Categorical predictors

WiSe23/24

Daniela Palleschi

2023-10-11

Table of contents

Learning Objectives

Today we will learn…

- about cateogorical predictors
- how to interpret different contrast coding

Set-up environment

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

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

Load data

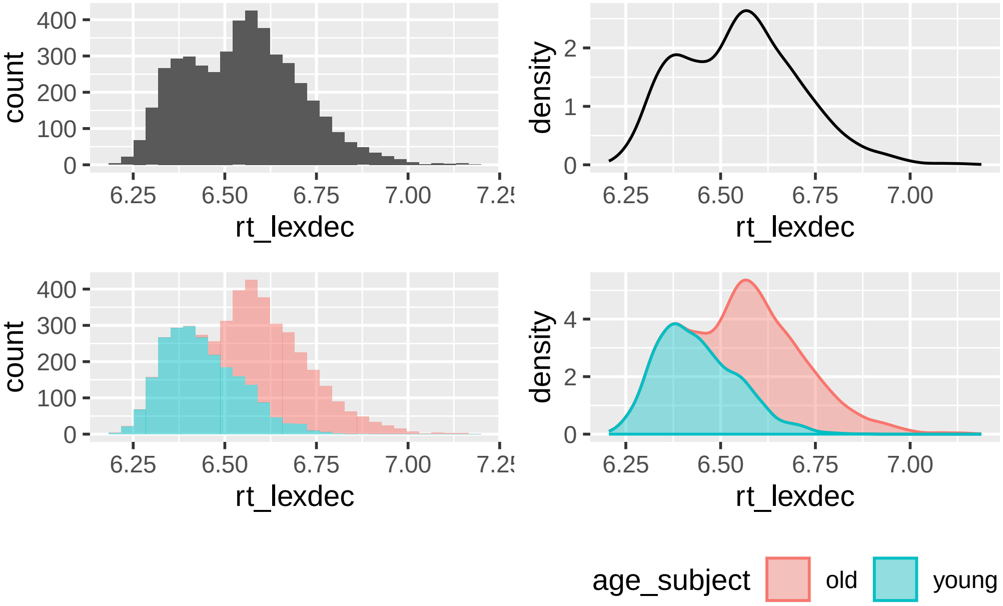
- load in the the dataset from the `languageR` package

```
df_freq_eng <-
  as.data.frame(english) |>
```

```
# keep relevant variables
dplyr::select(RTlexdec, RTnaming, Word, LengthInLetters, AgeSubject, WrittenFrequency) |
# rename some variables
rename(rt_lexdec = RTlexdec,
       rt_naming = RTnaming,
       freq_written = WrittenFrequency) |>
clean_names() |>
# standardize continuous predictors
mutate(
  freq_z = scale(freq_written),
  length_z = scale(length_in_letters)
) |>
# move 'word' to front
relocate(word) |>
# arrange alphabetically by 'word'
arrange(word)
```

Bimodal distribution

- in your exploratory data analysis, you might've noticed a *bimodal* distribution.



Bimodal distribution

- this is a *bimodal* distribution
  - there are two *modes* (most frequent value, i.e., peak in a histogram)
- We know that there were two subject groups: old and young
  - it might be that each group has a different mode

Re-run our model

- re-run our multiple regression model (reaction times ~ frequency + length)

```
fit_freq_length <-
  lm(rt_lexdec ~ freq_z*length_z,
     data = df_freq_eng)
```

Model fit and overfitting

```
glance(fit_freq_length)$r.squared
```

```
[1] 0.1896649
```

```
glance(fit_freq_length)$adj.r.squared
```

```
[1] 0.1891323
```

- seems like we don't have any overfitting in our model (R^2 and adjusted R^2 are comparable)

Model coefficients

- look at our coeffiecients.

```
tidy(fit_freq_length) |> select(term, estimate)
```

```
# A tibble: 4 x 2
  term             estimate
  <chr>               <dbl>
1 (Intercept)       6.55
2 freq_z           -0.0682
3 length_z          0.00328
4 freq_z:length_z -0.00196
```

- looks similar to the dataset we explored yesterday
- the bimodal distribution we saw earlier suggests age group could be an important a predictor
- does the effect of frequency and length also differ as a function of age group?

Categorical predictors

- we'd predict longer reading times for older participants than younger participants
  - although we should hypothesise *before* collecting and visualising our data!
- though age is numerical, all we have is two categories: old or young

Including a categorical predictor

- include `age_subject` in our model

```
fit_age <-
  lm(rt_lexdec ~ freq_z*length_z + age_subject,
     data = df_freq_eng)
```

Model fit

- compare R^2 and adjusted R^2
- R^2 our model without age as a predictor:

```
# rt_lexdec ~ freq_z*length_z
glance(fit_freq_length)$adj.r.squared
```

```
[1] 0.1891323
```

- R^2 our model with age as a predictor:

```
# rt_lexdec ~ freq_z*length_z + age_subject
glance(fit_age)$r.squared
```

```
[1] 0.6888949
```

- adjusted R^2 our model with age as a predictor:

```
# rt_lexdec ~ freq_z*length_z + age_subject
glance(fit_age)$adj.r.squared
```

```
[1] 0.6886222
```

- large increase in proportion of variance explained when we include age
- and the R^2 and adjusted R^2 values are comparable for the model with age
- this suggests that age captures variance that was not explained without it

Check for absence of collinearity

```
car::vif(fit_age)
```

```
         freq_z        length_z     age_subject freq_z:length_z
       1.012553        1.004461        1.000000        1.008108
```

- VIF values for all coefficients are near 1
  - this indicates that our predictors all contribute to the variance explained by the model and are not correlated

Contrasts

- let's take a look at our model estimates

```
tidy(fit_age) |> select(term,estimate)
```

```
# A tibble: 5 x 2
  term             estimate
  <chr>               <dbl>
1 (Intercept)       6.66
```

```
2 freq_z           -0.0682
3 length_z          0.00328
4 age_subjectyoung -0.222
5 freq_z:length_z  -0.00196
```

- there is a negative slope for `age_subjectyoung`
  - reaction times decrease when…what?
- how does a categorical variable get fit to a line?
- the factor levels (i.e., the categories in a categorical variable) are given numerical values
  - We call these numerical values mapped onto factor levels contrast coding

Dummy coding/treatment contrasts

- we can check the contrasts with `contrasts()`

```
contrasts(df_freq_eng$age_subject)
```

```
      young
old       0
young     1
```

- `old` was coded at 0 and `young` as 1
- our slope for `age_subjectyoung` represents the change in reaction times when we move from `old` to `young`
- this is called `treatment coding` (a.k.a., dummy coding), where one factor level is coded as 0 and the other as 1

Age-only model

- remove frequency and length to focus on `age_subject`
- use raw reaction times, to more easily interpret the results

```
fit_age <-
  lm(exp(rt_lexdec) ~ age_subject,
     data = df_freq_eng)
```

- what's the variance explained by our (simple) model with only age as a predictor?

```
glance(fit_age)$r.squared
```

```
[1] 0.4682224
```

- R^2 is lower than when we included frequency and length
  - but higher than our model with frequeny and length but no age

Age-only coefficients

```r
tidy(fit_age) |> select(term, estimate)
```

```
# A tibble: 2 x 2
  term             estimate
  <chr>               <dbl>
1 (Intercept)          787.
2 age_subjectyoung    -157.
```

- reaction times decrease by 157ms going from old to young group compared to the old group
- what does the intercept represent here?

```r
df_freq_eng |>
  select(rt_lexdec, age_subject) |>
  mutate(rt_lexdec = exp(rt_lexdec)) |>
  summary()
```

```
   rt_lexdec      age_subject
 Min.   : 495.4   old  :2284
 1st Qu.: 617.4   young:2284
 Median : 699.6
 Mean   : 708.1
 3rd Qu.: 775.3
 Max.   :1323.2
```

- don't see the intercept value there

Summarisinggroup effects

- our intercept was 786.72, but that wasn't the grand mean reaction time
  - what is the intercept?
- how does `rt_lexdec` look for the two groups?

```r
df_freq_eng |>
  select(rt_lexdec, age_subject) |>
  mutate(rt_lexdec = exp(rt_lexdec)) |>
  summarise(mean = mean(rt_lexdec),
            min = min(rt_lexdec),
            max = max(rt_lexdec),
    .by = "age_subject"
  )
```

```
  age_subject     mean    min    max
1       young 629.5473 495.38  971.8
2         old 786.7200 603.77 1323.2
```

- the intercept corresponds to the mean reaction time for the old group. Why?
  - because `old` coded as 0

Intercept at 0

- the intercept corresponds to the value of y when x is 0
  - when predictors are *centered*, this will correspond to the mean value of y, because when $x = 0$ it aligns with the centre value of y
  - when predictors are not centered, this will correspond to the value of y when x is 0 in the original unit of measurement

Default contrasts

- which variable is coded as 0?
  - R simply takes the first level name alphabetically: `old` comes before `young`, so `old` was automatically taken as the 'baseline' to which `young` was compared
- if we were to add the slope to the intercept, we would get the mean for the *young* group. Why is this?

```r
coef(fit_age)['(Intercept)'] + coef(fit_age)['age_subjectyoung']
```

```
(Intercept) 
   629.5473 
```

Simple linear regression as a two-sample t-test

- this actually is the same thing as a *t*-test:

```
t.test(exp(rt_lexdec) ~ age_subject, data = df_freq_eng)
```

```

    Welch Two Sample t-test

data:  exp(rt_lexdec) by age_subject
t = 63.406, df = 4144.6, p-value < 0.00000000000000022
alternative hypothesis: true difference in means between group old and group young is not equ
95 percent confidence interval:
 152.3128 162.0325
sample estimates:
  mean in group old mean in group young
           786.7200            629.5473
```

- if we compare this to our model, we see that the *t*- and *p*-values are identical (more on these later).

```
tidy(fit_age)
```

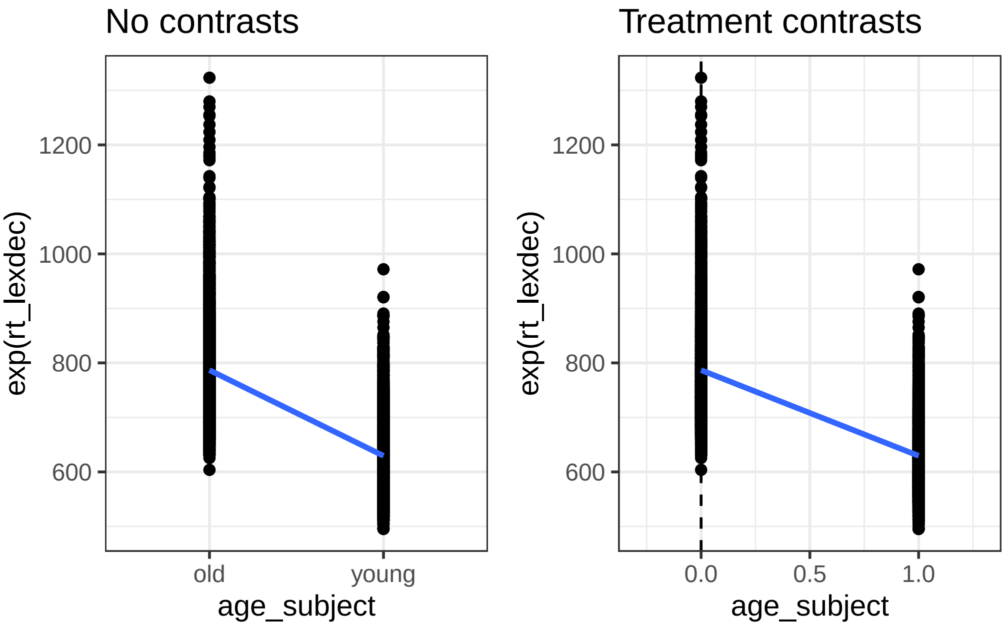
```
# A tibble: 2 x 5
  term             estimate std.error statistic p.value
  <chr>               <dbl>     <dbl>     <dbl>   <dbl>
1 (Intercept)          787.      1.75     449.        0
2 age_subjectyoung    -157.      2.48     -63.4       0
```

Visualing treatment contrasts

```
fig_nocontrasts <-
df_freq_eng |>
  ggplot() +
  aes(x = age_subject, y = exp(rt_lexdec)) +
  labs(title = "No contrasts") +
  # geom_vline(xintercept = 0, linetype="dashed", size = .5) +
  geom_point(position = position_dodge(.6)) +
  geom_smooth(method = 'lm', aes(group=1)) + theme_minimal() +
  theme_bw()
```

```r
fig_treatment <-
df_freq_eng |>
  mutate(age_subject = if_else(age_subject=="young",1,0)) |>
  ggplot() +
  aes(x = age_subject, y = exp(rt_lexdec)) +
  labs(title = "Treatment contrasts") +
  geom_vline(xintercept = 0, linetype="dashed", size = .5) +
  geom_point(position = position_dodge(.6)) +
  geom_smooth(method = 'lm', aes(group=1)) + theme_minimal() +
  theme_bw()

fig_nocontrasts + fig_treatment
```



Sum contrasts/coding

- sum coding is another frequently used coding scheme
  - essentially centring categorical variables
- simplifies interpretation of interaction effects
- instead of 0 and 1, we set our contrasts to +/-1 or 0.5 (I prefer 0.5)

Setting sum contrasts

- ensure we're working with a factor

```
# first, make sure your variable is a factor
df_freq_eng$age_subject <- as.factor(df_freq_eng$age_subject)
```

- check it is a factor (could do this first)

```
# check
class(df_freq_eng$age_subject)
```

```
[1] "factor"
```

`contr.sum()`

- we can use the `contr.sum()` function to set sum contrasts
  - takes as its argument the number of factor levels

```
# next, you could use the contr.sum() function
contrasts(df_freq_eng$age_subject) <- contr.sum(2) # where 2 means we have 2 levels
contrasts(df_freq_eng$age_subject)
```

```
      [,1]
old      1
young   -1
```

- `old` is coded as -1 and `young` as $+1$
- I prefer to use +/-0.5 for reasons we don't need to go into here
  - I would also prefer to have `young` coded in the negative value, and `old` in the positive value
  - this aids in the way I interpret the slope: a change in reaction times for the older group compared to the younger group

By-hand

```r
#or, you could manually control the sum contrasts
# check the order of the levels
levels(df_freq_eng$age_subject)
```

```
[1] "old"   "young"
```

```r
# code 'old' as +.5 and 'young' as -.5
contrasts(df_freq_eng$age_subject) <- c(+0.5, -0.5)
contrasts(df_freq_eng$age_subject)
```

```
      [,1]
old    0.5
young -0.5
```

Model with sum coded factor

- run our model

```r
fit_age_sum <-
  lm(exp(rt_lexdec) ~ age_subject,
     data = df_freq_eng)
```

```r
glance(fit_age_sum)$r.squared
```

```
[1] 0.4682224
```

```r
glance(fit_age)$r.squared
```

```
[1] 0.4682224
```

- no difference in variance account for by our model (remember, centering a variable just shifts values, doesn't affect the relationship between values)

Coefficients

```
tidy(fit_age_sum) |> select(term,estimate)
```

```
# A tibble: 2 x 2
  term         estimate
  <chr>           <dbl>
1 (Intercept)      708.
2 age_subject1     157.
```

- there is a difference in the intercept
    - and a change in sign in our slope. Why is this?

Treatment/Dummy vs. Sum contrasts

```
fig_sum1 <-
df_freq_eng |>
  mutate(age_subject = if_else(age_subject=="young",-1,1)) |>
  ggplot() +
  aes(x = age_subject, y = exp(rt_lexdec)) +
  labs(title = "Sum contrasts") +
  geom_vline(xintercept = 0, linetype="dashed", size = .5) +
  geom_point(position = position_dodge(.6)) +
  geom_smooth(method = 'lm', aes(group=1)) + theme_minimal() +
  theme_bw()

fig_sum5 <-
df_freq_eng |>
  mutate(age_subject = if_else(age_subject=="young",-.5,.5)) |>
  ggplot() +
  aes(x = age_subject, y = exp(rt_lexdec)) +
  labs(title = "Sum contrasts") +
  geom_vline(xintercept = 0, linetype="dashed", size = .5) +
  geom_point(position = position_dodge(.6)) +
  geom_smooth(method = 'lm', aes(group=1)) + theme_minimal() +
  theme_bw()

fig_treatment + fig_sum5 + plot_annotation(tag_levels = "A")
```

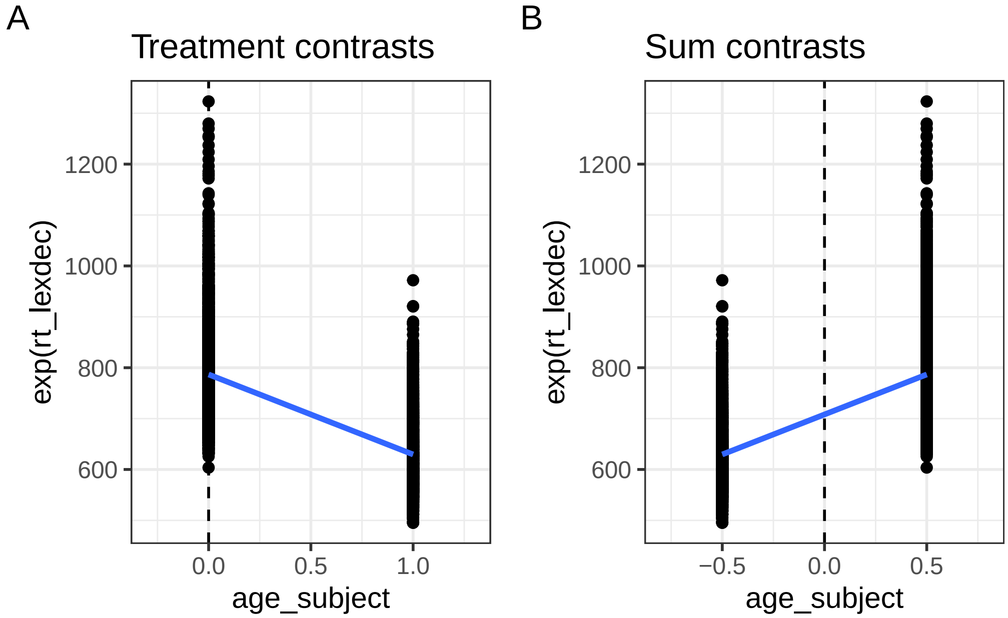


Figure 1: The difference in slope corresponds to which level is coded as 0 (dummy coding) or -5/-1 (sum coding)

Intercept

- the intercept value is now the overall mean of all observed reaction times, because now the y value when x equals zero lies in the middle of the two groups
- the slope magnitude (i.e., size of the value) hasn't changed, because the difference betwen the two group means has not changed

```
mean(exp(df_freq_eng$rt_lexdec))
```

```
[1] 708.1336
```

Exploring predicted values

- let's explore the predicted values of our model with a categorical variable

```
fitted(fit_age)[1:6]
```

```
     338     1790     3125     3957     3313     4145 
629.5473 786.7200 629.5473 786.7200 629.5473 786.7200 
```

- there are only 2 values, 630 and 787
  - these correspond to the means for each group that we saw above
  - they also seem to be in a pattern: mean(young), mean(old), mean(young), mean(old), etc.
  - how does this correspond to the age group of the participant for the first ten observations?

```r
df_freq_eng$age_subject[1:6]
```

```
[1] young old   young old   young old  
attr(,"contrasts")
      [,1]
old    0.5
young -0.5
Levels: old young
```

- first ten observations in our data are in young-old pairs. What are the first values in the raw data?

```r
exp(df_freq_eng$rt_lexdec[1:6])
```

```
[1] 623.61 775.67 617.10 715.52 575.70 742.19
```

- what is the difference between these reaction times and the fitted values?

```r
exp(df_freq_eng$rt_lexdec[1:6]) - fitted(fit_age)[1:6]
```

```
       338       1790       3125       3957       3313       4145 
 -5.937299 -11.049991 -12.447299 -71.199991 -53.847299 -44.529991 
```

```r
residuals(fit_age)[1:6]
```

```
       338       1790       3125       3957       3313       4145 
 -5.937299 -11.049991 -12.447299 -71.199991 -53.847299 -44.529991 
```

---

term	description/other terms

- we see again that predicted values correspond to the x value for the corresponding row in the dataframe
  - but with our two-level factor, we only have two x values, young and old

---

```
df_freq_eng <-
  augment(fit_age, df_freq_eng)
```

```
df_freq_eng |>
  select(word, age_subject, rt_lexdec, .fitted, .resid) |>
  mutate(rt_lexdec = exp(rt_lexdec)) |>
  head()
```

```
# A tibble: 6 x 5
  word  age_subject rt_lexdec .fitted .resid
  <fct> <fct>           <dbl>   <dbl>  <dbl>
1 ace   young            624.    630.  -5.94
2 ace   old              776.    787. -11.0
3 act   young            617.    630. -12.4
4 act   old              716.    787. -71.2
5 add   young            576.    630. -53.8
6 add   old              742.    787. -44.5
```

Summary

- we saw that the equation for a straight line boils down to its intercept and slope
- we fit our first linear model with a categorical predictor

Important terms

Learning Objectives

Today we learned…

- about cateogorical predictors
- how to interpret different contrast coding

Task

Follow the instructions on the website (Multiple regression > Task) (or continue to the next slides).

Reading time data

We'll use a dataset from Biondo et al. (2022), an eye-tracking reading study exploring the processing of adverb-tense concord in Spanish past and future tenses. Participants read sentences that began with a temporal adverb (e.g., yesterday/tomorrow), and had a verb marked with the congruent or incongruent tense (past/future).

Load in the data.

```r
df_tense <-
  read_csv(here("data", "Biondo.Soilemezidi.Mancini_dataset_ET.csv"),
           locale = locale(encoding = "Latin1") # for special characters in Spanish
           ) |>
  mutate(gramm = ifelse(gramm == "0", "ungramm", "gramm")) |>
  clean_names()
```

Treatment contrasts

We will look at the measure *total reading time* (`tt`) at the *verb* region (`roi == 4`). Subset the data to only include the verb region.

```r
df_verb <-
  df_tense |>
  filter(roi == 4)
```

1. Run a simple linear model with (log-transformed) total reading time (`tt`) as an independent variable and grammaticality (`gramm`) as a dependent variable. Use treatment contrasts.

2. Inspect your coefficients again. What conclusions do you draw?

3. Run model diagnostics:

- check model assumptions where relevant (normality, constant variance, collinearity)
- check model fit (R^2)

Sum contrasts

1. Re-run your model with sum contrasts.
2. Inspect your coefficients again. Do your conclusions change?
3. Re-run your model diagnostics. How does it compare to your first model?

Multiple regression

1. Add verb tense (`verb_t`: past, future) as a predictor, including an interaction term. Use sum contrasts.
2. Inspect your coefficients again. Do your conclusions change?
3. Re-run your model diagnostics. How does it compare to the last models?

Literaturverzeichnis

Biondo, N., Soilemezidi, M., & Mancini, S. (2022). Yesterday is history, tomorrow is a mystery: An eye-tracking investigation of the processing of past and future time reference during sentence reading. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *48*(7), 1001–1018. https://doi.org/10.1037/xlm0001053