

additional prefixes

Zachary Houghton

2025-12-24

Added a bunch of prefixes → single model with prefix as RE?

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.2.3
## Warning: package 'ggplot2' was built under R version 4.2.3
## Warning: package 'tibble' was built under R version 4.2.3
## Warning: package 'tidyr' was built under R version 4.2.3
## Warning: package 'readr' was built under R version 4.2.3
## Warning: package 'purrr' was built under R version 4.2.3
## Warning: package 'dplyr' was built under R version 4.2.3
## Warning: package 'stringr' was built under R version 4.2.3
## Warning: package 'lubridate' was built under R version 4.2.3

## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.2      v readr      2.1.4
## v forcats    1.0.0      v stringr   1.5.1
## v ggplot2    3.5.1      v tibble    3.2.1
## v lubridate  1.9.3      v tidyr     1.3.0
## v purrr      1.0.2

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(brms)
```

```
## Warning: package 'brms' was built under R version 4.2.3

## Loading required package: Rcpp
## Loading 'brms' package (version 2.20.4). Useful instructions
## can be found by typing help('brms'). A more detailed introduction
## to the package is available through vignette('brms_overview').
##
## Attaching package: 'brms'
##
## The following object is masked from 'package:stats':
##
## ar
```

```

corpus = read_csv('../Data/nonce_and_attested_binoms.csv')

## Rows: 725 Columns: 21
## -- Column specification -----
## Delimiter: ","
## chr (5): Word1, Word2, Alpha, Nonalpha, Weird?
## dbl (16): OverallFreq, Word1_freq, Word2_freq, Form, Percept, Culture, Power...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
data_all_prefixes_all_models_attested_novel = read_csv('../Data/ALL_MODELS_ALL_PREFIXES_NOVEL_AND_ATTES')
mutate(log_odds = preference) %>%
separate(binom, c('Word1', 'and', 'Word2'), remove = F, sep = ' ') %>%
select(-and) %>%
#mutate(across(2:3, tolower)) %>%
left_join(corpus) %>%
mutate(checkpoint = 'main') %>%
mutate(y_vals = 0.02191943 + 0.23925834*Form + 0.24889543*Percept + 0.41836997*Culture + 0.259673
mutate(GenPref = 1/(1+exp(-1*y_vals))) %>%
mutate(GenPref = GenPref - 0.5)

## Rows: 253750 Columns: 9
## -- Column specification -----
## Delimiter: ","
## chr (5): WordA, WordB, binom, model, prompt_text
## dbl (4): alpha_logprob, nonalpha_logprob, prompt_value, preference
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Joining with `by = join_by(Word1, Word2)`
data_all_prefixes_all_models = read_csv('../Data/ALL_MODELS_ALL_PREFIXES_NOVEL_AND_ATTENDED.csv') %>%
mutate(log_odds = preference) %>%
separate(binom, c('Word1', 'and', 'Word2'), remove = F, sep = ' ') %>%
select(-and) %>%
#mutate(across(2:3, tolower)) %>%
left_join(corpus) %>%
mutate(checkpoint = 'main') %>%
mutate(y_vals = 0.02191943 + 0.23925834*Form + 0.24889543*Percept + 0.41836997*Culture + 0.259673
mutate(GenPref = 1/(1+exp(-1*y_vals))) %>%
mutate(GenPref = GenPref - 0.5) %>%
filter(Attested == 0)

## Rows: 253750 Columns: 9
## -- Column specification -----
## Delimiter: ","
## chr (5): WordA, WordB, binom, model, prompt_text
## dbl (4): alpha_logprob, nonalpha_logprob, prompt_value, preference
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Joining with `by = join_by(Word1, Word2)`
bigram_counts = read_csv('../Data/olmo_bigram_freqs.csv')

```

```

## Rows: 308 Columns: 2
## -- Column specification -----
## Delimiter: ","
## chr (1): ngram
## dbl (1): count
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
onegram_corpus_size = 1560674367427
count_and = 43378012800

bigram_counts = bigram_counts %>%
  mutate(ngram = str_replace_all(ngram, '\\t', ' '))

data_all_prefixes_all_models = data_all_prefixes_all_models %>%
  rename(no_final_stress = `*BStress`)

data_all_prefixes_all_models = data_all_prefixes_all_models %>%
  mutate(bigram1_alpha = paste0(Word1, ' and'),
         bigram2_alpha = paste0('and ', Word2),
         bigram1_nonalpha = paste0(Word2, ' and'),
         bigram2_nonalpha = paste0('and ', Word1)
  )

data_all_prefixes_all_models = data_all_prefixes_all_models %>%
  left_join(bigram_counts, by = c('bigram1_alpha' = 'ngram')) %>%
  rename(count_bigram1_alpha = count) %>%
  left_join(bigram_counts, by = c('bigram2_alpha' = 'ngram')) %>%
  rename(count_bigram2_alpha = count) %>%
  left_join(bigram_counts, by = c('bigram1_nonalpha' = 'ngram')) %>%
  rename(count_bigram1_nonalpha = count) %>%
  left_join(bigram_counts, by = c('bigram2_nonalpha' = 'ngram')) %>%
  rename(count_bigram2_nonalpha = count) %>%
  mutate(corpus_size = onegram_corpus_size) %>%
  mutate(count_and = count_and)

data_all_prefixes_all_models = data_all_prefixes_all_models %>%
  mutate(bigram_prob_alpha = (Word1_freq / corpus_size) * (count_bigram1_alpha / Word1_freq) * (count_bigram2_alpha / Word2_freq),
         bigram_prob_nonalpha = (Word2_freq / corpus_size) * (count_bigram1_nonalpha / Word2_freq) * (count_bigram2_nonalpha / Word1_freq),
         log_bigram_prob_alpha = log(bigram_prob_alpha),
         log_bigram_prob_nonalpha = log(bigram_prob_nonalpha),
         log_bigram_odds_ratio = log_bigram_prob_alpha - log_bigram_prob_nonalpha)

```

Model

```

options(contrasts = c("contr.sum", "contr.sum"))

prior_probs = c(
  prior(student_t(3, 0, 5), class = 'Intercept'),

```

```

prior(student_t(3, 0, 5), class = 'sigma'),
prior(student_t(3, 0, 5), class = 'b')
)

data_all_prefixes_all_models$prompt_value = factor(data_all_prefixes_all_models$prompt_value)

all_models_all_prefixes = brm(log_odds ~ GenPref * model + (GenPref * model | prompt_value) + ( 1 | Alp
  data = data_all_prefixes_all_models,
  prior = prior_probs,
  iter = 20000,
  warmup = 10000,
  chains = 4,
  cores = 4,
  #control = list(adapt_delta=0.99, max_treedepth = 15),
  #control = list(max_treedepth = 20),
  file = '../Data/all_models_all_prefixes'
)

#
# data_all_prefixes_all_models_attested_novel_and_attested = data_all_prefixes_all_models_attested_nove
# mutate(RelFreq = case_when(
#   Attested == 1 ~ RelFreq,
#   Attested == 0 ~ 0),
#   OverallFreq = case_when(
#     Attested == 1 ~ OverallFreq,
#     Attested == 0 ~ 0
#   ))
# data_all_prefixes_all_models_attested_novel$Attested = factor(data_all_prefixes_all_models_attested_n

all_models_all_prefixes_bigrams = brm(log_odds ~ GenPref * model * log_bigram_odds_ratio + (GenPref * m
  data = data_all_prefixes_all_models,
  prior = prior_probs,
  iter = 20000,
  warmup = 10000,
  chains = 4,
  cores = 4,
  control = list(adapt_delta=0.99, max_treedepth = 15),
  #control = list(max_treedepth = 20),
  file = '../Data/all_models_all_prefixes_bigrams'
)

fixef(all_models_all_prefixes)

```

##	Estimate	Est.Error	Q2.5	Q97.5
## Intercept	-0.03404857	0.08898473	-0.2094651800	0.140733697
## GenPref	1.88671411	0.63970335	0.6464901123	3.136547806
## model1	0.10431425	0.02983103	0.0457547096	0.162586700
## model2	0.02869317	0.03039814	-0.0309879610	0.088005029
## model3	-0.01471215	0.02667745	-0.0669249244	0.038046865
## model4	-0.06738395	0.02697718	-0.1203912100	-0.014188458
## model5	-0.14161780	0.02985411	-0.2008401980	-0.083208601
## model6	0.05376846	0.02712218	0.0005760425	0.107155329

```
## GenPref:model11 -0.42021847 0.21520056 -0.8407431988 0.006262447
## GenPref:model12 -0.92290309 0.24425708 -1.4011194842 -0.439587546
## GenPref:model13 -2.71985309 0.23029335 -3.1723118174 -2.262152756
## GenPref:model14 0.50790697 0.26321584 -0.0055496869 1.023527792
## GenPref:model15 0.98309369 0.20025124 0.5898872300 1.376228227
## GenPref:model16 1.70836712 0.20354497 1.3106880613 2.108634766
```

```
fixef(all_models_all_prefixes_bigrams)
```

```
## Estimate Est.Error Q2.5
## Intercept -0.010699103 0.08898188 -0.18443250
## GenPref 1.623864428 0.65959055 0.33377611
## model11 0.117718845 0.02997732 0.05865393
## model12 0.040422941 0.03060349 -0.01930349
## model13 -0.034034854 0.02677150 -0.08606555
## model14 -0.062796305 0.02735915 -0.11682350
## model15 -0.149771828 0.02996937 -0.20849726
## model16 0.038381505 0.02719196 -0.01443979
## log_bigram_odds_ratio 0.091423309 0.14099201 -0.18509377
## GenPref:model11 -0.669851328 0.21987204 -1.09880895
## GenPref:model12 -1.118557265 0.24997523 -1.61097109
## GenPref:model13 -2.551130695 0.23427046 -3.00968067
## GenPref:model14 0.577129289 0.26735432 0.05569173
## GenPref:model15 1.156409326 0.20478305 0.75525454
## GenPref:model16 1.784267866 0.20745762 1.37555714
## GenPref:log_bigram_odds_ratio -2.104254201 1.29119571 -4.67371255
## model11:log_bigram_odds_ratio 0.003218036 0.04258608 -0.07992529
## model12:log_bigram_odds_ratio 0.026802788 0.04240667 -0.05627072
## model13:log_bigram_odds_ratio -0.187890404 0.04184970 -0.26909335
## model14:log_bigram_odds_ratio 0.147314485 0.04214684 0.06498400
## model15:log_bigram_odds_ratio 0.015911254 0.04256464 -0.06807474
## model16:log_bigram_odds_ratio -0.197456094 0.04229170 -0.28024679
## GenPref:model11:log_bigram_odds_ratio -2.162772491 0.39422502 -2.93060812
## GenPref:model12:log_bigram_odds_ratio -1.634397143 0.39456724 -2.41204497
## GenPref:model13:log_bigram_odds_ratio 0.850400895 0.39346117 0.08666489
## GenPref:model14:log_bigram_odds_ratio 1.094688501 0.39488358 0.31842976
## GenPref:model15:log_bigram_odds_ratio 1.566251563 0.39440564 0.79669164
## GenPref:model16:log_bigram_odds_ratio 0.046968322 0.39397816 -0.72882167
## Q97.5
## Intercept 0.169309703
## GenPref 2.924128315
## model11 0.176458802
## model12 0.100987551
## model13 0.018537881
## model14 -0.009519236
## model15 -0.090614355
## model16 0.092028618
## log_bigram_odds_ratio 0.371779394
## GenPref:model11 -0.237632133
## GenPref:model12 -0.623318369
## GenPref:model13 -2.087019202
## GenPref:model14 1.103335512
## GenPref:model15 1.557475931
## GenPref:model16 2.192759315
## GenPref:log_bigram_odds_ratio 0.418023867
```

```

## model1:log_bigram_odds_ratio      0.086480059
## model2:log_bigram_odds_ratio      0.109704904
## model3:log_bigram_odds_ratio     -0.106383466
## model4:log_bigram_odds_ratio      0.230071195
## model5:log_bigram_odds_ratio      0.099980528
## model6:log_bigram_odds_ratio     -0.114729861
## GenPref:model1:log_bigram_odds_ratio -1.391769716
## GenPref:model2:log_bigram_odds_ratio -0.854314282
## GenPref:model3:log_bigram_odds_ratio  1.620264367
## GenPref:model4:log_bigram_odds_ratio  1.870911209
## GenPref:model5:log_bigram_odds_ratio  2.341526430
## GenPref:model6:log_bigram_odds_ratio  0.813248851

data_all_prefixes_gpt2xl = data_all_prefixes_all_models %>%
  filter(model == 'gpt2xl')

data_all_prefixes_olmo2_1b = data_all_prefixes_all_models %>%
  filter(model == 'olmo2_1b')

data_all_prefixes_olmo7b = data_all_prefixes_all_models %>%
  filter(model == 'olmo7b')

data_all_prefixes_gpt2 = data_all_prefixes_all_models %>%
  filter(model == 'gpt2')

data_all_prefixes_gptoss = data_all_prefixes_all_models %>%
  filter(model == 'gptoss120b')

data_all_prefixes_olmo3_32b = data_all_prefixes_all_models %>%
  filter(model == 'olmo3_32b')

data_all_prefixes_olmo3_7b = data_all_prefixes_all_models %>%
  filter(model == 'olmo3_7b')

all_models_gpt2xl = brm(log_odds ~ GenPref + (GenPref | prompt_value) + ( 1 | Alpha),
  data = data_all_prefixes_gpt2xl,
  prior = prior_probs,
  iter = 20000,
  warmup = 10000,
  chains = 4,
  cores = 4,
  control = list(adapt_delta=0.99, max_treedepth = 15),
  #control = list(max_treedepth = 20),
  file = '../Data/all_models_gpt2xl_all_prefixes'
)

all_models_olmo2_1b = brm(log_odds ~ GenPref + (GenPref | prompt_value) + ( 1 | Alpha),
  data = data_all_prefixes_olmo2_1b,
  prior = prior_probs,
  iter = 22000,
  warmup = 11000,
  chains = 4,
  cores = 4,

```

```

        control = list(adapt_delta=0.99, max_treedepth = 17),
        #control = list(max_treedepth = 20),
        file = '../Data/all_models_olmo2_1b_all_prefixes'
    )

prior_probs = c(
    prior(student_t(3, 0, 0.25), class = 'Intercept'),
    prior(student_t(3, 0, 0.25), class = 'sigma'),
    prior(student_t(3, 0, 0.25), class = 'b')
)

all_models_olmo7b = brm(log_odds ~ GenPref + (GenPref | prompt_value) + ( 1 | Alpha),
    data = data_all_prefixes_olmo7b,
    prior = prior_probs,
    iter = 30000,
    warmup = 15000,
    chains = 4,
    cores = 4,
    control = list(adapt_delta=0.999, max_treedepth = 20),
    #control = list(max_treedepth = 20),
    file = '../Data/all_models_olmo7b_all_prefixes'
)

all_models_gpt2 = brm(log_odds ~ GenPref + (GenPref | prompt_value) + ( 1 | Alpha),
    data = data_all_prefixes_gpt2,
    prior = prior_probs,
    iter = 30000,
    warmup = 15000,
    chains = 4,
    cores = 4,
    control = list(adapt_delta=0.999, max_treedepth = 20),
    #control = list(max_treedepth = 20),
    file = '../Data/all_models_gpt2'
)

all_models_gptoss = brm(log_odds ~ GenPref + (GenPref | prompt_value) + ( 1 | Alpha),
    data = data_all_prefixes_gptoss,
    prior = prior_probs,
    iter = 30000,
    warmup = 15000,
    chains = 4,
    cores = 4,
    control = list(adapt_delta=0.999, max_treedepth = 20),
    #control = list(max_treedepth = 20),
    file = '../Data/all_models_gptoss'
)

all_models_olmo3_32b = brm(log_odds ~ GenPref + (GenPref | prompt_value) + ( 1 | Alpha),
    data = data_all_prefixes_olmo3_32b,
    prior = prior_probs,
    iter = 30000,
    warmup = 15000,

```

```

        chains = 4,
        cores = 4,
        control = list(adapt_delta=0.999, max_treedepth = 20),
        #control = list(max_treedepth = 20),
        file = '../Data/all_models_olmo3_32b'
    )

all_models_olmo3_7b = brm(log_odds ~ GenPref + (GenPref | prompt_value) + ( 1 | Alpha),
    data = data_all_prefixes_olmo3_7b,
    prior = prior_probs,
    iter = 30000,
    warmup = 15000,
    chains = 4,
    cores = 4,
    control = list(adapt_delta=0.999, max_treedepth = 20),
    #control = list(max_treedepth = 20),
    file = '../Data/all_models_olmo3_7b'
)

```

```
fixef(all_models_all_prefixes)
```

##	Estimate	Est.Error	Q2.5	Q97.5
## Intercept	-0.03404857	0.08898473	-0.2094651800	0.140733697
## GenPref	1.88671411	0.63970335	0.6464901123	3.136547806
## model1	0.10431425	0.02983103	0.0457547096	0.162586700
## model2	0.02869317	0.03039814	-0.0309879610	0.088005029
## model3	-0.01471215	0.02667745	-0.0669249244	0.038046865
## model4	-0.06738395	0.02697718	-0.1203912100	-0.014188458
## model5	-0.14161780	0.02985411	-0.2008401980	-0.083208601
## model6	0.05376846	0.02712218	0.0005760425	0.107155329
## GenPref:model1	-0.42021847	0.21520056	-0.8407431988	0.006262447
## GenPref:model2	-0.92290309	0.24425708	-1.4011194842	-0.439587546
## GenPref:model3	-2.71985309	0.23029335	-3.1723118174	-2.262152756
## GenPref:model4	0.50790697	0.26321584	-0.0055496869	1.023527792
## GenPref:model5	0.98309369	0.20025124	0.5898872300	1.376228227
## GenPref:model6	1.70836712	0.20354497	1.3106880613	2.108634766

```
fixef(all_models_gpt2xl)
```

##	Estimate	Est.Error	Q2.5	Q97.5
## Intercept	-0.005311742	0.1733886	-0.3452397	0.3365391
## GenPref	0.964725337	1.2592101	-1.4878264	3.4255640

```
fixef(all_models_olmo2_1b)
```

##	Estimate	Est.Error	Q2.5	Q97.5
## Intercept	-0.1005371	0.1668688	-0.4278578	0.2272569
## GenPref	2.2652945	1.2122825	-0.1725241	4.6127364

```
fixef(all_models_olmo7b)
```

##	Estimate	Est.Error	Q2.5	Q97.5
## Intercept	-0.02229001	0.1029021	-0.2270842	0.176839
## GenPref	0.89297824	0.8613566	-0.1711859	2.962225


```
fixef(all_models_gpt2)
```

```
##           Estimate Est.Error      Q2.5      Q97.5
## Intercept 0.04024677 0.1363444 -0.2220176 0.3109478
## GenPref   0.13302352 0.3863472 -0.5010176 1.0411523
```

```
fixef(all_models_gptoss)
```

```
##           Estimate Est.Error      Q2.5      Q97.5
## Intercept -0.02867357 0.08764891 -0.1995778 0.1428132
## GenPref   -0.13702807 0.32646313 -0.8926902 0.4248313
```

```
fixef(all_models_olmo3_32b)
```

```
##           Estimate Est.Error      Q2.5      Q97.5
## Intercept -0.1667890 0.1301792 -0.4294957 0.0804448
## GenPref    0.4673573 0.6965435 -0.3319806 2.4868437
```

```
fixef(all_models_olmo3_7b)
```

```
##           Estimate Est.Error      Q2.5      Q97.5
## Intercept -0.01821676 0.1315097 -0.2815057 0.2376105
## GenPref    0.68261890 0.9403545 -0.2951167 3.3926109
```

random effects plots

```
library(dplyr)
library(tidyr)
library(ggplot2)
```

```
# 1. reshape wide (all terms and all metrics)
```

```
re_df = raneef(all_models_all_prefixes)$prompt_value
```

```
re_df <- as.data.frame.table(re_df, responseName = "value")
names(re_df) <- c("prompt_value", "metric", "term", "value")
```

```
re_wide <- re_df %>%
  pivot_wider(
    names_from = c(term, metric),
    values_from = value
  ) %>%
  mutate(
    model3_Estimate = -(model1_Estimate + model2_Estimate),
    model3_Q2.5     = -(model1_Q97.5 + model2_Q97.5),
    model3_Q97.5    = -(model1_Q2.5 + model2_Q2.5)
  )
```

```
# 2. pivot long again so ggplot can facet
```

```
re_long <- re_wide %>%
  pivot_longer(
    cols = -prompt_value,
    names_to = c("term", "metric"),
    names_sep = "_",
    values_to = "value"
  )
```

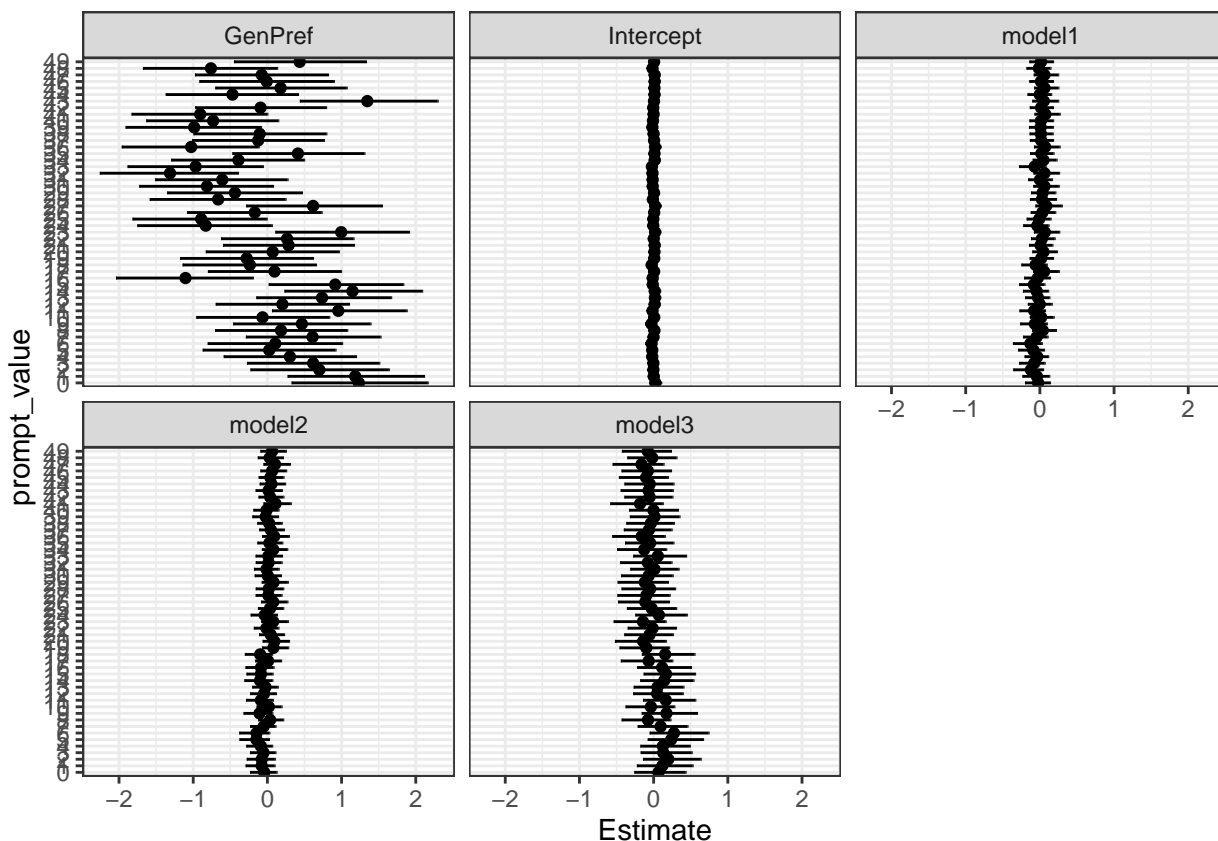
```

# 3. only keep useful terms
plot_df <- re_long %>%
  filter(term %in% c("Intercept", "GenPref", "model1", "model2", "model3"))

# 4. create wide data for ribbons
plot_df_wide <- plot_df %>%
  pivot_wider(names_from = metric, values_from = value)

# 5. plot EVERYTHING properly
ggplot(plot_df_wide, aes(x = Estimate, y = prompt_value)) +
  geom_point() +
  geom_errorbarh(aes(xmin = Q2.5, xmax = Q97.5), height = 0) +
  facet_wrap(~ term, scales = "fixed") +
  theme_bw()

```



```

# 1. reshape wide (all terms and all metrics)
re_df = ranef(all_models_all_prefixes_bigrams)$prompt_value

re_df <- as.data.frame.table(re_df, responseName = "value")
names(re_df) <- c("prompt_value", "metric", "term", "value")

re_wide <- re_df %>%
  pivot_wider(
    names_from = c(term, metric),
    values_from = value
  )

```

```

) %>%
mutate(
  model3_Estimate = -(model1_Estimate + model2_Estimate),
  model3_Q2.5     = -(model1_Q97.5 + model2_Q97.5),
  model3_Q97.5    = -(model1_Q2.5 + model2_Q2.5)
)

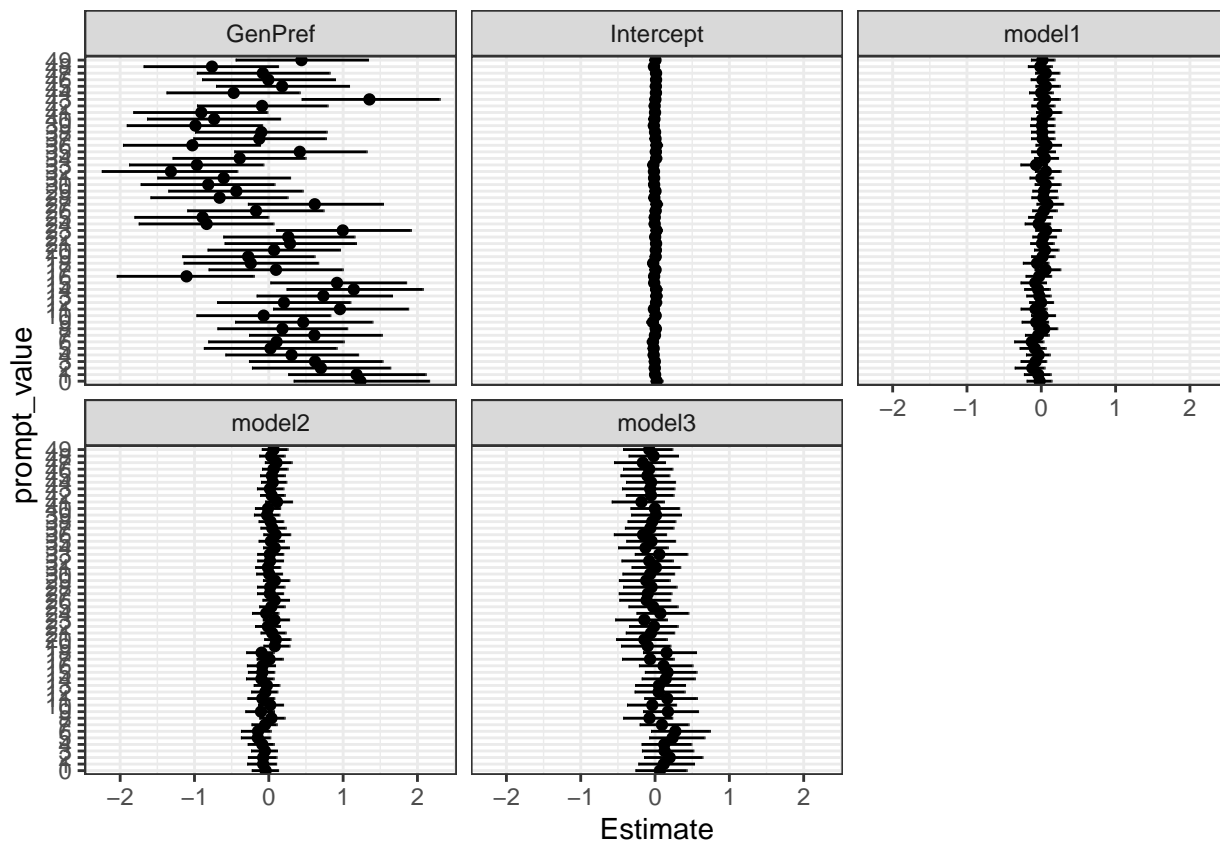
# 2. pivot long again so ggplot can facet
re_long <- re_wide %>%
  pivot_longer(
    cols = -prompt_value,
    names_to = c("term", "metric"),
    names_sep = "_",
    values_to = "value"
  )

# 3. only keep useful terms
plot_df <- re_long %>%
  filter(term %in% c("Intercept", "GenPref", "model1", "model2", "model3"))

# 4. create wide data for ribbons
plot_df_wide <- plot_df %>%
  pivot_wider(names_from = metric, values_from = value)

# 5. plot EVERYTHING properly
ggplot(plot_df_wide, aes(x = Estimate, y = prompt_value)) +
  geom_point() +
  geom_errorbarh(aes(xmin = Q2.5, xmax = Q97.5), height = 0) +
  facet_wrap(~ term, scales = "fixed") +
  theme_bw()

```



individual constraints

```
all_models_all_prefixes_all_constraints = brm(log_odds ~ (Culture + Power + Freq + Len + no_final_stress),
  data = data_all_prefixes_all_models,
  prior = prior_probs,
  iter = 6000,
  warmup = 3000,
  chains = 4,
  cores = 4,
  #control = list(adapt_delta=0.99, max_treedepth = 15),
  #control = list(max_treedepth = 20),
  file = '../Data/all_models_all_prefixes_all_constraints'
)
```

```
library(car)
```

```
## Warning: package 'car' was built under R version 4.2.3
```

```
## Loading required package: carData
```

```
##
```

```
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
```

```
##
```

```
## recode
```

```
## The following object is masked from 'package:purrr':
```

```
##
```

```
##      some
lm_vif <- lm(
  log_odds ~
    (Culture + Power + Freq + Len + no_final_stress +
     Intense + Percept + Lapse),
  data = data_all_prefixes_all_models
)

car::vif(lm_vif)

##      Culture      Power      Freq      Len no_final_stress
##      1.431293      1.407791      1.407957      1.346660      3.924262
##      Intense      Percept      Lapse
##      1.341107      1.442133      4.798560

#drop lapse and no_final_stress
all_models_all_prefixes_all_constraints_vif_dropped = brm(log_odds ~ (Culture + Power + Freq + Len + In
  data = data_all_prefixes_all_models,
  prior = prior_probs,
  iter = 6000,
  warmup = 3000,
  chains = 4,
  cores = 4,
  #control = list(adapt_delta=0.99, max_treedepth = 15),
  #control = list(max_treedepth = 20),
  file = '../Data/all_models_all_prefixes_all_constraints_vif_dropped'
)

fixef(all_models_all_prefixes_all_constraints)

##      Estimate Est.Error      Q2.5      Q97.5
## Intercept      -0.034032183 0.07995685 -0.1934604545 0.122126268
## Culture      0.350355535 0.13644493 0.0895815302 0.624532846
## Power      0.257882499 0.13532014 -0.0036052267 0.529470057
## Freq      -0.032080893 0.04472789 -0.1197454280 0.054775474
## Len      -0.028613038 0.07210442 -0.1654204884 0.114696407
## no_final_stress      -0.129384617 0.15601271 -0.4436882525 0.175714239
## Intense      0.312819200 0.20339737 -0.0690734991 0.721874879
## Percept      0.138969744 0.11374242 -0.0823027186 0.364797342
## Lapse      -0.080183537 0.12445324 -0.3229109410 0.160835081
## model1      0.055159943 0.02627229 0.0043787754 0.106915315
## model2      -0.013107549 0.02632785 -0.0637671309 0.039413629
## model3      0.014355875 0.02595126 -0.0361436999 0.064934517
## model4      -0.037492150 0.02631633 -0.0890026774 0.014461059
## model5      -0.121661559 0.02645497 -0.1732064925 -0.069029444
## model6      0.052147335 0.02636013 0.0006882669 0.103254285
## Culture:model1      0.255854487 0.04305573 0.1720646249 0.340330230
## Culture:model2      0.048474649 0.04318093 -0.0356750764 0.133807535
## Culture:model3      -0.335719220 0.04340684 -0.4219920192 -0.250928021
## Culture:model4      -0.367389717 0.04375133 -0.4535355505 -0.282962820
## Culture:model5      0.180132973 0.04365929 0.0946864238 0.266472554
## Culture:model6      0.287435145 0.04305691 0.2028892490 0.372281453
```

```

## Power:model1 -0.601265408 0.04624828 -0.6909133122 -0.510999614
## Power:model2 -0.269430726 0.04523641 -0.3583348011 -0.181592471
## Power:model3 -0.316013840 0.04599826 -0.4069351329 -0.225821043
## Power:model4 0.498517050 0.04650850 0.4076235585 0.589067495
## Power:model5 0.192567809 0.04612259 0.1021123982 0.281578246
## Power:model6 0.437214223 0.04600246 0.3470762009 0.528180916
## Freq:model1 0.016770920 0.01389528 -0.0105656993 0.044024407
## Freq:model2 -0.013030354 0.01364270 -0.0396467636 0.013569895
## Freq:model3 -0.089698824 0.01372724 -0.1166297981 -0.062938145
## Freq:model4 0.045378816 0.01387927 0.0178815147 0.072311990
## Freq:model5 -0.022814353 0.01385403 -0.0498429667 0.004090255
## Freq:model6 -0.004294552 0.01359418 -0.0310542783 0.022323096
## Len:model1 0.091014002 0.02360011 0.0454642209 0.137524432
## Len:model2 0.103682158 0.02337747 0.0582834048 0.149443555
## Len:model3 0.209284683 0.02348043 0.1637636045 0.255385161
## Len:model4 -0.247539672 0.02371105 -0.2939472333 -0.201694424
## Len:model5 -0.209494798 0.02368269 -0.2555346169 -0.163646251
## Len:model6 -0.011240638 0.02367103 -0.0579331552 0.034762004
## no_final_stress:model1 -0.492125052 0.06587522 -0.6205809918 -0.361376105
## no_final_stress:model2 -0.517102452 0.06587227 -0.6446235415 -0.388559986
## no_final_stress:model3 0.011865713 0.06209674 -0.1105407387 0.134799630
## no_final_stress:model4 0.181055079 0.06403415 0.0554905967 0.305650064
## no_final_stress:model5 0.148021245 0.06414367 0.0253655403 0.273882481
## no_final_stress:model6 0.317577409 0.06524053 0.1890330120 0.447765199
## Intense:model1 -0.660443411 0.07264682 -0.8034432518 -0.516950148
## Intense:model2 0.326989439 0.07212206 0.1885118050 0.469006616
## Intense:model3 -0.360144568 0.07136006 -0.4979681245 -0.221939501
## Intense:model4 1.072921212 0.07270726 0.9299832692 1.214085044
## Intense:model5 -0.499446612 0.07251163 -0.6424791065 -0.357417428
## Intense:model6 0.350997920 0.07201268 0.2113191372 0.493580913
## Percept:model1 -0.050895543 0.03994110 -0.1287755281 0.027150659
## Percept:model2 -0.252656050 0.04052819 -0.3308575744 -0.173917100
## Percept:model3 -0.120105187 0.03979176 -0.1972268761 -0.042974356
## Percept:model4 0.189960794 0.04034159 0.1113540823 0.270388622
## Percept:model5 0.344377898 0.04033853 0.2654865413 0.423054511
## Percept:model6 -0.128431854 0.03958772 -0.2066270844 -0.050761137
## Lapse:model1 0.187641560 0.05246631 0.0838515049 0.287827805
## Lapse:model2 0.360601207 0.05209609 0.2597172484 0.462840795
## Lapse:model3 -0.019613644 0.04997323 -0.1184469539 0.078986178
## Lapse:model4 0.108303584 0.05151803 0.0104448050 0.210085167
## Lapse:model5 -0.241860740 0.05133207 -0.3431787799 -0.141699904
## Lapse:model6 -0.234223033 0.05186978 -0.3368973450 -0.132998319

```

```
## And now we do backward model selection
```

```

all_models_all_prefixes_individual_constraints = brm(log_odds ~ (Culture + Power + Freq + Len) * model +
  data = data_all_prefixes_all_models,
  prior = prior_probs,
  iter = 6000,
  warmup = 3000,
  chains = 4,
  cores = 4,
  #control = list(adapt_delta=0.99, max_treedepth = 15),
  #control = list(max_treedepth = 20),
  file = '../Data/all_models_all_prefixes_individual_constraints'

```

```

    )

gpt2xl_all_prefixes_individual_constraints = brm(log_odds ~ (Culture + Power + Freq + Len) + (Culture +
  data = data_all_prefixes_gpt2xl,
  prior = prior_probs,
  iter = 6000,
  warmup = 3000,
  chains = 4,
  cores = 4,
  #control = list(adapt_delta=0.99, max_treedepth = 15),
  #control = list(max_treedepth = 20),
  file = '../Data/gpt2xl_all_prefixes_individual_constraints'
)

olmo2_1b_all_prefixes_individual_constraints = brm(log_odds ~ (Culture + Power + Freq + Len) + (Culture +
  data = data_all_prefixes_olmo2_1b,
  prior = prior_probs,
  iter = 9000,
  warmup = 4500,
  chains = 4,
  cores = 4,
  #control = list(adapt_delta=0.99, max_treedepth = 15),
  #control = list(max_treedepth = 20),
  file = '../Data/olmo2_1b_all_prefixes_individual_constraints'
)

olmo7b_all_prefixes_individual_constraints = brm(log_odds ~ (Culture + Power + Freq + Len) + (Culture +
  data = data_all_prefixes_olmo7b,
  prior = prior_probs,
  iter = 8000,
  warmup = 4000,
  chains = 4,
  cores = 4,
  #control = list(adapt_delta=0.99, max_treedepth = 15),
  #control = list(max_treedepth = 20),
  file = '../Data/olmo7b_all_prefixes_individual_constraints'
)

# olmo7b_all_prefixes_individual_constraints_test = brm(log_odds ~ (Culture + Power + Freq + Len) + (1
#
#   data = data_all_prefixes_olmo7b,
#   prior = prior_probs,
#   iter = 8000,
#   warmup = 4000,
#   chains = 4,
#   cores = 4,
#   #control = list(adapt_delta=0.99, max_treedepth = 15),
#   #control = list(max_treedepth = 20),
#   #file = '../Data/olmo7b_all_prefixes_individual_constraints'
#
# )

gpt2_all_prefixes_individual_constraints = brm(log_odds ~ (Culture + Power + Freq + Len) + (Culture + P
  data = data_all_prefixes_gpt2,
  prior = prior_probs,

```

```

        iter = 8000,
        warmup = 4000,
        chains = 4,
        cores = 4,
        #control = list(adapt_delta=0.99, max_treedepth = 15),
        #control = list(max_treedepth = 20),
        file = '../Data/gpt2_all_prefixes_individual_constraints_test'
    )

gptoss_all_prefixes_individual_constraints = brm(log_odds ~ (Culture + Power + Freq + Len) + (Culture +
    data = data_all_prefixes_gptoss,
    prior = prior_probs,
    iter = 8000,
    warmup = 4000,
    chains = 4,
    cores = 4,
    #control = list(adapt_delta=0.99, max_treedepth = 15),
    #control = list(max_treedepth = 20),
    file = '../Data/gptoss_all_prefixes_individual_constraints'
)

olmo3_32b_all_prefixes_individual_constraints = brm(log_odds ~ (Culture + Power + Freq + Len) + (Culture
    data = data_all_prefixes_olmo3_32b,
    prior = prior_probs,
    iter = 8000,
    warmup = 4000,
    chains = 4,
    cores = 4,
    #control = list(adapt_delta=0.99, max_treedepth = 15),
    #control = list(max_treedepth = 20),
    file = '../Data/olmo3_32b_all_prefixes_individual_constraints'
)

olmo3_7b_all_prefixes_individual_constraints = brm(log_odds ~ (Culture + Power + Freq + Len) + (Culture
    data = data_all_prefixes_olmo3_7b,
    prior = prior_probs,
    iter = 8000,
    warmup = 4000,
    chains = 4,
    cores = 4,
    #control = list(adapt_delta=0.99, max_treedepth = 15),
    #control = list(max_treedepth = 20),
    file = '../Data/olmo3_7b_all_prefixes_individual_constraints'
)

fixef(all_models_all_prefixes_individual_constraints)

```

##	Estimate	Est.Error	Q2.5	Q97.5
## Intercept	-0.036650705	0.08160227	-0.200916935	0.119639594
## Culture	0.384817353	0.13448958	0.128627368	0.650823713
## Power	0.282674258	0.13176896	0.035919057	0.551801708
## Freq	-0.011902061	0.04354518	-0.097347262	0.074430065
## Len	-0.062208026	0.06842453	-0.196602163	0.070970243


```
## model1      0.095173179 0.02546022 0.044742168 0.145127846
## model2      0.053555913 0.02566149 0.003513109 0.104549589
## model3      0.027611508 0.02585058 -0.023237444 0.078286593
## model4     -0.073896789 0.02554878 -0.123943768 -0.023986754
## model5     -0.172754022 0.02556766 -0.222226433 -0.123461969
## model6      0.047723009 0.02531393 -0.001312399 0.097150142
## Culture:model1 0.237921435 0.04093075 0.158762032 0.317520198
## Culture:model2 -0.086962648 0.04018178 -0.165947146 -0.009267387
## Culture:model3 -0.367147667 0.04124398 -0.449063869 -0.287634515
## Culture:model4 -0.325710874 0.04013037 -0.405148514 -0.248023833
## Culture:model5 0.340811028 0.04093678 0.260561497 0.421542610
## Culture:model6 0.233926870 0.04115441 0.153627717 0.314451044
## Power:model1 -0.700250512 0.04165763 -0.781870838 -0.617008900
## Power:model2 -0.103778309 0.04198750 -0.186493351 -0.022101323
## Power:model3 -0.396802544 0.04133124 -0.477966960 -0.315135756
## Power:model4 0.761348100 0.04199564 0.678874543 0.843732864
## Power:model5 0.015565028 0.04136922 -0.066365028 0.095336269
## Power:model6 0.455297128 0.04261889 0.372479436 0.538613871
## Freq:model1 0.030917187 0.01370288 0.003887307 0.057564846
## Freq:model2 -0.007035555 0.01363156 -0.033959539 0.019145722
## Freq:model3 -0.089553491 0.01359587 -0.116121002 -0.063083951
## Freq:model4 0.022879634 0.01337802 -0.003306034 0.049180306
## Freq:model5 -0.015355329 0.01358254 -0.042031548 0.011376392
## Freq:model6 0.002879547 0.01365835 -0.023735571 0.029649844
## Len:model1 0.053871307 0.02102679 0.013034816 0.095148835
## Len:model2 0.145611003 0.02125301 0.103139078 0.186682821
## Len:model3 0.193374686 0.02102173 0.152074392 0.235231729
## Len:model4 -0.148463280 0.02093099 -0.190299444 -0.107424158
## Len:model5 -0.276501144 0.02104115 -0.318045949 -0.234842827
## Len:model6 -0.031469972 0.02134299 -0.073333694 0.010457647
```

```
fixef(gpt2xl_all_prefixes_individual_constraints)
```

```
##           Estimate Est.Error      Q2.5      Q97.5
## Intercept 0.0002545650 0.14430261 -0.2870336 0.2805305
## Culture   0.1966257904 0.20733479 -0.1820054 0.6426104
## Power     0.1272163788 0.20605449 -0.2595788 0.5560325
## Freq      0.0002873262 0.07938344 -0.1570666 0.1539826
## Len       0.0550616686 0.12071749 -0.1777679 0.2959928
```

```
fixef(olmo2_1b_all_prefixes_individual_constraints)
```

```
##           Estimate Est.Error      Q2.5      Q97.5
## Intercept -0.07902558 0.13371354 -0.3463314 0.17453422
## Culture   0.08142694 0.18362558 -0.2687737 0.45665523
## Power     0.88526809 0.28414388 0.3236498 1.44352091
## Freq      0.01496153 0.07716305 -0.1347296 0.16461143
## Len       -0.16123251 0.11776068 -0.3926491 0.06603183
```

```
fixef(olmo7b_all_prefixes_individual_constraints)
```

```
##           Estimate Est.Error      Q2.5      Q97.5
## Intercept -0.018891039 0.10146021 -0.217462144 0.1756954
## Culture   0.315750815 0.16444152 -0.003262526 0.6410623
## Power     0.221161274 0.16127185 -0.082037077 0.5562074
## Freq      0.047816877 0.05521188 -0.059515520 0.1553464
```

```
## Len      -0.001272537 0.09078743 -0.179295593 0.1730591
```

```
fixef(gpt2_all_prefixes_individual_constraints)
```

```
##           Estimate Est.Error      Q2.5      Q97.5
## Intercept  0.04106157 0.13311326 -0.2172583 0.3021968
## Culture    0.44857277 0.23258170 0.0267706 0.9298423
## Power      -0.17944136 0.19688881 -0.6035241 0.1802345
## Freq       0.02160311 0.07725416 -0.1285541 0.1706317
## Len        -0.02031985 0.11650522 -0.2470409 0.2118540
```

```
fixef(gptoss_all_prefixes_individual_constraints)
```

```
##           Estimate Est.Error      Q2.5      Q97.5
## Intercept -0.008962328 0.08394493 -0.17709897 0.15747347
## Culture    0.049763691 0.13467024 -0.21462043 0.31692455
## Power      -0.029644967 0.13372110 -0.29217664 0.23030362
## Freq       -0.107704173 0.04719754 -0.20118930 -0.01564115
## Len        0.113297068 0.07871063 -0.03729878 0.26930043
```

```
fixef(olmo3_32b_all_prefixes_individual_constraints)
```

```
##           Estimate Est.Error      Q2.5      Q97.5
## Intercept -0.164743687 0.1229122 -0.4080269 0.07598353
## Culture    0.614845960 0.2198184 0.1896864 1.04674546
## Power      0.225935246 0.1824826 -0.1112723 0.60491497
## Freq       -0.007683387 0.0691434 -0.1424753 0.12954478
## Len        -0.284722628 0.1119374 -0.5073376 -0.06937351
```

```
fixef(olmo3_7b_all_prefixes_individual_constraints)
```

```
##           Estimate Est.Error      Q2.5      Q97.5
## Intercept  0.004027597 0.12627073 -0.24370090 0.2508073
## Culture    0.490294623 0.23531525 0.06480464 0.9656488
## Power      0.595226270 0.25601767 0.12113466 1.0993270
## Freq       0.018366149 0.07473862 -0.12907132 0.1662645
## Len        -0.080748930 0.11077395 -0.29842514 0.1349232
```

Model Selection

```
##Lapse and no_final_stress droppped for having low vif for all models
```

```
gpt2xl_all_prefixes_all_constraints = brm(log_odds ~ (Culture + Power + Freq + Len + Intense + Percept)
      data = data_all_prefixes_gpt2xl,
      prior = prior_probs,
      iter = 6000,
      warmup = 3000,
      chains = 4,
      cores = 4,
      #control = list(adapt_delta=0.99, max_treedepth = 15),
      #control = list(max_treedepth = 20),
      file = '../Data/gpt2xl_all_prefixes_all_constraints'
    )
```

```
gpt2xl_all_prefixes_percept_power_dropped = brm(log_odds ~ (Culture + Freq + Len + Intense) + (Culture +
      data = data_all_prefixes_gpt2xl,
```

```

prior = prior_probs,
iter = 6000,
warmup = 3000,
chains = 4,
cores = 4,
#control = list(adapt_delta=0.99, max_treedepth = 15),
#control = list(max_treedepth = 20),
file = '../Data/gpt2xl_all_prefixes_percept_power_dropped'
)

olmo2_1b_all_prefixes_all_constraints = brm(log_odds ~ (Culture + Power + Freq + Len + Intense + Percept) + (Culture + Power + Freq + Len + Intense + Percept)
data = data_all_prefixes_olmo2_1b,
prior = prior_probs,
iter = 9000,
warmup = 4500,
chains = 4,
cores = 4,
#control = list(adapt_delta=0.99, max_treedepth = 15),
#control = list(max_treedepth = 20),
file = '../Data/olmo2_1b_all_prefixes_all_constraints'
)

olmo2_1b_all_prefixes_freq_dropped = brm(log_odds ~ (Culture + Power + Len + Intense + Percept) + (Culture + Power + Len + Intense + Percept)
data = data_all_prefixes_olmo2_1b,
prior = prior_probs,
iter = 9000,
warmup = 4500,
chains = 4,
cores = 4,
#control = list(adapt_delta=0.99, max_treedepth = 15),
#control = list(max_treedepth = 20),
file = '../Data/olmo2_1b_all_prefixes_freq_dropped'
)

olmo7b_all_prefixes_all_constraints = brm(log_odds ~ (Culture + Power + Freq + Len + Intense + Percept) + (Culture + Power + Freq + Len + Intense + Percept)
data = data_all_prefixes_olmo7b,
prior = prior_probs,
iter = 8000,
warmup = 4000,
chains = 4,
cores = 4,
#control = list(adapt_delta=0.99, max_treedepth = 15),
#control = list(max_treedepth = 20),
file = '../Data/olmo7b_all_prefixes_all_constraints'
)

olmo7b_all_prefixes_len_dropped = brm(log_odds ~ (Culture + Power + Freq + Intense + Percept) + (Culture + Power + Freq + Intense + Percept)
data = data_all_prefixes_olmo7b,
prior = prior_probs,
iter = 8000,
warmup = 4000,
chains = 4,

```

```

cores = 4,
#control = list(adapt_delta=0.99, max_treedepth = 15),
#control = list(max_treedepth = 20),
file = '../Data/olmo7b_all_prefixes_len_dropped'
)

gpt2_all_prefixes_all_constraints = brm(log_odds ~ (Culture + Power + Freq + Len + Intense + Percept) +
data = data_all_prefixes_gpt2,
prior = prior_probs,
iter = 8000,
warmup = 4000,
chains = 4,
cores = 4,
#control = list(adapt_delta=0.99, max_treedepth = 15),
#control = list(max_treedepth = 20),
file = '../Data/gpt2_all_prefixes_all_constraints'
)

gpt2_all_prefixes_freq_intense_dropped = brm(log_odds ~ (Culture + Power + Len + Percept) + (Culture +
data = data_all_prefixes_gpt2,
prior = prior_probs,
iter = 8000,
warmup = 4000,
chains = 4,
cores = 4,
#control = list(adapt_delta=0.99, max_treedepth = 15),
#control = list(max_treedepth = 20),
file = '../Data/gpt2_all_prefixes_freq_intense_dropped'
)

gptoss_all_prefixes_all_constraints = brm(log_odds ~ (Culture + Power + Freq + Len + Intense + Percept)
data = data_all_prefixes_gptoss,
prior = prior_probs,
iter = 8000,
warmup = 4000,
chains = 4,
cores = 4,
#control = list(adapt_delta=0.99, max_treedepth = 15),
#control = list(max_treedepth = 20),
file = '../Data/gptoss_all_prefixes_all_constraints'
)

gptoss_all_prefixes_culture_percept_dropped = brm(log_odds ~ (Power + Freq + Len + Intense) + (Power +
data = data_all_prefixes_gptoss,
prior = prior_probs,
iter = 8000,
warmup = 4000,

```

```

        chains = 4,
        cores = 4,
        #control = list(adapt_delta=0.99, max_treedepth = 15),
        #control = list(max_treedepth = 20),
        file = '../Data/gptoss_all_prefixes_culture_percept_dropped'
    )

olmo3_32b_all_prefixes_all_constraints = brm(log_odds ~ (Culture + Power + Freq + Len + Intense + Percept),
    data = data_all_prefixes_olmo3_32b,
    prior = prior_probs,
    iter = 8000,
    warmup = 4000,
    chains = 4,
    cores = 4,
    #control = list(adapt_delta=0.99, max_treedepth = 15),
    #control = list(max_treedepth = 20),
    file = '../Data/olmo3_32b_all_prefixes_all_constraints'
)

olmo3_32b_all_prefixes_freq_dropped = brm(log_odds ~ (Culture + Power + Len + Intense + Percept) + (Culture * Power),
    data = data_all_prefixes_olmo3_32b,
    prior = prior_probs,
    iter = 8000,
    warmup = 4000,
    chains = 4,
    cores = 4,
    #control = list(adapt_delta=0.99, max_treedepth = 15),
    #control = list(max_treedepth = 20),
    file = '../Data/olmo3_32b_all_prefixes_freq_dropped'
)

olmo3_7b_all_prefixes_all_constraints = brm(log_odds ~ (Culture + Power + Freq + Len + Intense + Percept),
    data = data_all_prefixes_olmo3_7b,
    prior = prior_probs,
    iter = 8000,
    warmup = 4000,
    chains = 4,
    cores = 4,
    #control = list(adapt_delta=0.99, max_treedepth = 15),
    #control = list(max_treedepth = 20),
    file = '../Data/olmo3_7b_all_prefixes_all_constraints'
)

olmo3_7b_all_prefixes_freq_percept_dropped = brm(log_odds ~ (Culture + Power + Len + Intense) + (Culture * Power),
    data = data_all_prefixes_olmo3_7b,
    prior = prior_probs,
    iter = 8000,
    warmup = 4000,
    chains = 4,
    cores = 4,
    #control = list(adapt_delta=0.99, max_treedepth = 15),
    #control = list(max_treedepth = 20),
    file = '../Data/olmo3_7b_all_prefixes_freq_percept_dropped'
)

```

```

)

fixef(gpt2xl_all_prefixes_all_constraints)

##           Estimate Est.Error      Q2.5      Q97.5
## Intercept  0.005428510 0.14220928 -0.2805975 0.2787851
## Culture    0.204849371 0.21240163 -0.1727138 0.6706435
## Power      0.069085900 0.20145844 -0.3170392 0.4861751
## Freq       -0.002581978 0.07902598 -0.1566084 0.1549722
## Len        0.043329240 0.12485780 -0.2009697 0.2894997
## Intense    0.365022647 0.34407325 -0.1792820 1.1644825
## Percept    -0.028614855 0.18165570 -0.3888987 0.3298569

fixef(olmo2_1b_all_prefixes_all_constraints)

##           Estimate Est.Error      Q2.5      Q97.5
## Intercept -0.06350839 0.13463743 -0.3416200 0.19562920
## Culture    0.07201977 0.18558619 -0.2856113 0.44957338
## Power      0.66620425 0.27122171  0.1628141 1.20409159
## Freq       0.00608074 0.07314131 -0.1371021 0.15183931
## Len       -0.19484511 0.11645130 -0.4296487 0.03330722
## Intense    0.95287691 0.48244623  0.1062995 1.93367596
## Percept    0.19003915 0.18651557 -0.1538731 0.58322199

fixef(olmo7b_all_prefixes_all_constraints)

##           Estimate Est.Error      Q2.5      Q97.5
## Intercept -0.004935696 0.10606304 -0.21938188 0.1997255
## Culture    0.288408074 0.17572811 -0.03304399 0.6542858
## Power      0.208113018 0.16666498 -0.10467856 0.5477454
## Freq       0.047237295 0.05657975 -0.06624020 0.1563133
## Len       -0.009572069 0.09065664 -0.18820520 0.1666347
## Intense    0.130992757 0.21569065 -0.26508030 0.6011214
## Percept    0.095643011 0.13892980 -0.17420958 0.3729199

fixef(gpt2_all_prefixes_all_constraints)

##           Estimate Est.Error      Q2.5      Q97.5
## Intercept  0.05146721 0.13595295 -0.21318008 0.3226756
## Culture    0.41263260 0.24174414 -0.01878102 0.9376518
## Power     -0.17959429 0.19851070 -0.59961873 0.1903835
## Freq       0.02182783 0.07472779 -0.12693955 0.1660141
## Len       -0.02689049 0.11349697 -0.25307028 0.1932199
## Intense   -0.04256189 0.24359292 -0.55610196 0.4198171
## Percept    0.10536144 0.17713505 -0.23144673 0.4614053

fixef(gptoss_all_prefixes_all_constraints)

##           Estimate Est.Error      Q2.5      Q97.5
## Intercept -0.004664545 0.09034198 -0.18095181 0.17328013
## Culture    0.049560019 0.13865563 -0.21732867 0.32754440
## Power     -0.044741306 0.13293406 -0.30885271 0.21323506
## Freq      -0.108823959 0.04833144 -0.20310582 -0.01411989
## Len        0.115310948 0.07814625 -0.03917815 0.26761636
## Intense    0.084948876 0.18987189 -0.27474834 0.47506937
## Percept    0.006149094 0.11956495 -0.23163487 0.24081233

```

```
fixef(olmo3_32b_all_prefixes_all_constraints)
```

	Estimate	Est.Error	Q2.5	Q97.5
## Intercept	-0.128818229	0.13071851	-0.39615837	0.12167191
## Culture	0.486390184	0.21833988	0.06913116	0.91554952
## Power	0.237391722	0.19130322	-0.11207676	0.63521279
## Freq	-0.005194701	0.06915302	-0.13955970	0.13171849
## Len	-0.292910016	0.11209137	-0.52082728	-0.07165446
## Intense	0.014486337	0.23192014	-0.44928050	0.49217797
## Percept	0.323681249	0.18240759	-0.01227949	0.69884905

```
fixef(olmo3_7b_all_prefixes_all_constraints)
```

	Estimate	Est.Error	Q2.5	Q97.5
## Intercept	0.004590039	0.12896599	-0.24581669	0.2531100
## Culture	0.531622812	0.24694069	0.08016639	1.0485852
## Power	0.473941583	0.24114602	0.03542822	0.9688246
## Freq	0.013247539	0.07393107	-0.12994885	0.1595430
## Len	-0.084056981	0.11363976	-0.31118314	0.1404966
## Intense	0.472635880	0.35368462	-0.08942153	1.2859389
## Percept	-0.026173941	0.16617441	-0.35739883	0.2945419

```
fixef(gpt2xl_all_prefixes_percept_power_dropped)
```

	Estimate	Est.Error	Q2.5	Q97.5
## Intercept	-0.0023569575	0.14192402	-0.2813767	0.2682509
## Culture	0.2101311257	0.20767881	-0.1630058	0.6545749
## Freq	0.0002606582	0.07912883	-0.1504977	0.1575764
## Len	0.0471443754	0.12122911	-0.1827893	0.2901037
## Intense	0.4126984800	0.34786755	-0.1484722	1.2069446

```
fixef(olmo2_1b_all_prefixes_freq_dropped)
```

	Estimate	Est.Error	Q2.5	Q97.5
## Intercept	-0.06260567	0.1318452	-0.32846411	0.19198112
## Culture	0.07017622	0.1798975	-0.27359276	0.43593578
## Power	0.67242474	0.2733772	0.16087566	1.22327169
## Len	-0.19758423	0.1155037	-0.42517565	0.02418417
## Intense	0.94499349	0.4830590	0.09746546	1.92375865
## Percept	0.19374304	0.1835420	-0.14751838	0.58134832

```
fixef(olmo7b_all_prefixes_len_dropped)
```

	Estimate	Est.Error	Q2.5	Q97.5
## Intercept	-0.003410082	0.10089205	-0.20480621	0.1929672
## Culture	0.287756275	0.17089872	-0.03228894	0.6324467
## Power	0.211701152	0.16841675	-0.09685871	0.5682005
## Freq	0.045154748	0.05673066	-0.06623652	0.1567871
## Intense	0.121943264	0.20698859	-0.27201394	0.5543612
## Percept	0.103816987	0.13673074	-0.16203246	0.3757286

```
fixef(gpt2_all_prefixes_freq_intense_dropped)
```

	Estimate	Est.Error	Q2.5	Q97.5
## Intercept	0.05378357	0.1344684	-0.2091791	0.3153895
## Culture	0.43032461	0.2290535	0.0114542	0.8943082
## Power	-0.16768861	0.1889397	-0.5652668	0.1839176

```
## Len      -0.02295287 0.1108969 -0.2410178 0.1950233
## Percept  0.11945215 0.1761284 -0.2122358 0.4831260
```

```
fixef(gptoss_all_prefixes_culture_percept_dropped)
```

```
##           Estimate Est.Error      Q2.5      Q97.5
## Intercept -0.01172708 0.08655887 -0.18752336 0.15765751
## Power     -0.03933542 0.13286198 -0.30668652 0.22202955
## Freq      -0.10224736 0.04534869 -0.19237127 -0.01398976
## Len       0.10796358 0.07880892 -0.04553773 0.26276137
## Intense   0.07854653 0.17982459 -0.27113654 0.44266710
```

```
fixef(olmo3_32b_all_prefixes_freq_dropped)
```

```
##           Estimate Est.Error      Q2.5      Q97.5
## Intercept -0.14185423 0.1238028 -0.39110092 0.09760030
## Culture   0.48781179 0.2042494 0.10592191 0.90714058
## Power     0.22477367 0.1823360 -0.12008356 0.60072394
## Len      -0.29723094 0.1103393 -0.51510037 -0.08493314
## Intense   0.01292947 0.2213360 -0.43259722 0.45999270
## Percept   0.32401737 0.1794900 -0.01621159 0.68541210
```

```
fixef(olmo3_7b_all_prefixes_freq_percept_dropped)
```

```
##           Estimate Est.Error      Q2.5      Q97.5
## Intercept 0.01238036 0.1242328 -0.23400000 0.2581281
## Culture   0.52962485 0.2156542 0.11426355 0.9449829
## Power     0.48622508 0.2420789 0.04868182 1.0029226
## Len      -0.07749524 0.1094477 -0.29719643 0.1322704
## Intense   0.48374214 0.3509433 -0.09469213 1.2449018
```

```
# 1. reshape wide (all terms and all metrics)
```

```
re_df = ranef(all_models_all_prefixes_individual_constraints)$prompt_value
```

```
re_df <- as.data.frame.table(re_df, responseName = "value")
names(re_df) <- c("prompt_value", "metric", "term", "value")
```

```
re_wide <- re_df %>%
  pivot_wider(
    names_from = c(term, metric),
    values_from = value)
# ) %>%
# mutate(
#   model3_Estimate = -(model1_Estimate + model2_Estimate),
#   model3_Q2.5     = -(model1_Q97.5   + model2_Q97.5),
#   model3_Q97.5    = -(model1_Q2.5    + model2_Q2.5)
# )
```

```
# 2. pivot long again so ggplot can facet
```

```
re_long <- re_wide %>%
  pivot_longer(
    cols = -prompt_value,
    names_to = c("term", "metric"),
    names_sep = "_",
    values_to = "value")
```



```

)

# 3. only keep useful terms
plot_df <- re_long %>%
  filter(term %in% c("Intercept", "Power", "Freq", "Len"))

# 4. create wide data for ribbons
plot_df_wide <- plot_df %>%
  pivot_wider(names_from = metric, values_from = value)

# 5. plot EVERYTHING properly
ggplot(plot_df_wide, aes(x = Estimate, y = prompt_value)) +
  geom_point() +
  geom_errorbarh(aes(xmin = Q2.5, xmax = Q97.5), height = 0) +
  facet_wrap(~ term, scales = "fixed") +
  theme_bw()

```

