Linguistic data: Quantitative Analysis and Visualisation

Mixed-effects linear models

Ilya Schurov, Olga Lyashevskaya, George Moroz, Alla Tambovtseva

Linear mixed-effects models

Based on this Lab11.

At this seminar we will work with the data set `ReductionRussian.txt`. Pavel Duryagin ran an experiment on perception of vowel reduction in Russian language. The data set includes the following variables:

- `time1`: reaction time 1;
- `time2`: reaction time 2;
- `duration`: duration of the vowel in the stimuly (in milliseconds, ms);
- `f1`: the first formant (if don't know about formants, see here);
- `f2`: the second formant;
- `f3`: the third formant;
- `vowel`: vowel type.

Vowel classified according the 3-fold classification (`A` - a under stress, `a` - a/o as in the first syllable before the stressed one, `y` (stands for *shva*) - a/o as in the second etc. syllable before the stressed one or after the stressed syllable, cf. *g[y]g[a]t[A]l[y]*, *gogotala*, 'guffawed').

Our goal for today is to understand how the first formant depends on the values of the second formant and whether this relationship is different for different types of vowels.

Let us load this data set from the txt-file using `read.table()` function. Please, note that we should add the option `header=TRUE` to tell R that the first row should be read as a row with column names.

```
sh <- read.table("https://raw.githubusercontent.com/LingData2019/LingData/master/data/duryagin_Reductio
```

Look at the summary of our data and make sure all variables have correct types:

```
summary(sh)
```

```
##      time1            duration           time2              f2
##  Min.   :  3.659   Min.   :0.01455   Min.   :  3.721   Min.   :1242
##  1st Qu.: 39.082   1st Qu.:0.04388   1st Qu.: 39.109   1st Qu.:1315
##  Median : 88.232   Median :0.06336   Median : 88.266   Median :1372
##  Mean   : 93.227   Mean   :0.07272   Mean   : 93.261   Mean   :1386
##  3rd Qu.:143.149   3rd Qu.:0.09769   3rd Qu.:143.191   3rd Qu.:1462
##  Max.   :191.228   Max.   :0.16493   Max.   :191.244   Max.   :1577
##        f1              f3       vowel
##  Min.   :365.0   Min.   :1718   a:44
##  1st Qu.:483.5   1st Qu.:2162   A:68
##  Median :587.5   Median :2216   y:48
##  Mean   :592.8   Mean   :2220
##  3rd Qu.:695.2   3rd Qu.:2302
##  Max.   :860.0   Max.   :2426
```

As later we will work with mixed-effects models, it is important to understand how many rows with missing values our data frame has (mixed-effects models work correctly when the share of NA's is small).

Let's count rows with missing values:

```
# ! - negation of complete.cases()
sum(!complete.cases(sh))
```

```
## [1] 0
```

As we see, no rows with missing values are detected, we can go on.

In our data we have three groups of vowels (see above). Let's look at the summary statistics by groups:

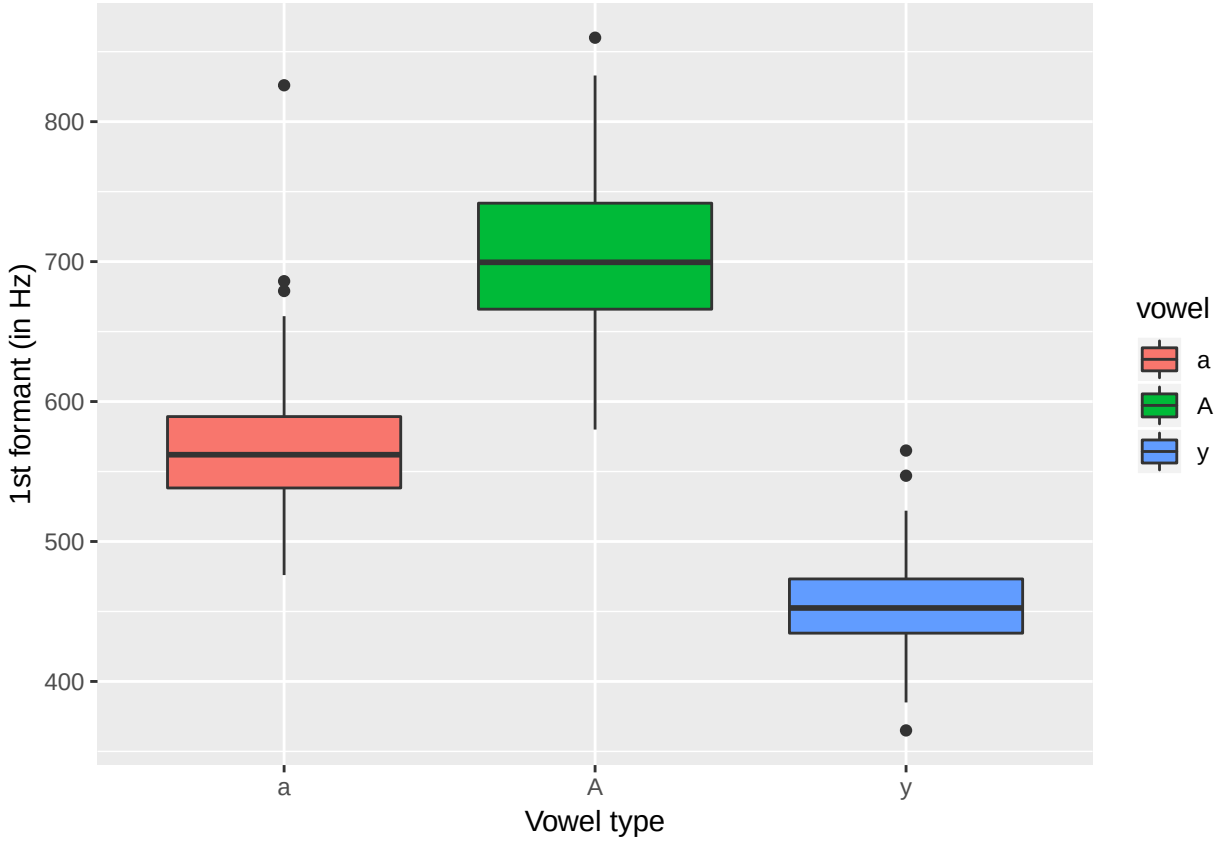
```
library(tidyverse)

sh %>% group_by(vowel) %>% summarise(n = n(),
                                     mean_f1 = mean(f1),
                                     mean_f2 = mean(f2))
```

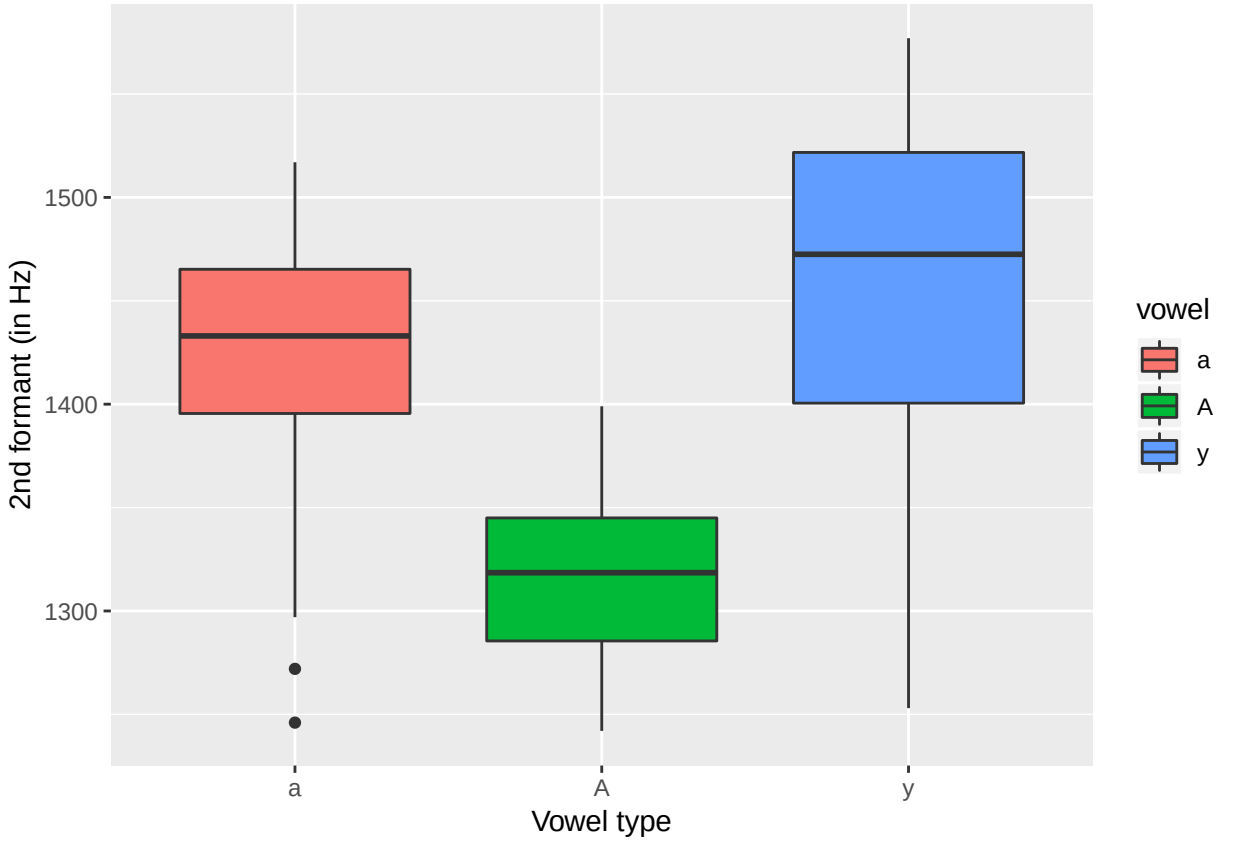
```
## # A tibble: 3 x 4
##   vowel     n mean_f1 mean_f2
##   <fct> <int>   <dbl>   <dbl>
## 1 a        44    573.   1421.
## 2 A        68    704.   1316.
## 3 y        48    454.   1452.
```

As we can see, the groups are approximately of the same size (balanced), and the mean values of the first formant and of the second formant differ by groups. Now we can visualize the distribution of formants by groups using `ggplot2`.

```
ggplot(data = sh, aes(x = vowel, y = f1, fill = vowel)) +
  geom_boxplot() +
  labs(x = "Vowel type", y = "1st formant (in Hz)")
```

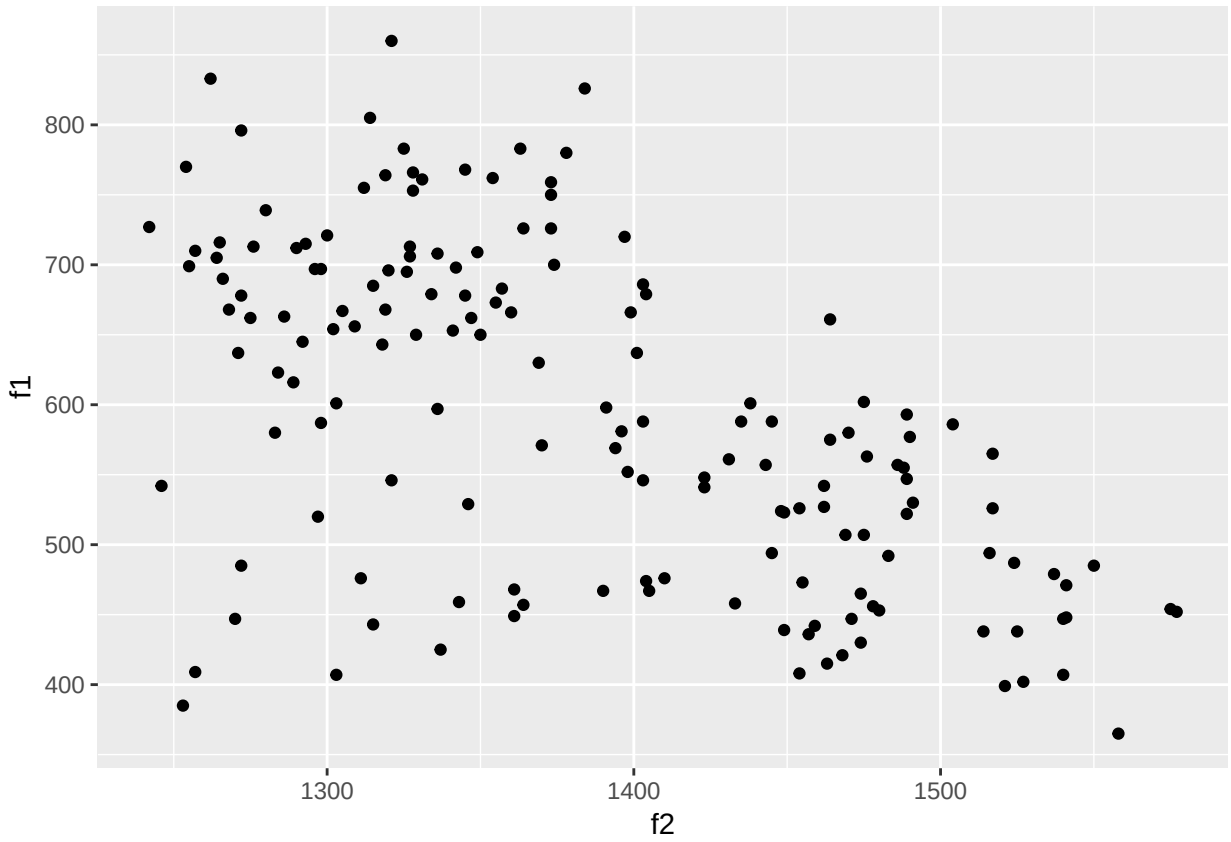


```
ggplot(data = sh, aes(x = vowel, y = f2, fill = vowel)) +
  geom_boxplot() +
labs(x = "Vowel type", y = "2nd formant (in Hz)")
```



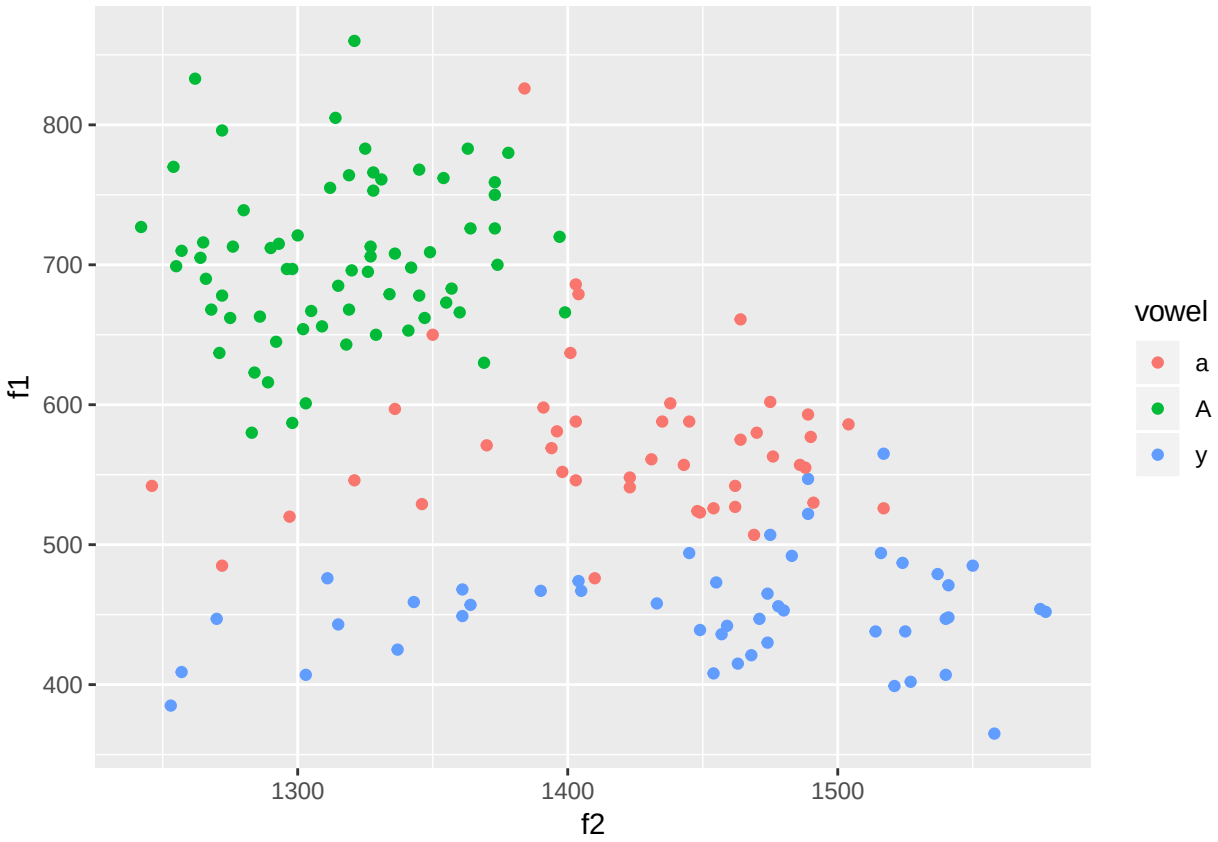
Again we can see the differences by groups: the median values of `f1` and `f2` are different, the range of values is also different. Now we visualize the relationship between `f1` and `f2`:

```
ggplot(data = sh, aes(x = f2, y = f1)) + geom_point()
```



This scatter plot is interesting. On the one hand, the relationship between `f1` and `f2` is negative: the higher are the values of `f2`, the lower the values of `f1` are. On the other hand, if we take a closer look, we will see that there are different groups of points, and the relationship between `f1` and `f2` can be different as well. Now let us add grouping to this graph:

```
ggplot(data = sh, aes(x = f2, y = f1, color = vowel)) + geom_point()
```



Now we can see that there are three different clusters, three groups of points that go one by one from the top to the bottom. If we try to add regression lines to all these clouds of points separately, the intercept will be certainly different, but slopes will be approximately the same. We can check it calculating correlation coefficients by groups:

```
# cor() can be written inside summarise()
sh %>% group_by(vowel) %>% summarise(corr = cor(f1, f2))
```

```
## # A tibble: 3 x 2
##   vowel    corr
##   <fct>   <dbl>
## 1 a     -0.0113
## 2 A      0.103
## 3 y      0.182
```

Correlation coefficients are quite low, not very different.

Bonus (for those who are interested):

If you need correlation coefficients by groups with p-values, you can get them as well:

```
sh %>% group_by(vowel) %>% summarise(corr = cor.test(f1, f2)$estimate,
                                     pvalue = cor.test(f1, f2)$p.value)
```

```
## # A tibble: 3 x 3
##   vowel    corr pvalue
##   <fct>   <dbl>  <dbl>
## 1 a     -0.0113  0.942
## 2 A      0.103   0.402
## 3 y      0.182   0.216
```

All correlation coefficients are insignificant at the 5% level of significance (and at any common significance level).

Now we can proceed to regression models. Let us start with a simple linear model. Fit a model `f1 ~ f2`:

```
sm <- lm(f1 ~ f2, data = sh)
summary(sm)
```

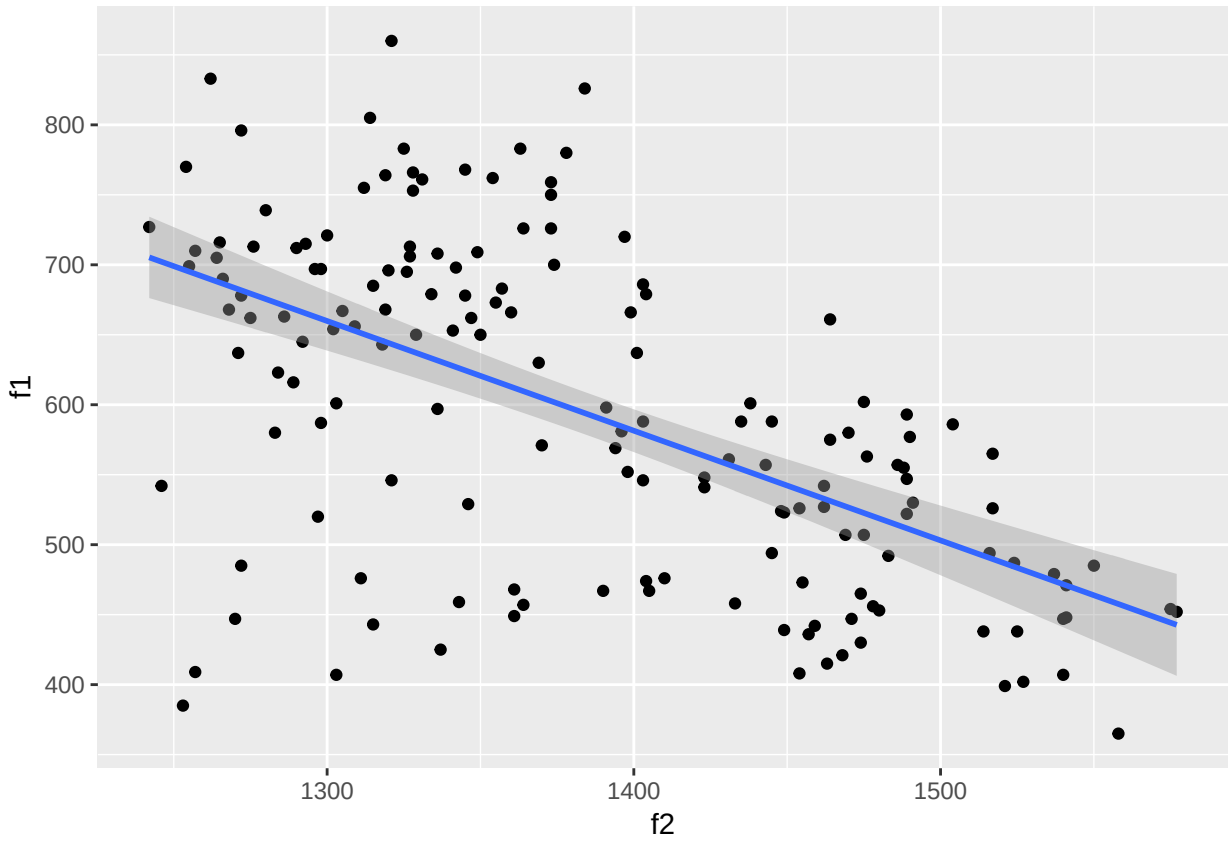
```
##
## Call:
## lm(formula = f1 ~ f2, data = sh)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -311.684  -54.682    9.209   56.291  232.010
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1678.94083  121.68477  13.797  < 2e-16 ***
## f2            -0.78392    0.08765  -8.944 9.53e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 96.69 on 158 degrees of freedom
## Multiple R-squared:  0.3361, Adjusted R-squared:  0.3319
## F-statistic: 79.99 on 1 and 158 DF,  p-value: 9.533e-16
```

Revise an interpretation.

Interpretation: the effect of the second formant on the first formant is statistically significant at the 5% (and even 0.1%) level of significance, we reject the null hypothesis about the coefficient equal to zero. If `f2` increases by one Hz, `f1` decreases by 0.78 on average.

We can add a regression line to our scatterplot:

```
ggplot(data = sh, aes(x = f2, y = f1)) + geom_point() +
  geom_smooth(method=lm)
```



Now let's fit a model with a categorical (factor, qualitative) predictor, vowel group.

```
sm_dummy <- lm(f1 ~ f2 + vowel, data = sh)
summary(sm_dummy)
```

```
##
## Call:
## lm(formula = f1 ~ f2 + vowel, data = sh)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -121.578  -33.689   -2.358   21.357  255.399
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  477.30253   95.35913   5.005 1.49e-06 ***
## f2             0.06741    0.06688   1.008    0.315
## vowelA       137.78567   12.40005  11.112  < 2e-16 ***
## vowely      -121.63215   11.22084 -10.840  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 52.86 on 156 degrees of freedom
## Multiple R-squared:  0.8041, Adjusted R-squared:  0.8003
## F-statistic: 213.4 on 3 and 156 DF,  p-value: < 2.2e-16
```

Why this model is different from the previous one? Now the coefficient of `f2` is positive! So, if we consider grouping, the effect of the second formant is not definitely negative. Moreover, it is insignificant. Hence, the

predicted (average) value of the first formant mainly depends on the vowel group.

The equation of this model is the following:

$$\text{f1} = 477.30 + 0.07 \times \text{f2} + 137.78 \times \text{vowelA} - 121.63 \times \text{vowely}$$

The factor variable `vowel` is split in a set of dummy variables:

- `vowela`: 1 if the word contains the first type vowel, 0 otherwise;
- `vowelA`: 1 if the word contains the second type vowel, 0 otherwise;
- `vowely`: 1 if the word contains the third type vowel, 0 otherwise.

Why do we have only two groups of vowels? The first one is taken as a base category and ommited (it usually happens so the model can be estimated). A base category is a reference group, one we compare other groups with. Thus, judging by equation, we can say that: 1) the average value of `f1` is higher by 137.78 for cases with `vowelA` type of vowel than for cases with `vowela` type of vowel; 2) the average value of `f1` is lower by 121.63 for cases with `vowely` type of vowel than for cases with `vowela` type of vowel.

Now let us fit a new type of a model, a linear mixed-effects model with a random effect on the intercept for groups based on vowel type. So as to do this, we will need the library `lme4`, let's install it.

```
install.packages("lme4")
```

Fit a model:

```
library(lme4)
me <- lmer(f1 ~ f2 + (1|vowel), data=sh, REML = FALSE)
```

Notes:

1. We add a random effect on the intercept for different vowel type, so we write `(1|vowel)`. Such a syntax with pipes (`|`) is usually used in mixed-effects models in R.
2. We could safely skip the option `REML = FALSE`. There are two basic methods of estimating mixed-effects models in R, maximum likelihood method (ML) and restricted maximum likelihood method (REML). REML is used by default as a more general one, but we can turn it off and use a simple ML method, especially if our groups are balanced (approx. of the same size).

Get the summary of this model:

```
summary(me)
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula: f1 ~ f2 + (1 | vowel)
##    Data: sh
##
##      AIC      BIC   logLik deviance df.resid
##   1746.8   1759.1   -869.4   1738.8      156
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.3025 -0.6403 -0.0554  0.3892  4.8413
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  vowel    (Intercept) 11103    105.37
##  Residual              2777     52.69
## Number of obs: 160, groups:  vowel, 3
##
## Fixed effects:
```

```
##              Estimate Std. Error t value
## (Intercept) 492.60132  111.11639   4.433
## f2            0.06036    0.06653   0.907
##
## Correlation of Fixed Effects:
##    (Intr)
## f2 -0.836
```

Interpretation:

1. First, we see some measures of model quality, for example, Akaike information criterion (AIC) and Bayesian information criterion (BIC). It is useless to interpret the AIC as is, we can only compare AICs of two models and choose one that has a lower AIC (if it is substantially correct, of course).
2. Then, we have some statistics on the random effects we added. There is the variance of the intercept and the variance of residuals. We can calculate the share of variance that is explained by random effects on groups:

$$\text{ICC} = \frac{11103}{11103 + 2777} = 0.799$$

This measure is called *intraclass correlation (ICC)* and shows whether the random effects we added on groups are really needed. In other words, how much of the variance of the dependent variable is expained by grouping. If ICC is very close to zero, it means that random effects are not really needed, we can safely use a more simple, an ordinary regression model. In our case this share is high, so it is sensible to use different intercepts for different groups in our model.

We can also calculate ICC using the `icc()` function from the `sjstats` library:

```
library(sjstats)
icc(me) # the same
```

```
##
## Intraclass Correlation Coefficient for Linear mixed model
##
## Family : gaussian (identity)
## Formula: f1 ~ f2 + (1 | vowel)
##
##   ICC (vowel): 0.7999
```

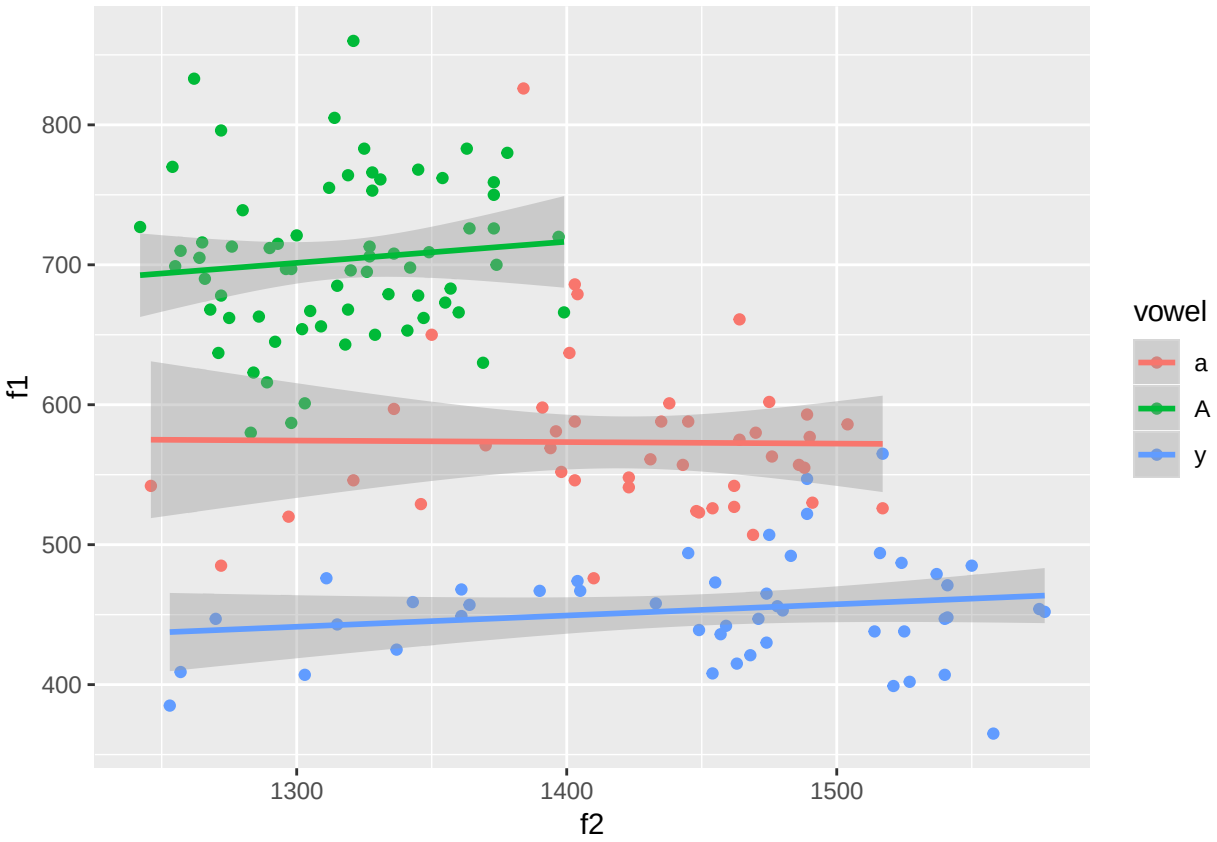
3. Coefficients from the *Fixed effects* part can be used as ordinary coefficients of independent variables in linear models. They are computed taking into account the differences between groups, so the coefficient of `f2` is not drastically different from one from the model with dummy variables for vowel types above, but different from one from the very first simple model.

We can write an equation of this model:

$$\text{f1} = 492.60 + 0.06 \times \text{f2}$$

Now let's visualise the results and add a regression line for each group of vowels to the scatter plot:

```
ggplot(data = sh, aes(x = f2, y = f1, color = vowel)) + geom_point() +
  geom_smooth(method=lm)
```



As we see, slopes are approximately the same, but intercepts are different.