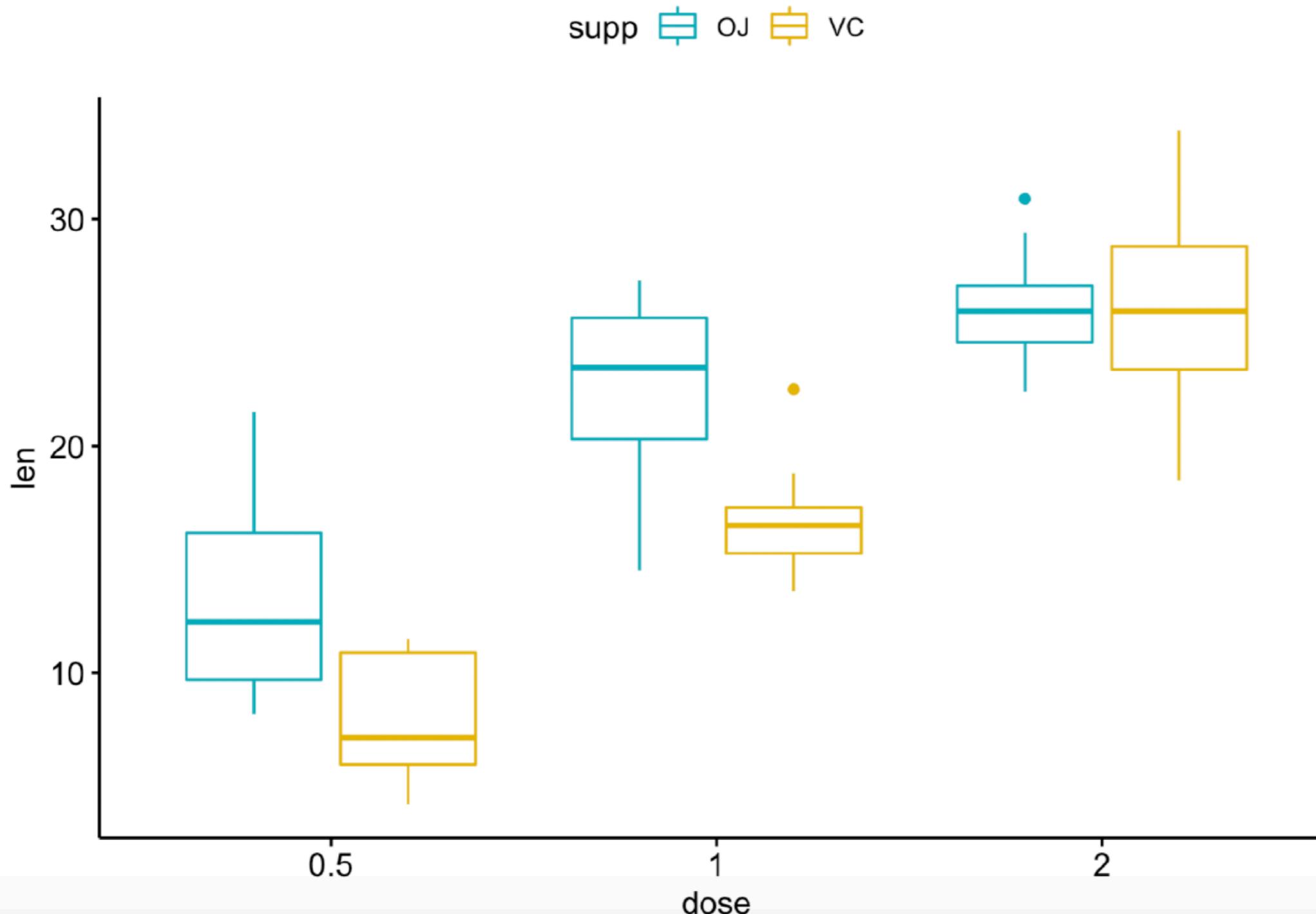
Проверять можно функцией `plot - plot(model)`



**Если факторов
несколько?**

Two-way ANOVA

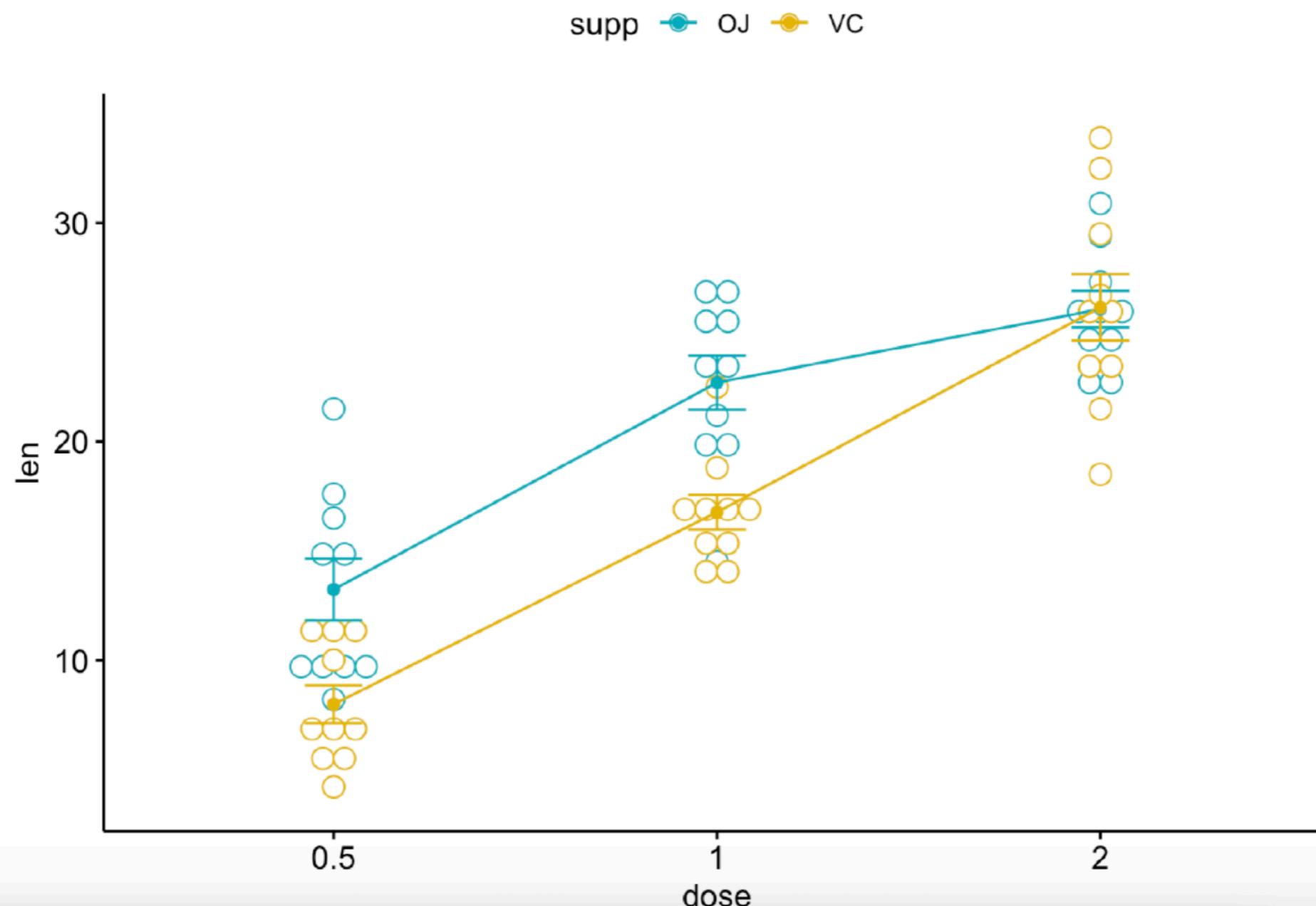
```
library("ggpubr")
ggboxplot(ToothGrowth, x = "dose", y = "len", color = "supp",
          palette = c("#00AFBB", "#E7B800"))
```



Two-way ANOVA

```
library("ggpubr")
ggline(ToothGrowth, x = "dose", y = "len", color = "supp",
       add = c("mean_se", "dotplot"),
       palette = c("#00AFBB", "#E7B800"))
```

`stat_bindot()` using `bins = 30`. Pick better value with `binwidth`.



Two-way ANOVA

```
ToothGrowth$dose <- factor(ToothGrowth$dose,  
                           levels = c(0.5, 1, 2),  
                           labels = c("D0.5", "D1", "D2"))  
res.aov2 <- aov(len ~ supp + dose, data = ToothGrowth)  
summary(res.aov2)
```

```
##                                Df Sum Sq Mean Sq F value    Pr(>F)  
## supp          1 205.4   205.4   14.02 0.000429 ***  
## dose          2 2426.4  1213.2   82.81 < 2e-16 ***  
## Residuals     56 820.4    14.7  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

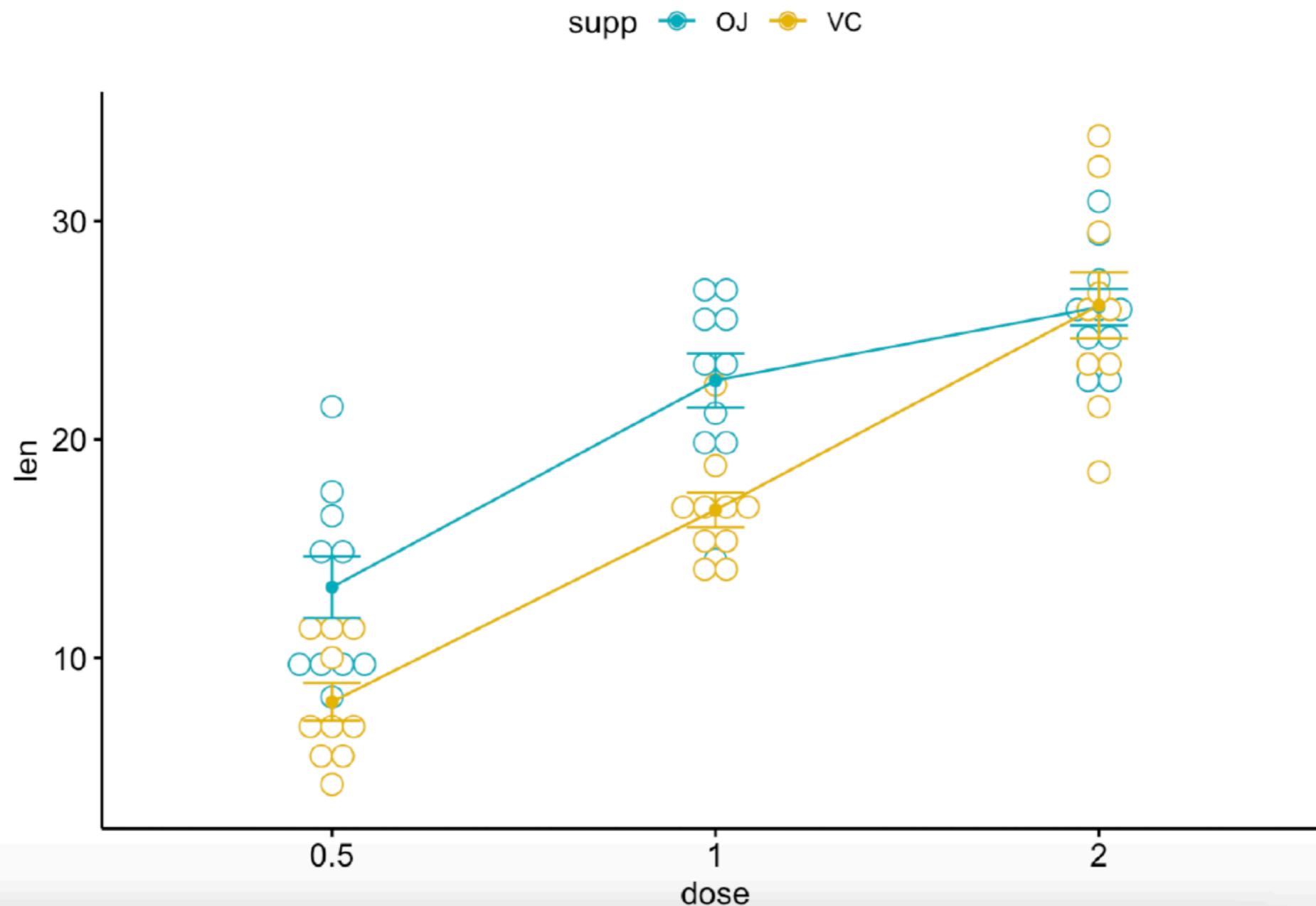
Первая H_0 - нет разницы между группами по supp ($u_1 = u_2 = \dots = u_n$)

Вторая H_0 - нет разницы между группами по dose ($u_1 = u_2 = \dots = u_n$)

Two-way ANOVA

```
library("ggpubr")
ggline(ToothGrowth, x = "dose", y = "len", color = "supp",
       add = c("mean_se", "dotplot"),
       palette = c("#00AFBB", "#E7B800"))
```

`stat_bindot()` using `bins = 30`. Pick better value with `binwidth`.



Two-way ANOVA

```
res.aov3 <- aov(len ~ supp * dose, data = ToothGrowth)
summary(res.aov3)
```

```
##                                Df Sum Sq Mean Sq F value    Pr(>F)
## supp                  1 205.4   205.4  15.572 0.000231 ***
## dose                  2 2426.4  1213.2  92.000 < 2e-16 ***
## supp:dose            2 108.3    54.2   4.107 0.021860 *
## Residuals             54 712.1    13.2
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Первая H0 - нет разницы между группами по supp ($u_1 = u_2 = \dots = u_n$)

Вторая H0 - нет разницы между группами по dose ($u_1 = u_2 = \dots = u_n$)

Третья H0 - нет влияния supp на dose

Post-hoc анализ

ANOVA сказала, что есть значимые различия

Что дальше?

Post-hoc анализ

ANOVA сказала, что есть значимые различия

Что дальше?

Искать, какие конкретно группы отличаются

Post-hoc анализ

ANOVA сказала, что есть значимые различия

Что дальше?

Искать, какие конкретно группы отличаются

Делаем попарные t-тесты.

Можно делать чуть более умные тесты, проводящие все сравнения вместе, например, Tukey HSD test.

Неучтенные эффекты

Предположим, что наша целевая переменная в действительности не объясняется нашими переменными

$$y_i = \mu + \beta x_i + \gamma z_i + \alpha_i + \epsilon_i$$

The diagram illustrates the components of a regression model. On the left, four labels are aligned vertically: 'Свободный коэффициент (intercept)', 'Объясняющая переменная 1', 'Объясняющая переменная 2', and 'Нормальный шум'. Arrows point from each label to its corresponding term in the equation above: the first arrow points to μ , the second to βx_i , the third to γz_i , and the fourth to ϵ_i .

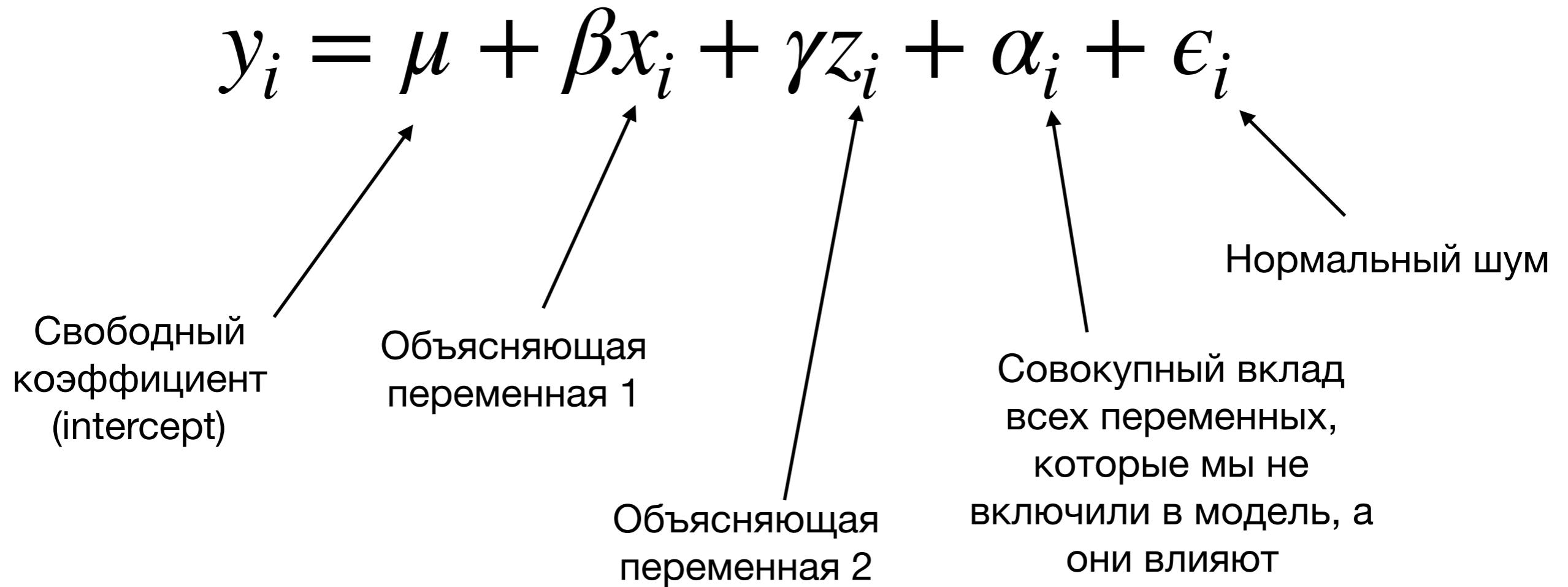
Свободный коэффициент (intercept)

Объясняющая переменная 1

Объясняющая переменная 2

Нормальный шум

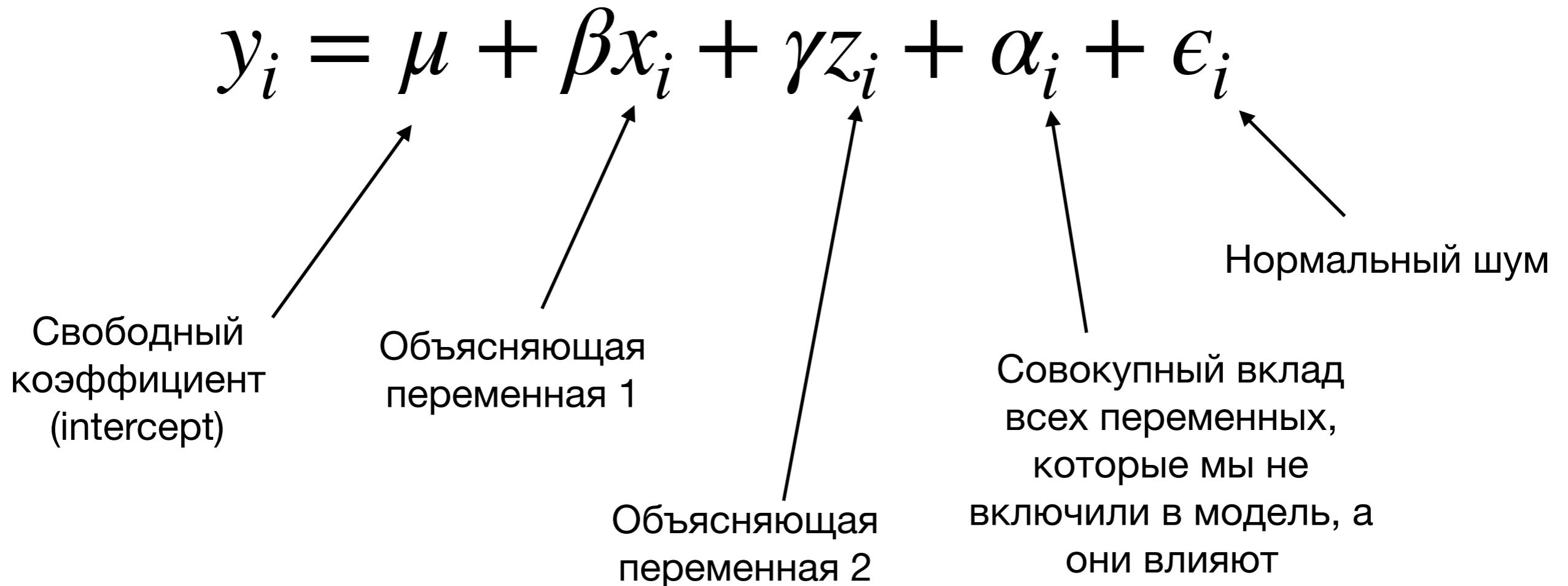
Неучтенные эффекты



Что будет, если мы попробуем аппроксимировать это моделью?

$$y_i = \mu + \beta x_i + \gamma z_i + \epsilon_i$$

Неучтенные эффекты



Что будет, если мы попробуем аппроксимировать это моделью?

$$y_i = \mu + \beta x_i + \gamma z_i + \epsilon_i$$

Модель будет пытаться распихать alpha по вкладам включенных нами переменных

Output 2.1 Regressions of ANTI on POV and SELF in 1990 and 1994

Dependent Variable: anti90 child antisocial behavior in 1990

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > t
Intercept	1	2.37482	0.38447	6.18	<.0001
self90	1	-0.05014	0.01870	-2.68	0.0075
pov90	1	0.59473	0.12629	4.71	<.0001

Dependent Variable: anti94 child antisocial behavior in 1994

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > t
Intercept	1	2.88797	0.44688	6.46	<.0001
self94	1	-0.06388	0.02113	-3.02	0.0026
pov94	1	0.54712	0.14765	3.71	0.0002

Что делать?

**У нас есть данные за два года. Предположим, что невключенные
эффекты остались прежними**

$$y_{i1} = \mu_1 + \beta x_{i1} + \gamma z_{i1} + \alpha_i + \epsilon_{i1}$$

$$y_{i2} = \mu_2 + \beta x_{i2} + \gamma z_{i2} + \alpha_i + \epsilon_{i2}$$

Что делать?

У нас есть данные за два года. Предположим, что невключенные
эффекты остались прежними

$$y_{i1} = \mu_1 + \beta x_{i1} + \gamma z_{i1} + \alpha_i + \epsilon_{i1}$$



$$y_{i2} = \mu_2 + \beta x_{i2} + \gamma z_{i2} + \alpha_i + \epsilon_{i2}$$



$$\Delta y_i = \Delta \mu + \beta \Delta x_i + \gamma \Delta z_i + \Delta \epsilon_i$$

И будем учить эту модель

Работает!

Output 2.2 Regression with Difference Scores

Dependent Variable: *antidiff*

<i>Variable</i>	<i>DF</i>	<i>Parameter Estimate</i>	<i>Standard Error</i>	<i>t Value</i>	<i>Pr > t </i>
<i>Intercept</i>	1	0.20923	0.06305	3.32	0.0010
<i>selfdiff</i>	1	-0.05615	0.01531	-3.67	0.0003
<i>povdiff</i>	1	-0.03631	0.12827	-0.28	0.7772

Fixed effects model

$$y_{it} = \mu_t + \beta x_{it} + \gamma z_{it} + \alpha_i + \epsilon_{it}$$

- Если обобщать для случая, когда у нас есть несколько периодов наблюдения/зон наблюдения и тд, то придет к модели, которая далее (почти) сводится к fixed effects model.
- Идея простая - просто добавляем в модель категориальную переменную, отвечающую за период наблюдения
- По сути - все это мы уже делали, когда просто строили регрессию с категориальными переменными

Почему называется fixed effects model?

Fixed effects model

$$y_{it} = \mu_t + \beta x_{it} + \gamma z_{it} + \alpha_i + \epsilon_{it}$$

- Если обобщать для случая, когда у нас есть несколько периодов наблюдения/зон наблюдения и тд, то придет к модели, которая далее (почти) сводится к fixed effects model.
- Идея простая - просто добавляем в модель категориальную переменную, отвечающую за период наблюдения
- По сути - все это мы уже делали, когда просто строили регрессию с категориальными переменными

Fixed effect - alpha, не зависит от t

```

data(Fatalities)
Fatalities$fatal_rate <- Fatalities$fatal / Fatalities$pop * 10000
fatal_fe_lm_mod <- lm(fatal_rate ~ beertax + state, data = Fatalities)
summary(fatal_fe_lm_mod)

```

```

##
## Call:
## lm(formula = fatal_rate ~ beertax + state, data = Fatalities)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.58696 -0.08284 -0.00127  0.07955  0.89780
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.47763   0.31336 11.098 < 2e-16 ***
## beertax     -0.65587  0.18785 -3.491 0.000556 ***
## stateaz     -0.56773  0.26667 -2.129 0.034107 *
## statear     -0.65495  0.21902 -2.990 0.003028 **
## stateca     -1.50947  0.30435 -4.960 1.21e-06 ***
## stateco     -1.48428  0.28735 -5.165 4.50e-07 ***
## statect     -1.86226  0.28053 -6.638 1.58e-10 ***
## statede     -1.30760  0.29395 -4.448 1.24e-05 ***
## statefl     -0.26813  0.13933 -1.924 0.055284 .
## ...
##   - - - - -

```

**Стандартные ошибки тут не совсем верные,
оценивать можно, к примеру, через бутстрэп**

Random effects model

$$y_{it} = \mu_t + \beta x_{it} + \gamma z_{it} + \alpha_i + \epsilon_{it}$$

$$\alpha_i \sim N(0, \sigma^2)$$

Дополнительно: Эффект alpha не зависит от признаков и шума

Датасет с драконами

- Драконы живут на разных горных массивах, в трех случайных локациях этих массивов
- Драконы имеют разную длину
- Драконы сдавали тест и получили за него баллы
- Хотим изучить зависимость результатов за тест от длины дракона

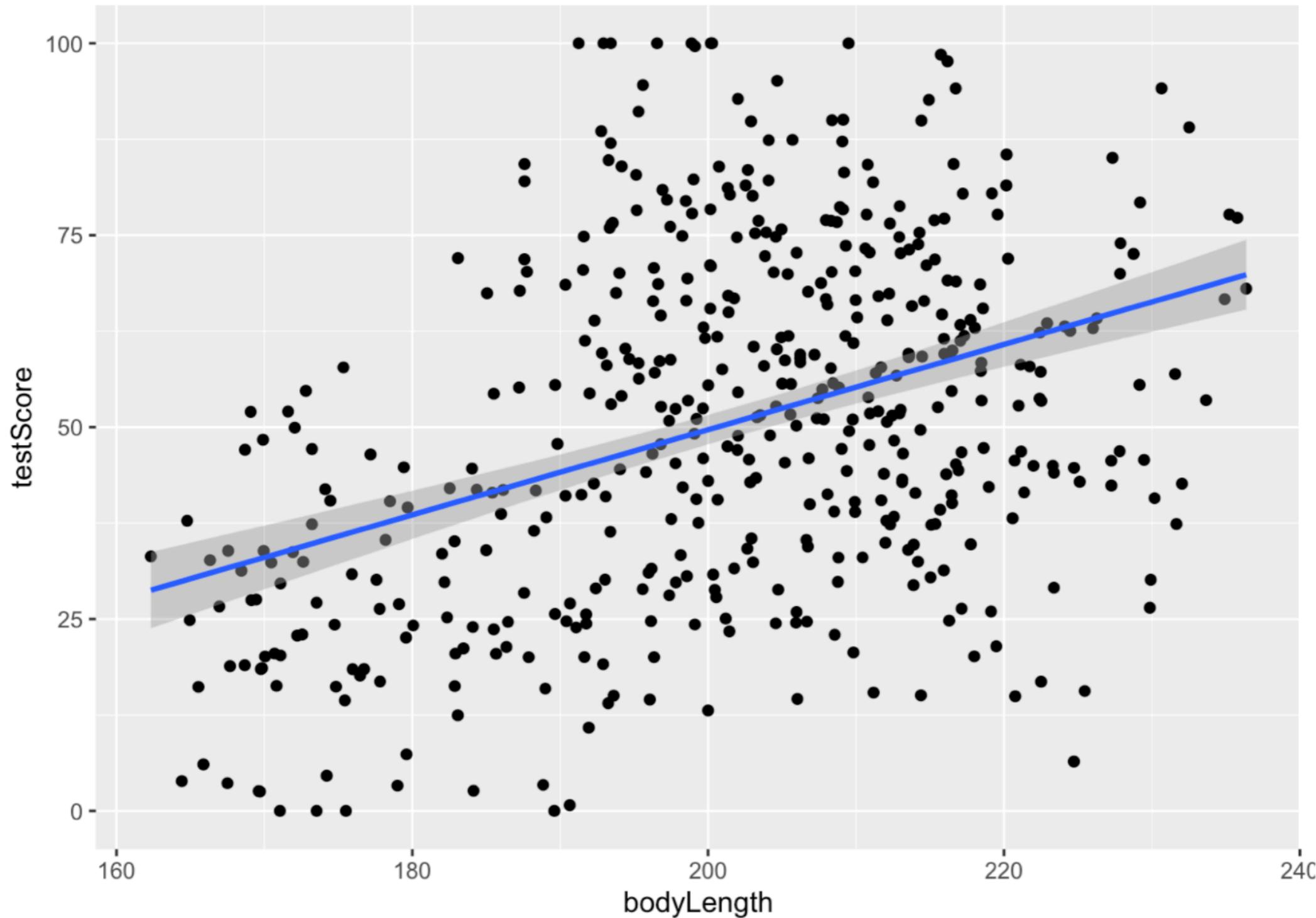
```
basic.lm <- lm(testScore ~ bodyLength, data = dragons)
summary(basic.lm)
```

```
##
## Call:
## lm(formula = testScore ~ bodyLength, 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) -61.31783   12.06694  -5.081 5.38e-07 ***
## bodyLength    0.55487    0.05975   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
```

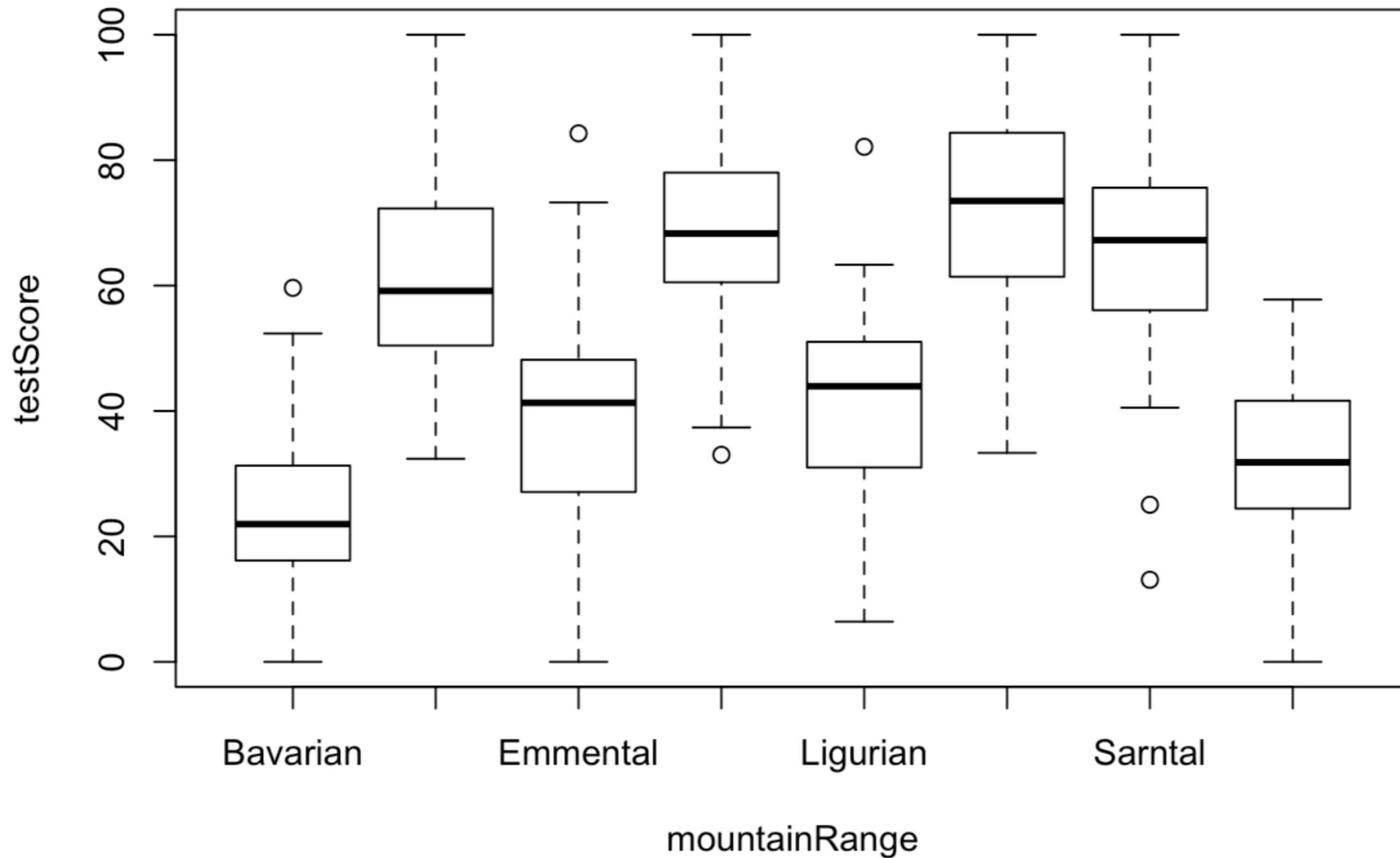
```
library(ggplot2) # load the package

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

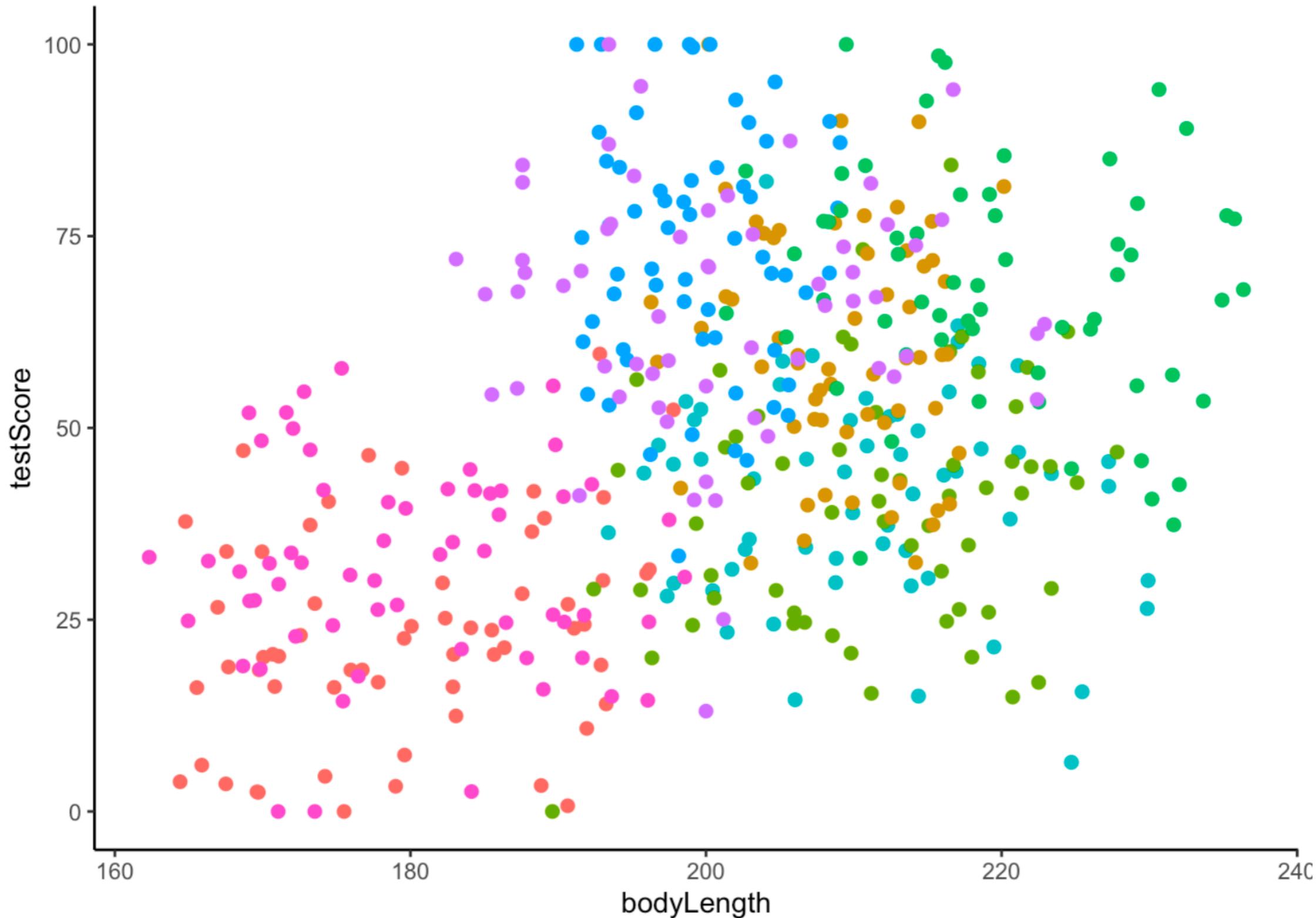
```
## `geom_smooth()` using formula 'y ~ x'
```



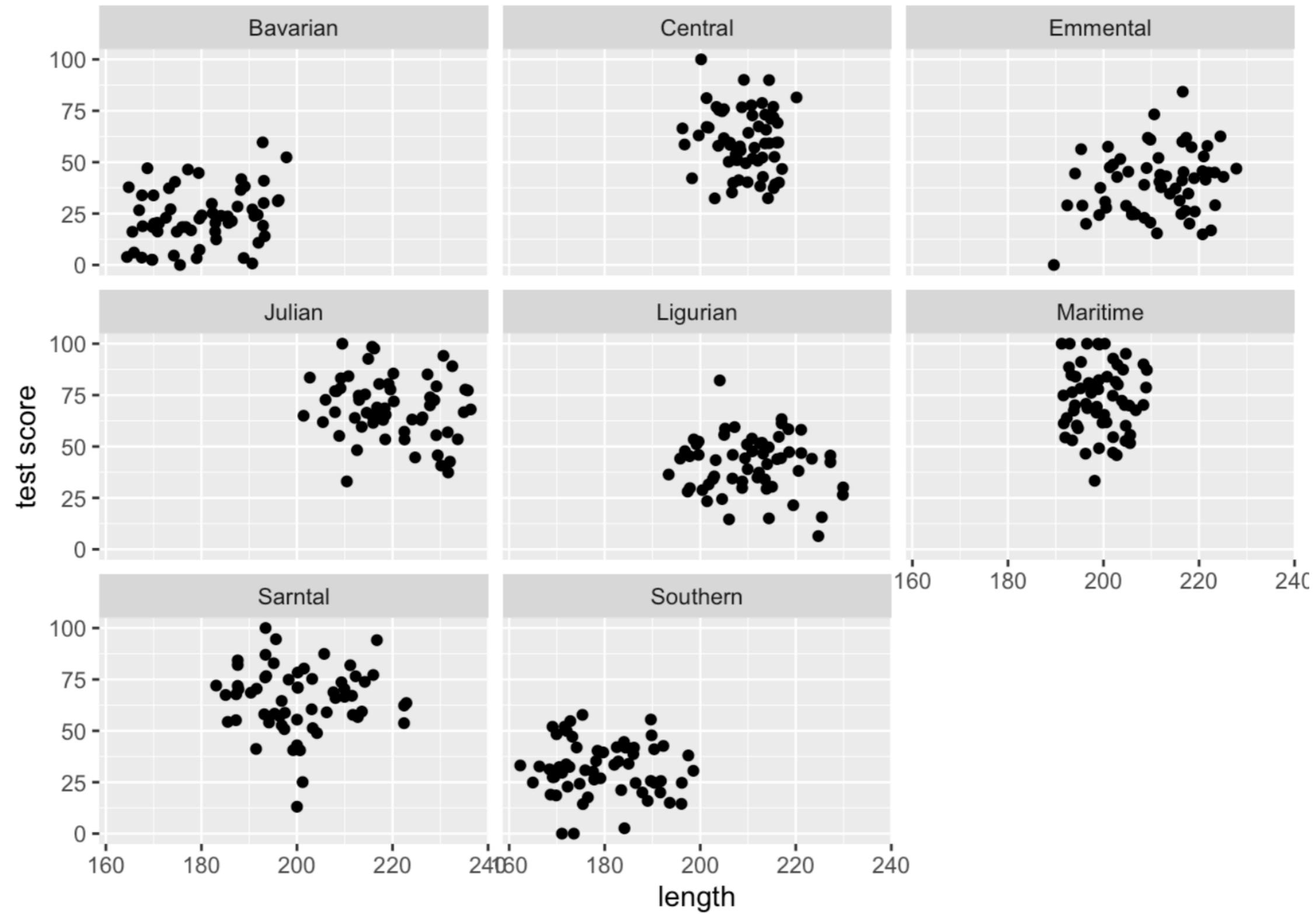
```
boxplot(testScore ~ mountainRange, data = dragons)
```



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



```
(split_plot <- ggplot(aes(bodyLength, testScore), data = dragons) +  
  geom_point() +  
  facet_wrap(~ mountainRange) + # create a facet for each mountain range  
  xlab("length") +  
  ylab("test score"))
```



```
mountain.lm <- lm(testScore ~ bodyLength + mountainRange, data = dragons)
summary(mountain.lm)
```

```
##  
## Call:  
## lm(formula = testScore ~ bodyLength + mountainRange, data = dragons)  
##  
## Residuals:  
##      Min       1Q   Median       3Q      Max  
## -52.263  -9.926   0.361   9.994  44.488  
##  
## Coefficients:  
##                               Estimate Std. Error t value Pr(>|t|)  
## (Intercept)           20.83051  14.47218  1.439  0.15072  
## bodyLength            0.01267   0.07974  0.159  0.87379  
## mountainRangeCentral 36.58277   3.59929 10.164 < 2e-16 ***  
## mountainRangeEmmental 16.20923   3.69665  4.385 1.43e-05 ***  
## mountainRangeJulian   45.11469   4.19012 10.767 < 2e-16 ***  
## mountainRangeLigurian 17.74779   3.67363  4.831 1.84e-06 ***  
## mountainRangeMaritime 49.88133   3.13924 15.890 < 2e-16 ***  
## mountainRangeSarntal  41.97841   3.19717 13.130 < 2e-16 ***  
## mountainRangeSouthern  8.51961   2.73128  3.119  0.00192 **  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## Residual standard error: 14.96 on 471 degrees of freedom  
## Multiple R-squared:  0.5843, Adjusted R-squared:  0.5773  
## F-statistic: 82.76 on 8 and 471 DF,  p-value: < 2.2e-16
```

```
mountain.lm <- lm(testScore ~ bodyLength + mountainRange, data = dragons)
summary(mountain.lm)
```

```
##  
## Call:  
## lm(formula = testScore ~ bodyLength + mountainRange, data = dragons)  
##  
## Residuals:  
##      Min       1Q   Median       3Q      Max  
## -52.263  -9.926   0.361   9.994  44.488  
##  
## Coefficients:  
##  
##                               Estimate Std. Error t value Pr(>|t|)  
## (Intercept)           20.83051  14.47218  1.439  0.15072  
## bodyLength            0.01267   0.07974  0.159  0.87379  
## mountainRangeCentral 36.58277   3.59929 10.164 < 2e-16 ***  
## mountainRangeEmmental 16.20923   3.69665  4.385 1.43e-05 ***  
## mountainRangeJulian   45.11469   4.19012 10.767 < 2e-16 ***  
## mountainRangeLigurian 17.74779   3.67363  4.831 1.84e-06 ***  
## mountainRangeMaritime 49.88133   3.13924 15.890 < 2e-16 ***  
## mountainRangeSarntal  41.97841   3.19717 13.130 < 2e-16 ***  
## mountainRangeSouthern  8.51961   2.73128  3.119  0.00192 **  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## Residual standard error: 14.96 on 471 degrees of freedom  
## Multiple R-squared:  0.5843, Adjusted R-squared:  0.5773  
## F-statistic: 82.76 on 8 and 471 DF,  p-value: < 2.2e-16
```

**Можно ли считать
mountainRange fixed
effect?**

```
mountain.lm <- lm(testScore ~ bodyLength + mountainRange, data = dragons)
summary(mountain.lm)
```

```
##  
## Call:  
## lm(formula = testScore ~ bodyLength + mountainRange, data = dragons)  
##  
## Residuals:  
##      Min       1Q   Median       3Q      Max  
## -52.263  -9.926   0.361   9.994  44.488  
##  
## Coefficients:  
##  
##                               Estimate Std. Error t value Pr(>|t|)  
## (Intercept)           20.83051  14.47218  1.439  0.15072  
## bodyLength            0.01267   0.07974  0.159  0.87379  
## mountainRangeCentral 36.58277   3.59929 10.164 < 2e-16 ***  
## mountainRangeEmmental 16.20923   3.69665  4.385 1.43e-05 ***  
## mountainRangeJulian   45.11469   4.19012 10.767 < 2e-16 ***  
## mountainRangeLigurian 17.74779   3.67363  4.831 1.84e-06 ***  
## mountainRangeMaritime 49.88133   3.13924 15.890 < 2e-16 ***  
## mountainRangeSarntal  41.97841   3.19717 13.130 < 2e-16 ***  
## mountainRangeSouthern  8.51961   2.73128  3.119  0.00192 **  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## Residual standard error: 14.96 on 471 degrees of freedom  
## Multiple R-squared:  0.5843, Adjusted R-squared:  0.5773  
## F-statistic: 82.76 on 8 and 471 DF,  p-value: < 2.2e-16
```

**Можно ли считать
mountainRange fixed
effect?**

Можно

```
mountain.lm <- lm(testScore ~ bodyLength + mountainRange, data = dragons)
summary(mountain.lm)
```

```
##  
## Call:  
## lm(formula = testScore ~ bodyLength + mountainRange, data = dragons)  
##  
## Residuals:  
##      Min       1Q   Median       3Q      Max  
## -52.263  -9.926   0.361   9.994  44.488  
##  
## Coefficients:  
##  
##                               Estimate Std. Error t value Pr(>|t|)  
## (Intercept)           20.83051  14.47218  1.439  0.15072  
## bodyLength            0.01267   0.07974  0.159  0.87379  
## mountainRangeCentral 36.58277   3.59929 10.164 < 2e-16 ***  
## mountainRangeEmmental 16.20923   3.69665  4.385 1.43e-05 ***  
## mountainRangeJulian   45.11469   4.19012 10.767 < 2e-16 ***  
## mountainRangeLigurian 17.74779   3.67363  4.831 1.84e-06 ***  
## mountainRangeMaritime 49.88133   3.13924 15.890 < 2e-16 ***  
## mountainRangeSarntal  41.97841   3.19717 13.130 < 2e-16 ***  
## mountainRangeSouthern  8.51961   2.73128  3.119  0.00192 **  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## Residual standard error: 14.96 on 471 degrees of freedom  
## Multiple R-squared:  0.5843, Adjusted R-squared:  0.5773  
## F-statistic: 82.76 on 8 and 471 DF,  p-value: < 2.2e-16
```

**Можно ли считать
mountainRange fixed
effect?**

Можно

**Но на самом деле
нам этот эффект
абсолютно в
исследовании не
нужен.**

**И, мы с тем же
успехом могли
взять и другие
горные массивы**

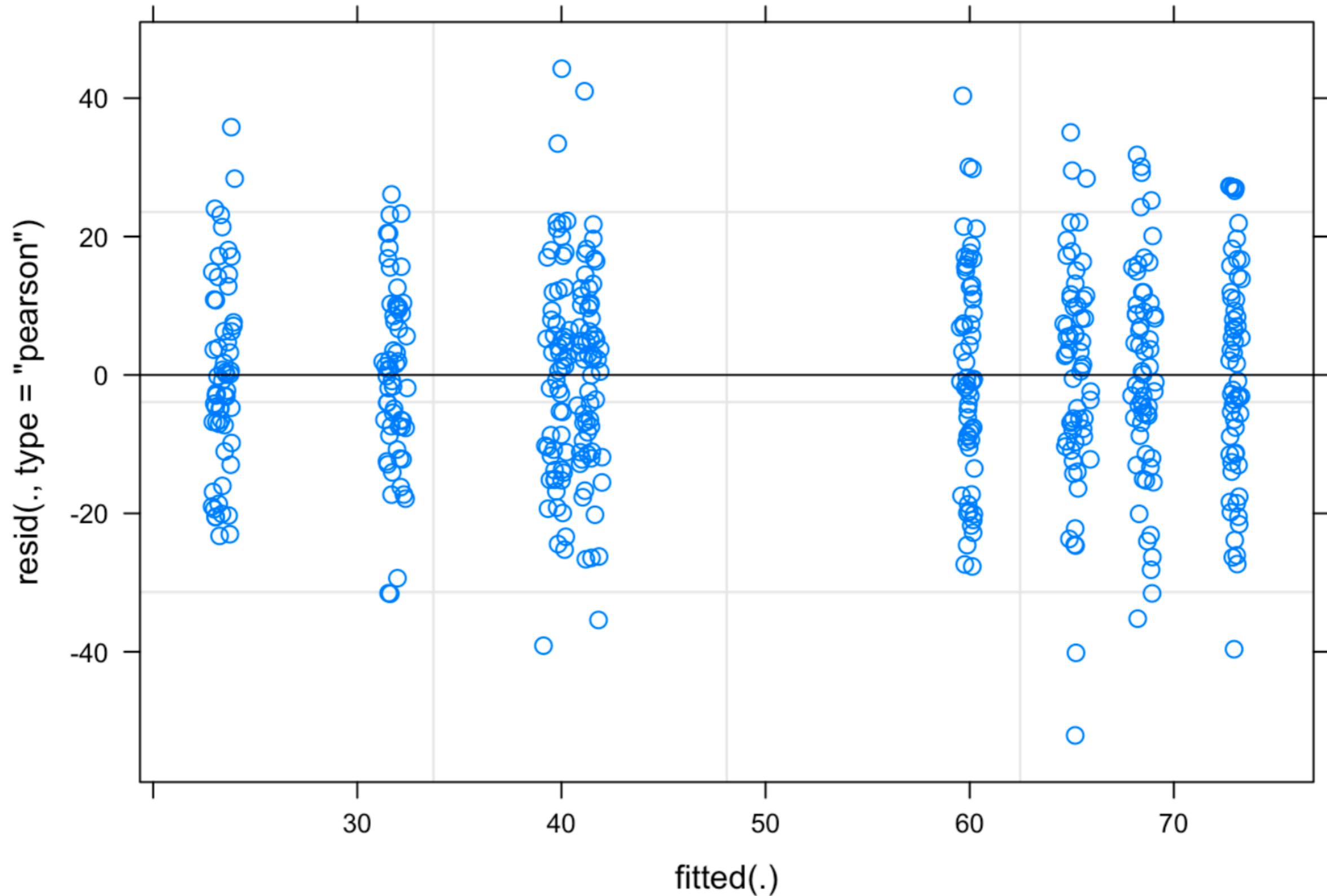
Fixed effect vs random effect

1. Если уровней фактора мало (меньше 5), то лучше сделать его **fixed effect**, чтобы модель давала оценки коэффициентов, на которые можно положиться
2. Если фактор является важной частью исследования и мы хотим далее делать о нем выводы, то это **fixed effect**
3. Если фактор просто группирует как-то наши данные (номер повторности, номер плашки и тд), но он нам не интересен, но он может оказывать эффект. В этом случае это **random effect**.
4. Еще один способ дать определение **random effect** - уровни фактора не имеют принципиального значения. Они просто выбраны из некой генеральной совокупности. Например - мы взяли для исследования больных из 10 больниц.

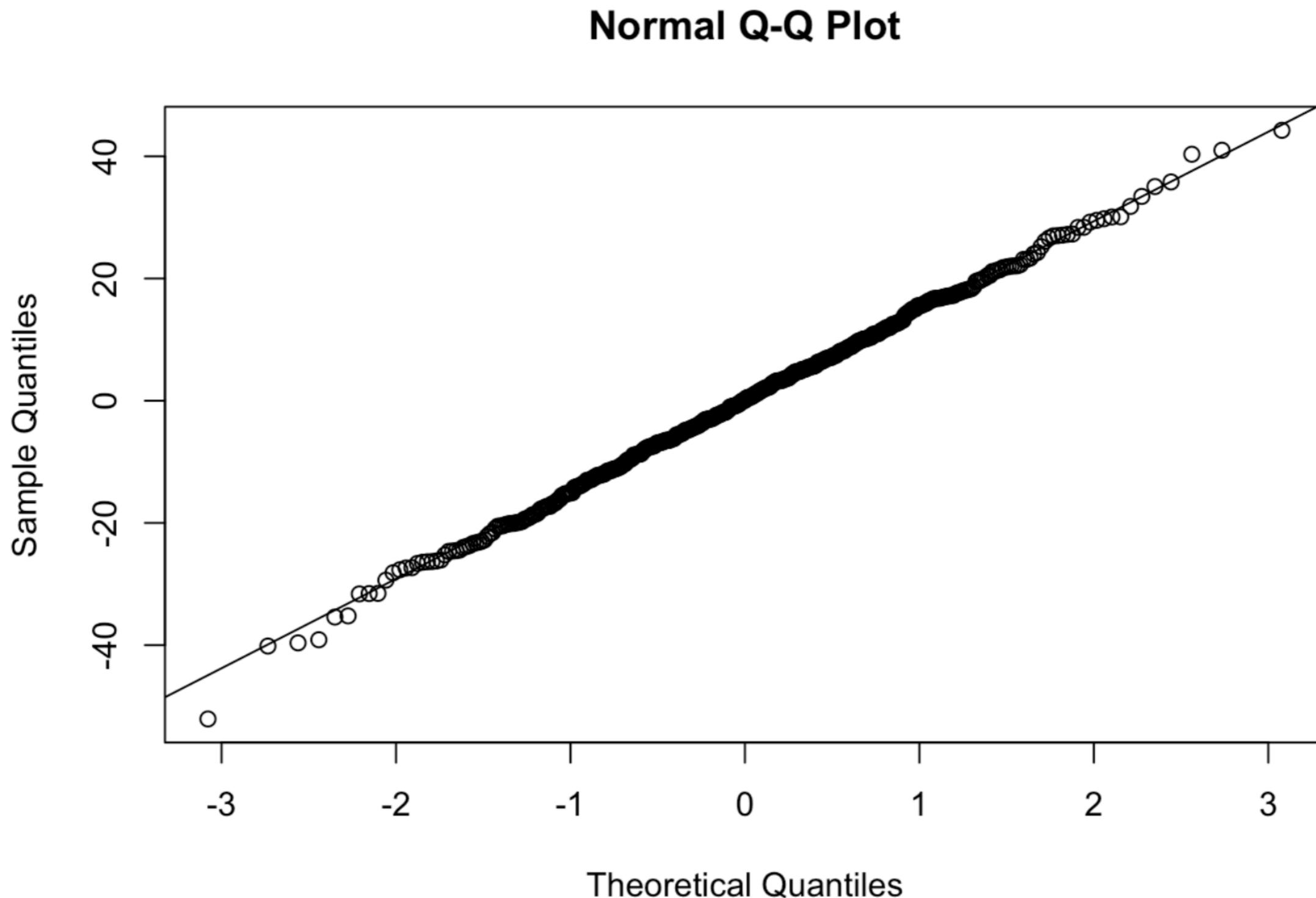
```
mixed.lmer <- lmer(testScore ~ bodyLength + (1 | mountainRange), data = dragons)
summary(mixed.lmer)
```

```
## Linear mixed model fit by REML [ 'lmerMod' ]
## Formula: testScore ~ bodyLength + (1 | mountainRange)
##   Data: dragons
##
## REML criterion at convergence: 3991.2
##
## 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) 43.70938  17.13489  2.551
## bodyLength   0.03316   0.07865  0.422
##
## Correlation of Fixed Effects:
##          (Intr)
## bodyLength -0.924
```

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



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



**У нас остались еще
регионы каждой горы**

Crossed random effects

- Представим, что наши драконы умеют летать между регионами одной горы
- Тогда мы могли протестировать одного и того же дракона несколько раз
- То есть либо у нас есть для некоторых драконов наблюдения есть для разных регионов (**partially crossed**)
- А, может, вообще каждый дракон побывал в каждом из регионов (**fully crossed**)

Crossed random effects



Наблюдаем рост лесопосадок в течении трех лет. Понимаем, что наблюдения из одного года могут быть более похожи друг на друга, понимаем, что и наблюдения из одного сезона одного года могут быть похожи друг на друга

Nested random effects

- Пусть у нас 50 семян в каждой лунке, 10 контрольных и 10 экспериментальных. Это 1000 семян всего. Пусть мы собираем данные о них каждый сезон в течении трех лет. И на каждом растении мы измеряем 5 листков. Это 60000 измерений. Но эти измерения **зависимы!**
- Потому регрессия вида $\text{leafLength} \sim \text{treatment}$ будет преувеличивать значимость результатов
- Такая регрессия учит все эти зависимости $\text{leafLength} \sim \text{treatment} + (1 | \text{Bed/Plant/Leaf})$
- Остается учесть то, что сезоны (Зима1, ... Осень3) могут оказывать влияние на листья.
- $\text{leafLength} \sim \text{treatment} + (1 | \text{Bed/Plant/Leaf}) + (1 | \text{Season})$

Возвращаясь к драконам

Хотим учесть локацию

```
mixed.WRONG <- lmer(testScore ~ bodyLength + (1|mountainRange) + (1|site), data = dragons) # treats the two random effects as if they are crossed
summary(mixed.WRONG)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: testScore ~ bodyLength + (1 | mountainRange) + (1 | site)
##   Data: dragons
##
## REML criterion at convergence: 3986.5
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.5079 -0.6489  0.0138  0.6976  3.0851
##
## Random effects:
##   Groups      Name        Variance Std.Dev.
##   mountainRange (Intercept) 409.90    20.246
##   site          (Intercept) 10.52     3.243
##   Residual                  219.19    14.805
## Number of obs: 480, groups: mountainRange, 8; site, 3
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept) 82.6667   21.6916   3.811
## bodyLength  -0.1603    0.1012  -1.584
##
## Correlation of Fixed Effects:
##           (Intr)
## bodyLength -0.940
```

Что не так?

Возвращаясь к драконам

Хотим учесть локацию

```
mixed.WRONC <- lmer(testScore ~ bodyLength + (1|mountainRange) + (1|site), data = dragons) # treats the two random effects as if they are crossed
summary(mixed.WRONC)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: testScore ~ bodyLength + (1 | mountainRange) + (1 | site)
##   Data: dragons
##
## REML criterion at convergence: 3986.5
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.5079 -0.6489  0.0138  0.6976  3.0851
##
## Random effects:
##   Groups      Name        Variance Std.Dev.
##   mountainRange (Intercept) 409.90    20.246
##   site          (Intercept) 10.52     3.243
##   Residual                    219.19    14.805
## Number of obs: 480, groups: mountainRange, 8; site, 3
##
```

```
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept) 82.6667   21.6916  3.811
## bodyLength -0.1603    0.1012 -1.584
##
```

```
## Correlation of Fixed Effects:
##           (Intr)
## bodyLength -0.940
```

Эффекты не crossed

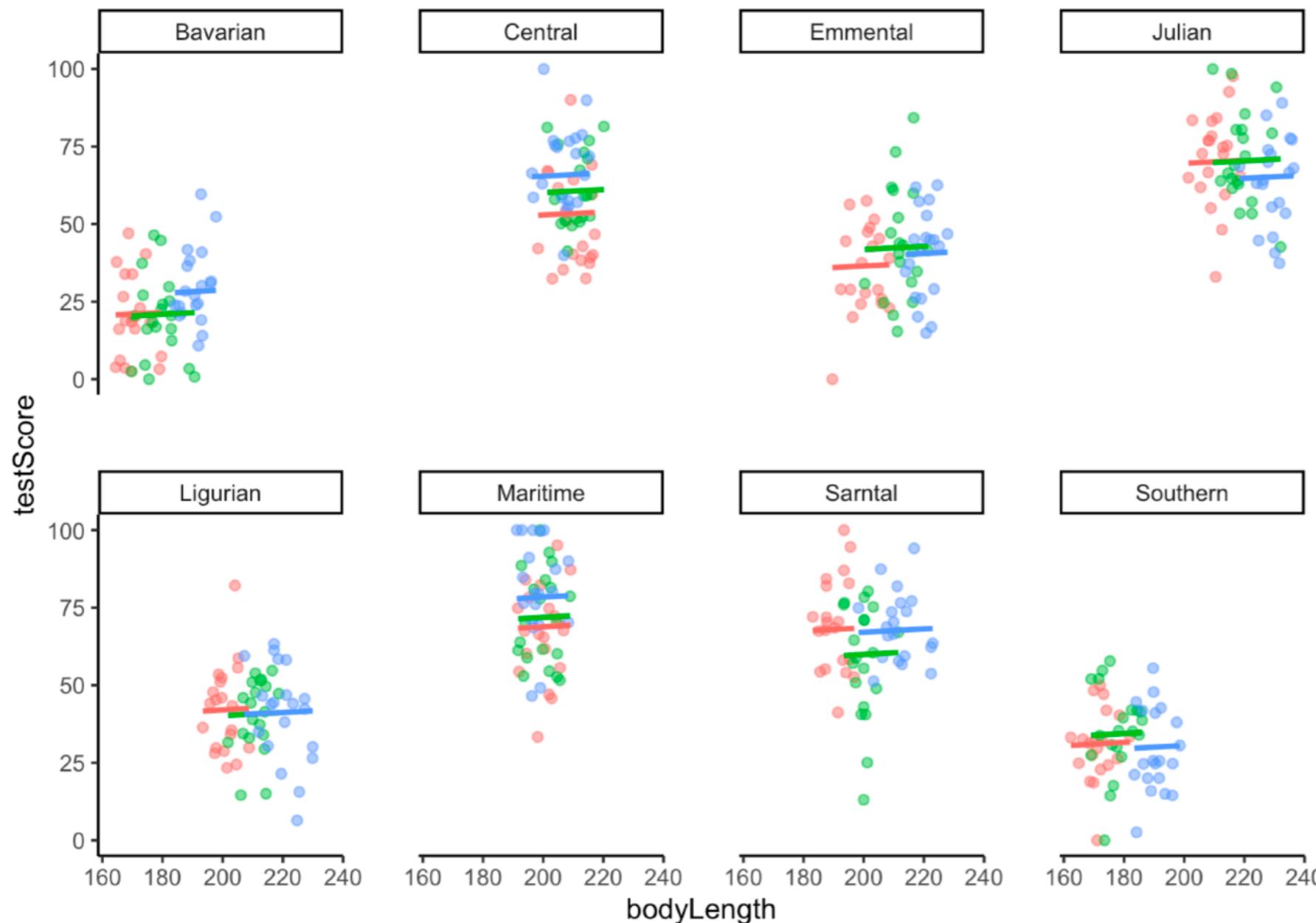
Возвращаясь к драконам

```
mixed.lmer2 <- lmer(testScore ~ bodyLength + (1|mountainRange/site) , data = dragons) # the syntax stays the same, but now the nesting is taken into account
summary(mixed.lmer2)
```

```
## Linear mixed model fit by REML [ 'lmerMod' ]
## Formula: testScore ~ bodyLength + (1 | mountainRange/site)
##   Data: dragons
##
## REML criterion at convergence: 3976
##
## 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) 40.06668  21.86373  1.833
## bodyLength   0.05126   0.10368  0.494
##
## Correlation of Fixed Effects:
##            (Intr)
## bodyLength -0.955
```

Все наклоны кривых одинаковы

```
(mm_plot <- ggplot(dragons, aes(x = bodyLength, y = testScore, colour = site)) +  
  facet_wrap(~mountainRange, nrow=2) +  # a panel for each mountain range  
  geom_point(alpha = 0.5) +  
  theme_classic() +  
  geom_line(data = cbind(dragons, pred = predict(mixed.lmer2)), aes(y = pred), size = 1) +  # adding predicted line from mixed model  
  theme(legend.position = "none",  
        panel.spacing = unit(2, "lines"))  # adding space between panels  
)
```



Напоминание

- В fixed model мы учитывали эффект категории на наклон следующим образом

$$Y = \beta_0 + \beta_1 \cdot X + \beta_2 \cdot Z + \beta_3 \cdot X \cdot Z + \epsilon$$

Вклад X **Вклад Z** **moderation effect Z**

Напоминание

В fixed model мы учитывали эффект категории на наклон следующим образом

```
model <- lm(Price_RUR ~ HDD_Gb * HDD_type, data=laptop)
summary(model)

##
## Call:
## lm(formula = Price_RUR ~ HDD_Gb * HDD_type, data = laptop)
##
## Residuals:
##    Min      1Q  Median      3Q     Max 
## -21886   -6049   -1461    2885   89344 
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 12430.529   1525.776   8.147 9.97e-15 ***
## HDD_Gb        17.270      2.488   6.941 2.38e-11 ***
## HDD_typeSSD  18232.081   4265.934   4.274 2.58e-05 ***
## HDD_Gb:HDD_typeSSD  80.870      12.874   6.281 1.17e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10480 on 302 degrees of freedom
## Multiple R-squared:  0.5457, Adjusted R-squared:  0.5412 
## F-statistic: 120.9 on 3 and 302 DF,  p-value: < 2.2e-16
```

Учитываем наклон кривых

В случае random effects, по сути делаем почти то же самое

$$y_{it} = \mu_t + \beta x_{it} + \alpha_i + \gamma_i x_{it} + \epsilon_{it}$$

$$\alpha_i \sim N(0, \sigma_\alpha^2)$$

$$\gamma_i \sim N(0, \sigma_\gamma^2)$$

Учитываем наклон кривых

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

```
## boundary (singular) fit: see ?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
##
```

Учитываем наклон кривых

```
(mm_plot <- ggplot(dragons, aes(x = bodyLength, y = testScore, colour = site)) +  
  facet_wrap(~mountainRange, nrow=2) +  # a panel for each mountain range  
  geom_point(alpha = 0.5) +  
  theme_classic() +  
  geom_line(data = cbind(dragons, pred = predict(mixed.ranslope)), aes(y = pred), size = 1) +  # adding predicted line from mixed model  
  theme(legend.position = "none",  
        panel.spacing = unit(2, "lines")) # adding space between panels  
)
```

