Linguistic Data: Quantitative Analysis and Visualisation

Multiple linear regression

Ilya Schurov, Olga Lyashevskaya, George Moroz, Alla Tambovtseva

Let’s work with the data set with the results of a psycholinguistic experiment dedicated to lexical decision and word naming. The brief description of the experiment is the following. In studies of lexical decision paticipants (also called subjects) are asked to decide whether the word shown on the screen is a real word or not. In other words, whether a word exists in the language or it is just an artificical word created using grammatical rules. In studies of word naming participants are asked to read a word shown aloud. Then the reaction time in miliseconds is measured: how fast a person clicks on the button *word* or *non-word* (lexical decision) or pronouns a word (word naming).

Load data on English words:

```
eng <- read.csv("http://math-info.hse.ru/f/2018-19/ling-data/english.csv")
```

Some of the variables:

- `AgeSubject`: age group of the subject: `young` versus `old`;
- `WordCategory`: word categories `N` (noun) and `V` (verb);
- `RTlexdec`: reaction time in visual lexical decision;
- `RTnaming`: reaction time in in word naming;
- `WrittenFrequency`: ;
- `LengthInLetters`: length of the word in letters;
- `FamilySize`: log morphological family size;
- `NumberSimplexSynsets`: the log-transformed count of synonym sets in WordNet in which the word is listed.

To see the complete description of this dataset (original one, we use a modified version here, for example, we work with reaction time in ms, not with its logarithm), you can install the library `languageR` and get the documentation for `english`:

```
?english
```

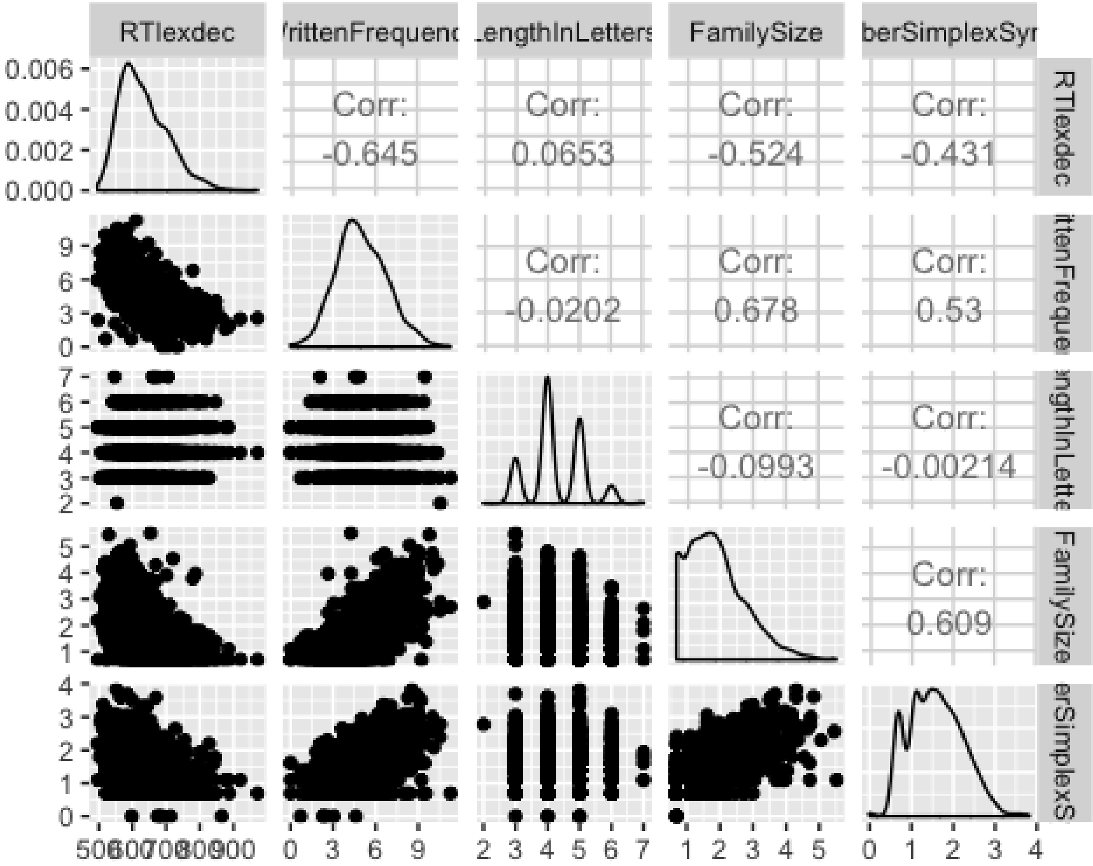
To make things simple we can choose variables for analysis and take a smaller sample — only young people.

```
library(tidyverse)
young <- eng %>% filter(AgeSubject == "young", WordCategory ==
"N")
small <- young %>% select(RTlexdec, WrittenFrequency,
                          LengthInLetters, FamilySize,
                          NumberSimplexSynsets)
```

Now we will work with the data frame `small`. We want to create a model that will explain how the reaction time in lexical decision depends (in statistical terms, not in cause-and-effect terms) on several features, namely word frequency, word length in letters, morphological family size and number of synonyms (synsect).

Before proceeding to models, usually it is really helpful to visualise the relationships between variables of interest. We can use the library `GGally` we discussed before.

```
library(GGally)
ggpairs(small)
```



At the first sight, all associations are logical: the more frequent is a word, the lower is the reaction time (people react quickly on frequent words), the longer is a word, the higher the reaction time and so on.

Now let's create a multiple linear regression model. To do so we use the same function `lm()` as for a bivariate linear model:

```
model <-  lm(data=small, RTlexdec ~ WrittenFrequency +
LengthInLetters + FamilySize + NumberSimplexSynsets)
summary(model)

##
## Call:
## lm(formula = RTlexdec ~ WrittenFrequency + LengthInLetters +
##     FamilySize + NumberSimplexSynsets, data = small)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -221.162  -36.239   -5.461   33.623  270.033
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            753.474      8.614  87.469  < 2e-16 ***
## WrittenFrequency       -20.870      1.098 -19.012  < 2e-16 ***
## LengthInLetters          3.706      1.708   2.170 0.030200 *
## FamilySize              -9.156      2.374  -3.857 0.000120 ***
## NumberSimplexSynsets    -9.598      2.867  -3.348 0.000836 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 53.27 on 1447 degrees of freedom
## Multiple R-squared:  0.4358, Adjusted R-squared:  0.4342
## F-statistic: 279.4 on 4 and 1447 DF,  p-value: < 2.2e-16
```

Interpretation:

- At the 5% level of significance for every independent variable we can reject the null hypothesis about the coefficient being equal to zero.

$$H_0:\beta_1=0\left(\mathit{no\ effect\ of\ independent\ variable\ on\ dependent}\right)$$

$$H_1:\beta_1\neq0\left(\mathit{there\ is\ effect\ of\ independent\ variable\ on\ dependent}\right)$$

  To be more precise, we can conclude that all coefficients are statistically significant, the effect of `WrittenFrequency` is significant at 0.001 level (0.1%, ***), the effect of `LengthInLetters` is significant at 0.05 (5%, *), the effects of `FamilySize` and `NumberSimplexSynsets` are also significant at 0.001 level (0.1%, ***).

- Equation of the model based on the output:

$$\mathit{RTlexdec}=753.47-20.87\times \mathit{WrittenFrequency}+3.71\times \mathit{LengthInLetters}-$$

$$-9.16\times \mathit{FamilySize}-9.6\times \mathit{NumberSynsets}$$

Thus, as all coefficients are significant, it is worth describing their effect in more detail. Some interpretations:

1. All else equal (*ceteris paribus*) if the written frequency of a word increases by 1, the reaction time decreases by 20.87 ms on average.

2. All else equal (*ceteris paribus*) if the length of a word increases by 1 letter, the reaction time increases by 3.71 ms on average.

- As for model quality, we can look at the last line in the output. Here $R^2=0.44$. There are no strict rules how to decide which R^2 is high enough, all depends on the field. Basically, values greater than 0.3-0.4 are sufficient. However, in statistics as we are mostly interested in relationships between variables, in interactions between them, in model design in general, a substantially plausible model is better than a poorer model but with higher R^2.

  At the last line in the output there is also one more p-value. What hypothesis is tested here?

$$H_0: \beta_1=\beta_2=\ldots=\beta_k=0\,(\textit{model is not better than the model with the intercept only})$$

$$H_1: \textit{at least one } \beta_i \neq 0\,(\textit{model is better than the model with the intercept only})$$

Informally, here we test the hypothesis that a model constructed is needed (not useless). Here p-value is close to 0, so we reject H_0 and conclude that at least one coefficient is statistically different from zero, model is worth considering. Frankly speaking, seldom can we construct a model that is completely useless, so where p-value is too high.

We can extract some elements of this output separately, for example, coefficients, residuals or fitted values (values of reaction time predicted by our model).

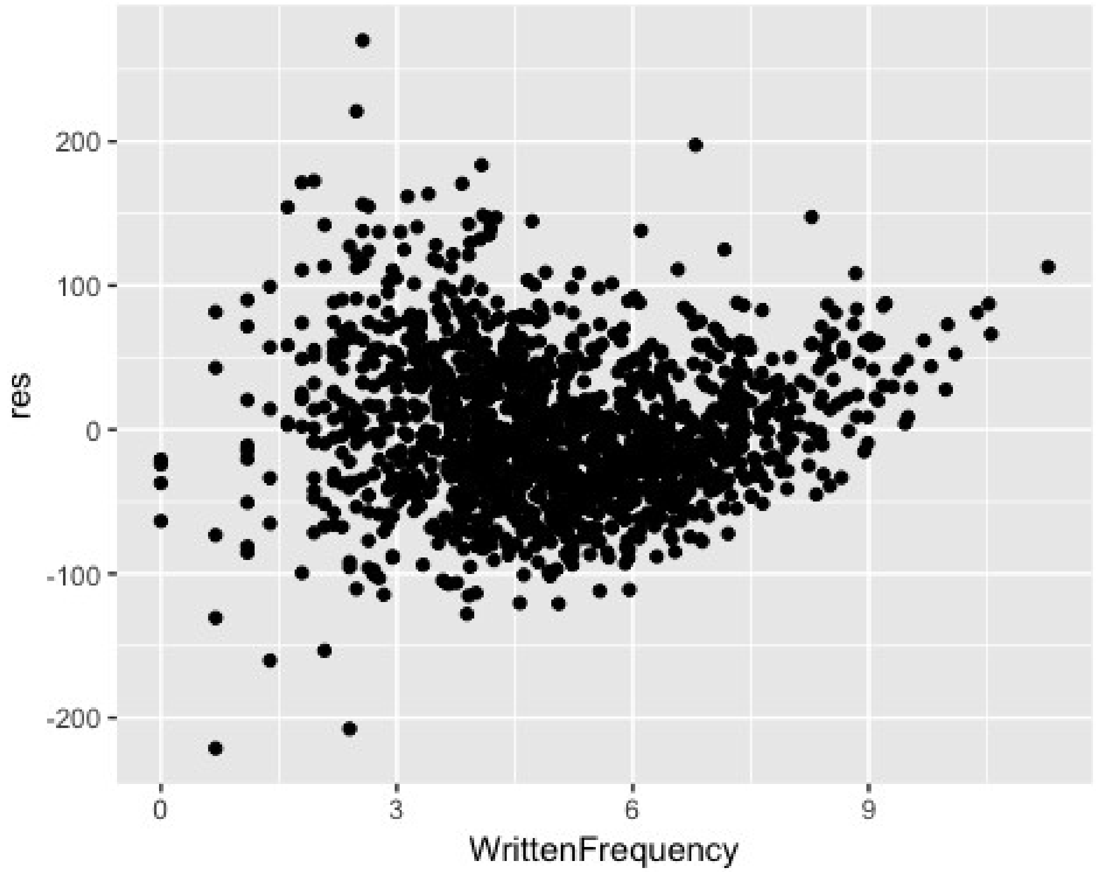
```
model$coefficients

##          (Intercept)     WrittenFrequency      LengthInLetters

##           753.474472           -20.870361             3.705659

##           FamilySize NumberSimplexSynsets
##            -9.156051            -9.597713

model$residuals[0:10]

##         1         2         3         4         5         6
7
##  31.28950 -53.99426 -54.07484 -18.62364  10.81357  46.15871 -
23.41527
##         8         9        10
## -34.62099 109.13492 -76.78837
```

```
model$fitted.values[0:10]
```

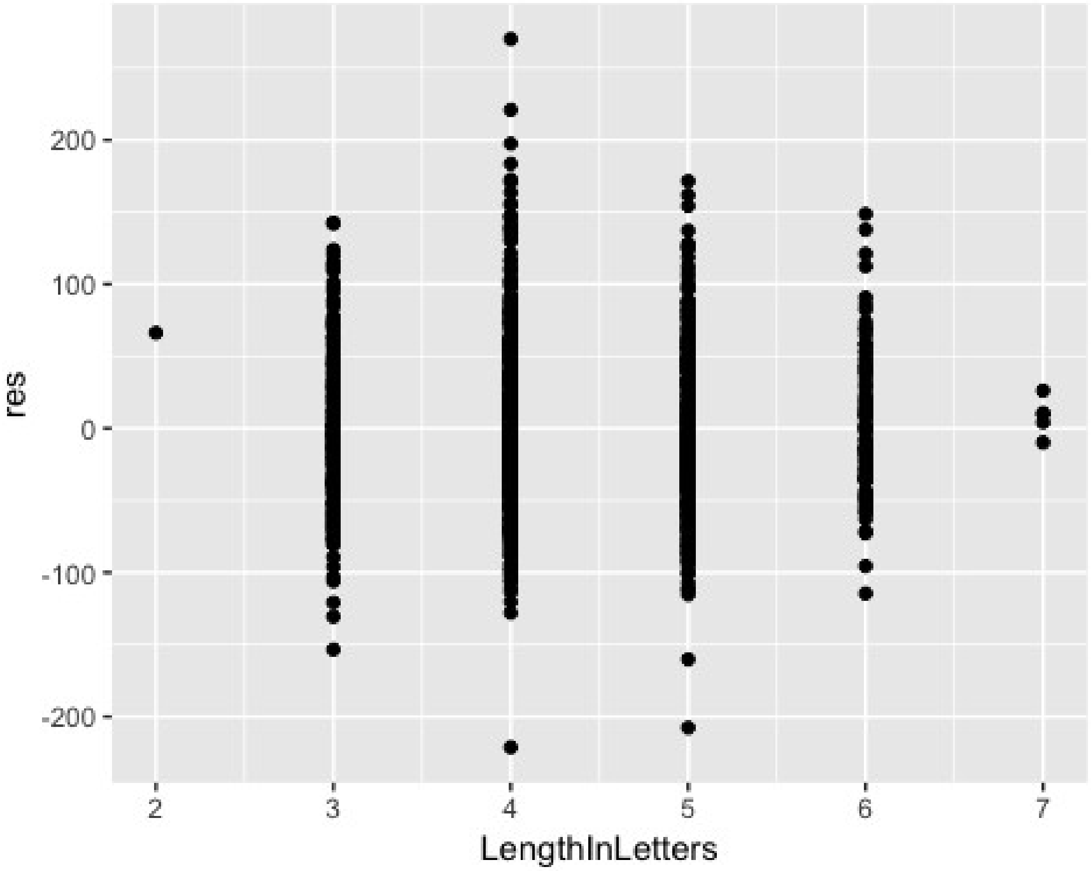
```
##        1        2        3        4        5        6        7
8
## 663.6005 654.3943 601.3448 635.2236 622.2664 640.5913 607.8153
561.4410
##        9       10
## 632.3451 613.1684
```

To check the fit of our model we can look at several graphs of residuals of this model. First, let us plot graphs *independent variable vs residuals* for every independent variable in our model. For convenience we can save residuals as a separate column in `small` data frame.

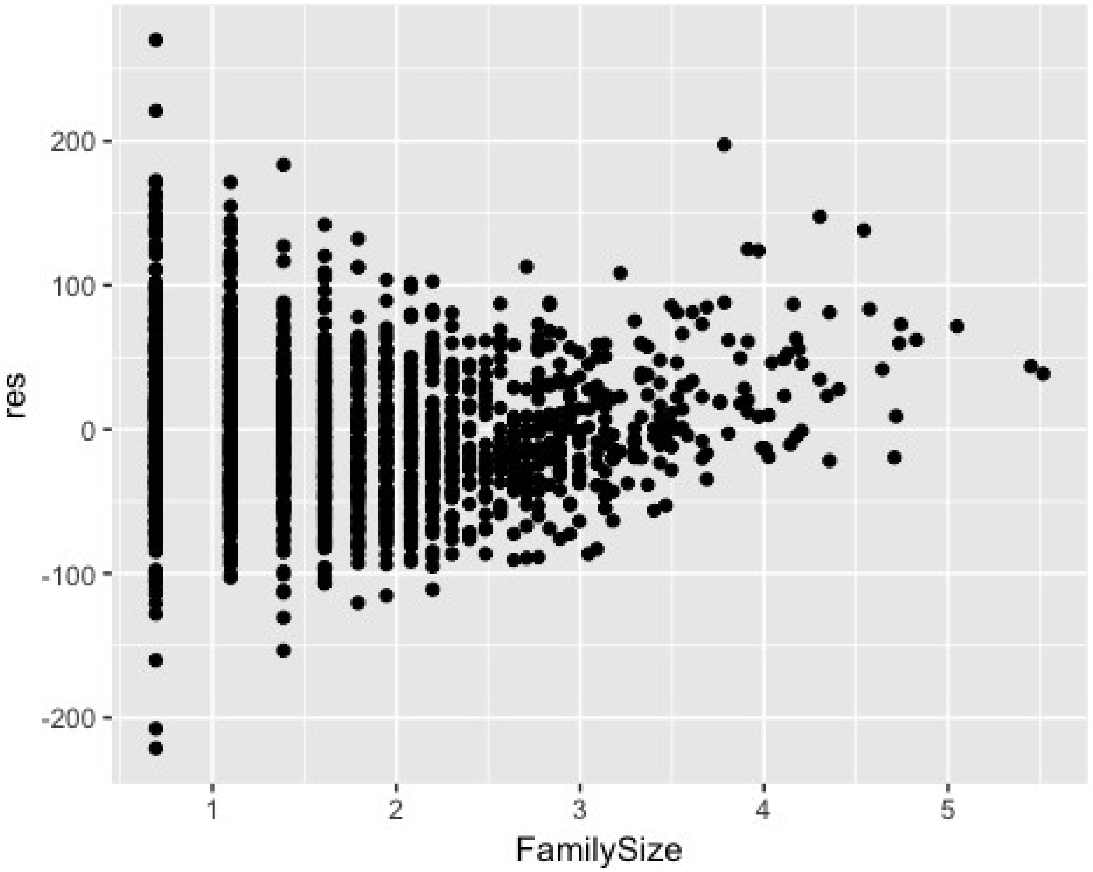
```
small <- small %>% mutate(res = model$residuals)
ggplot(data = small, aes(x = WrittenFrequency, y = res)) +
geom_point()
```



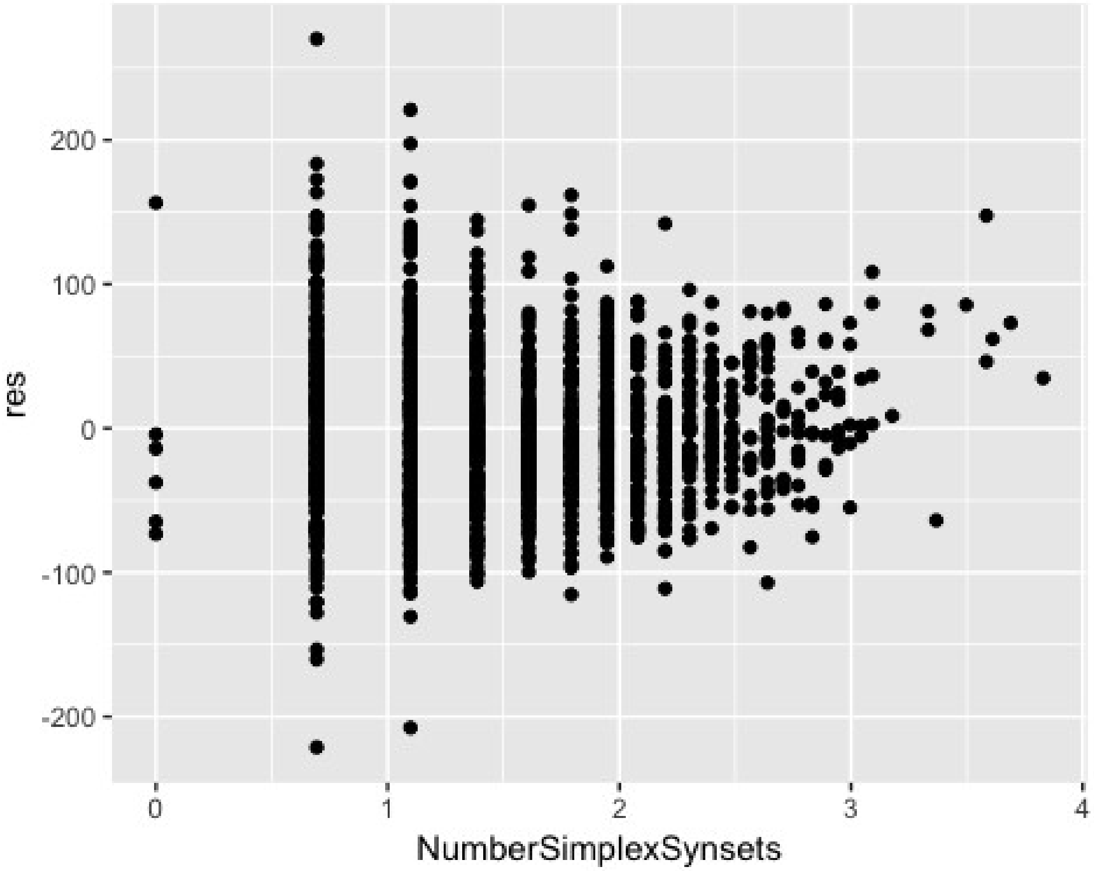
```
ggplot(data = small, aes(x = LengthInLetters, y = res)) +
geom_point()
```



```r
ggplot(data = small, aes(x = FamilySize, y = res)) + geom_point()
```



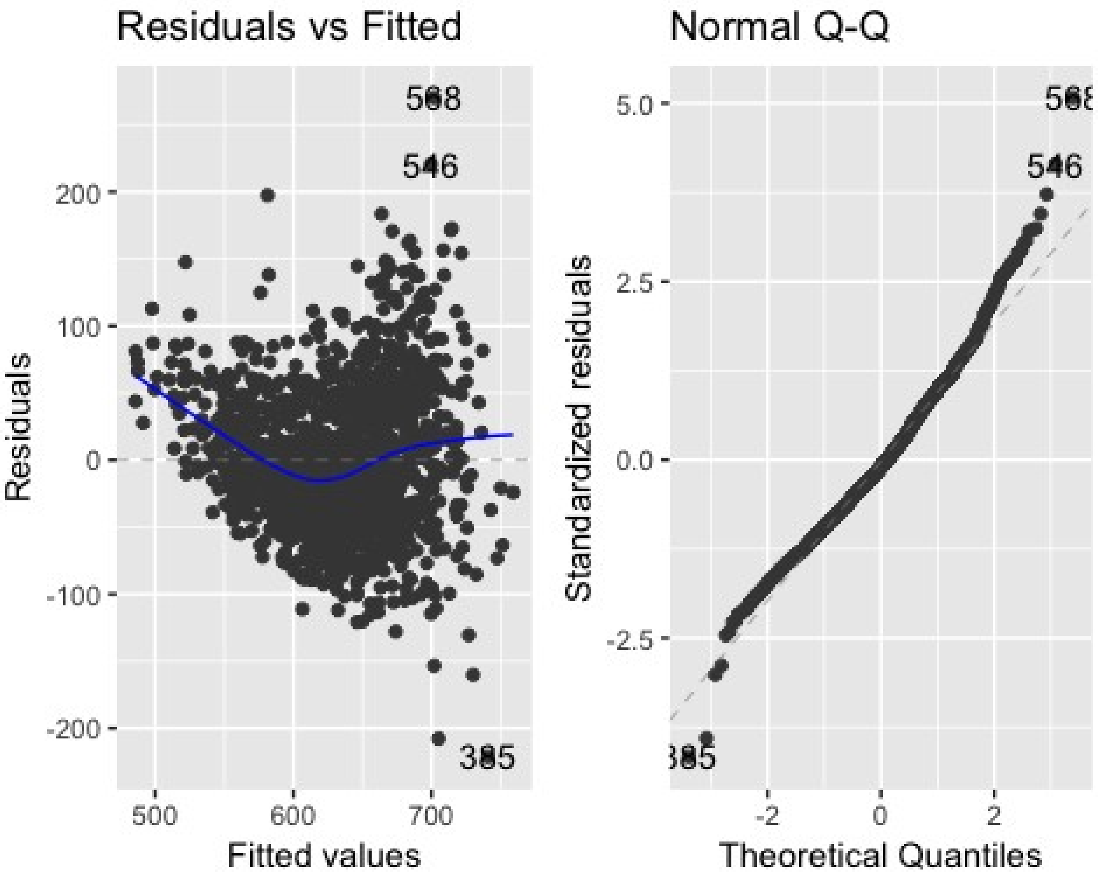
```
ggplot(data = small, aes(x = NumberSimplexSynsets, y = res)) +
geom_point()
```



Interestingly, in most these graphs we can see a slight non-linear pattern of points, so, it can serve as a sign that our model might have problems with specification. However, usually it is sensible to consider substantial knowledge about the variables we study (based on processes known, previous research. etc.)

Let's add more residuals plots. To do this, we need the library `ggfortify`.

```
install.packages("ggfortify")

library(ggfortify)
# which - which graphs to use (we chose two first)
autoplot(model, which = 1:2)
```



Interpretation:

- Graph on the left also allows us to detect non-linear patterns in residuals. It is *fitted values vs residuals*. Here again we have a contraversial situation: at the one hand, no clear quadratic/cubic/any non-linear relationship is seen here, but, on the other hand, in the middle we see some non-linearity.
- Graph on the right is called Q-Q plot or qqplot that stands for *quantiles-to-quantiles plot* because it compares theoretical quantiles of a certain distribution and empirical quantiles we get on our data. Here we have a normal Q-Q plot since it compares the distribution of residuals (points) with the normal distribution (reference line as a diagonal here). As we can see, there are some points on the left and on the right that are relatively far from the reference line, but in general, no serious deviations observed.

Now let us export the regression results into a document so as to get a regression table we usually see in academic papers or projects. By default R works with LaTeX code, however, if you work with Word/Libre Office/Open Office only, you can export the results into a doc-file as well. To do this you need the `stargazer` library. This name is derived from the expression “gaze at stars”, so stars for significance that we have seen in the regression output.

```
install.packages("stargazer")
```

We add `type=html` so as not to get LaTeX code and specify the name of the file.

```
library(stargazer)
stargazer(model, type = "html", out = "my model.doc")
```

Note: stars in `stargazer` are added in a slightly different way: one star corresponds to the 10% significance (`p<0.1`), two stars correspond to the 5% significance (`p<0.05`), three stars correspond to 1% significance (`p<0.01`).