Continuous variables

WiSe23/24

Daniela Palleschi

2023-10-10

Table of contents

This lecture is based on Ch. 5 (Correlation, Linear, and Nonlinear transformations) from Winter (2019).

Learning Objectives

Today we will learn…

- why and how to centre continuous predictors
- when and how to standardize continuous predictors
- why and how to log-transform continuous variables

Set-up environment

```
# suppress scientific notation
options(scipen=999)
```

```
# load libraries
pacman::p_load(
               tidyverse,
               here,
               broom,
               lme4,
               janitor,
               languageR)
```

Load data

```
df_freq <- read_csv(here("data", "ELP_frequency.csv")) |>
  clean_names()
```

- Reminder of our variables:

```
summary(df_freq)
```

```
    word                freq                rt        
Length:12          Min.   :    4.0   Min.   :507.4  
Class :character   1st Qu.:   57.5   1st Qu.:605.2  
Mode  :character   Median :  325.0   Median :670.8  
                   Mean   : 9990.2   Mean   :679.9  
                   3rd Qu.: 6717.8   3rd Qu.:771.2  
                   Max.   :55522.0   Max.   :877.5  
```

Linear transformations

- refer to constant changes across values that do not alter the relationship between these values
  - adding, subtracting, or multiplying by a constant value

- let's look at some common ways of linearly transforming our data, and the reasons behind doing so

Centering

- Centering is typically applied to predictor variables
  - subtracting the mean of a variable from each value
  - results in each centered value representing the original value's deviance from the mean (i.e., a mean-deviation score)
- What would a centered value of 0 represent in terms of the original values?

- let's centre our frequency values

`tidyverse`

```
# add centered variable
df_freq <-
  df_freq |>
  mutate(freq_c = freq-mean(freq))
```

base R

```
# add centered variable with base R
df_freq$freq_c <- df_freq$freq-mean(df_freq$freq)
```

Now let's fit our models.

```
# run our model with the original predictor
fit_rt_freq <-
  lm(rt ~ freq, data = df_freq)
```

```
# run our model with the centered predictor
fit_rt_freq_c <-
  lm(rt ~ freq_c, data = df_freq)
```

If we compare the coefficients from `fit_rt_freq` and `fit_rt_freq_c`, what do we see? The only difference is the intercept values: 713.706298 (uncentered) and 679.9166667 (centered).

```
mean(df_freq$rt)
```

```
[1] 679.9167
```

The intercept for a centered continuous predictor variable corresponds to the mean of a continuous response variable. This is crucial in interpreting interaction effects, which we will discuss tomorrow. For more detail on interpreting interactions, see Chapter 8 in Winter (2019) (we won't be discussing this chapter as a whole).

> Centering interval data
>
> If you have interval data with a specific upper and lower bound, you could alternatively subtract the median value. In linguistic research, this is most typically rating scale data. For example, if you have a dataset consisting of ratings from 1-7, you can centre these ratings by subtracting 4 from all responses. A centred response of -3 would correspond to the lowest rating (1), and of +3 to the highest rating (7), which 0 would correspond to a medial rating (4). This can also be helpful in plotting, as there is no question as to whether 1 or 7 was high or low, because all ratings are now centred around 0 (and negative numbers correspond to our intuition of low-ratings).

Standardizing (*z*-scoring)

We can also standardize continuous predictors by dividing centered values by the standard deviation of the sample. Let's look at our frequency/reaction time data again.

First, what are our mean and standard deviation? This will help us understand the changes to our variables as we center and stardardize them.

```
mean(df_freq$freq)
```

```
[1] 9990.167
```

```
sd(df_freq$freq)
```

```
[1] 18558.69
```

What are the first six values of `freq` in the original scale?

```
df_freq$freq[1:6]
```

```
[1] 55522 40629 14895  3992  3850   409
```

What are the first six values of `freq_c` in the centered scale? These should be the values of `freq` minus the mean of `freq` (which we saw above is 9990.1666667).

```
df_freq$freq_c[1:6]
```

```
[1] 45531.833 30638.833  4904.833 -5998.167 -6140.167 -9581.167
```

Now, let's create our standardised z-scores for frequency by dividing these centered values by the standard deviation of `freq` (which will be the same as the standard deviation of `freq_c`), and which we saw is 18558.6881679. Again, this can be done with `mutate()` from `dplyr`, or by using base R syntax.

```
# standardise using the tidyverse
df_freq <-
  df_freq |>
  mutate(freq_z = freq_c/sd(freq))
```

```
# standardize with base R
df_freq$freq_z <- df_freq$freq_c/sd(df_freq$freq)
```

```
head(df_freq)
```

```
# A tibble: 6 x 5
  word      freq    rt freq_c freq_z
  <chr>    <dbl> <dbl>  <dbl>  <dbl>
1 thing    55522  622. 45532.  2.45
2 life     40629  520. 30639.  1.65
3 door     14895  507.  4905.  0.264
4 angel     3992  637. -5998. -0.323
5 beer      3850  587. -6140. -0.331
6 disgrace   409  705  -9581. -0.516
```

Correlation

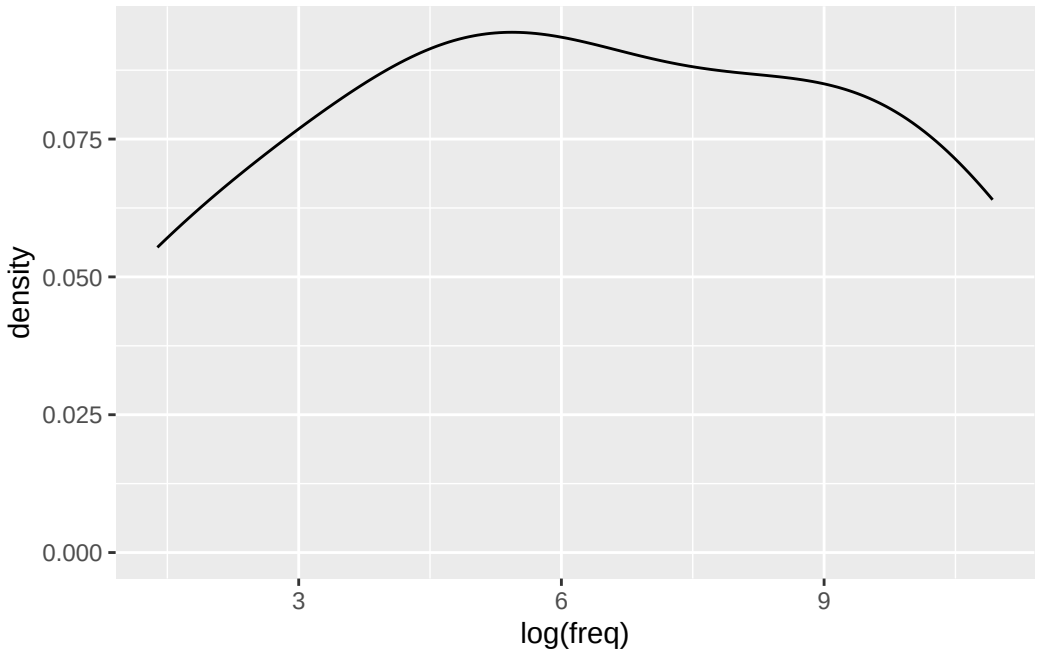
Non-linear transformations

This is really the meat and potates of dealing with continuous variables (depending on your subfield). In linguistic research, and especially experimental research, we often deal with continuous variables truncated/bound at 0. Reaction times, reading times and formant frequencies are all examples of such types of data: there is no such thing as a negative reading time or fundamental frequency. The problem with these types of data is that they are almost never normally distributed, which has implications for the normality of residuals for any line that tries to fit to these data. Very often, this type of data will have a 'positive skew', or a long tail off to the right (assuming larger values are plotting to the right). This shape is not symmetrical, meaning that the residuals tend to be much larger for larger values. It is also often the case that these very large, exceptional values will have a stronger influence on the line of best fit, leading to the coefficient estimates that are "suboptimal for the majority of data points" [@Baayen (2008); p. 92]. How do we deal with this nonnormality? We use non-linear transformations, the most common of which is the log-transformation.

Log-transformation

Let's look at our reaction time data again. We'll log-transform our reaction time data and frequency data. Note that in Winter (2019), frequency is transformed using log to the base 10 for interpretability, but we'll stick to the natural logarithm.

```
df_freq |>
  ggplot() +
  aes(x = log(freq)) +
  geom_density()
```




```r
df_freq <-
  df_freq |>
    mutate(rt_log = log(rt),
           freq_log = log(freq))
```

```r
lm(rt_log ~ freq_log, data = df_freq) |> tidy()
```

```
# A tibble: 2 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)   6.79     0.0611     111.   8.56e-17
2 freq_log     -0.0453   0.00871     -5.20 4.03e- 4
```

```r
# or, log-transform directly in the model syntax
lm(log(rt) ~ log(freq), data = df_freq) |> tidy()
```

```
# A tibble: 2 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)   6.79     0.0611     111.   8.56e-17
2 log(freq)    -0.0453   0.00871     -5.20 4.03e- 4
```

Learning Objectives

Today we learned...

- why and how to centre continuous predictors
- when and how to standardize continuous predictors
- why and how to log-transform continuous variables

Important terms

Term	Definition	Equation/Code
Centering	type of linear transformation	‘dplyr::mutate(variable = variable - mean(variable))‘

Take-home messages

Continuous data are often transformed before fitting a model to this data. Linear transformations, like adding or multiplying all values by a single value, are often performed on continuous predictors by means of centring and standardizing (when there are multiple continuous predictors). Non-linear transformations are often performed on continuous data with a positive skew (a few values much larger than the majority) in order to satisfy the normality assumption. Although the normality assumption refers to the normality of *residuals*, the distribution of the data will have implications for the distribution of the residuals. The most common non-linear transformation is the log-transformation (the inverse of the exponential), which shrinks values, especially making big numbers smaller. This has the result of squeezing big numbers towards smaller numbers, reducing the spread in the distribution (e.g., the log of 3 is 1.0986123, the log of 30 is 3.4011974, and the log of 30 is 5.7037825).

What to do with this information? If you have continuous data truncated at 0 (with no upperbound, e.g., reaction time data or fundamental frequency), visualise the data (histogram and Q-Q plot) in order to check its distribution. If it is not normally distributed, you will likely want to log-transform it. Is this data your *response* variable? Then that is all you will likely want to do. Is this data a *predictor* variable? Then you will want to centre it (subtract the mean of this variable from all values). Do you have more than one continuous predictor variable? Then standardizing these variables will facilitate the interpretation of interaction effects (we'll talk about these soon).

Task

Literaturverzeichnis

Baayen, R. H. (2008). *Analyzing Linguistic Data: A Practical Introduction to Statistics using R.*

Winter, B. (2019). Statistics for Linguists: An Introduction Using R. In *Statistics for Linguists: An Introduction Using R.* Routledge. https://doi.org/10.4324/9781315165547