# agreement-probing-stats

```
library(tidyr)
library(dplyr)
library(stringr)
library(readr)
library(readxl)
library(reshape2)

library(ggplot2)
library(viridis)
library(grid)
library(gtable)
library(lemon)

library(lme4)
library(lmerTest)
library(pbkrtest)
library(nlme)

library("broom.mixed")

library(extrafont)
```

## Loading datasets

```
gpt <- read_csv("gpt.csv")
```

```
## New names:
## Rows: 512 Columns: 94
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (7): N1, N2, Pred, Code, Sentence, subj_attention, obj_attention dbl (87):
## ...1, Unnamed: 0, Prot, Sent, Group, sh1, sh2, sh3, sh4, sh5, sh6,...
## 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.
## * `` -> `...1`
```

```
gpt_comp <- read_csv("../perplexity/ruGPT-35-13B_score_second.csv")
```

```
## New names:
## Rows: 512 Columns: 13
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (7): N1, N2, Pred, Code, Sentence, left, right dbl (6): ...1, Prot, Sent,
## Group, left_gpt_is_singular_score, left_gpt_is_pl...
## 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.
## * `` -> `...1`
```

```
rubert <- read_delim("rubert_all.csv", delim=",", locale = locale(decimal_mark = ","))
```

```
## New names:
## Rows: 512 Columns: 41
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (7): N1, N2, Pred, Code, Sentence, subj_attention, obj_attention dbl (34):
## ...1, Prot, Sent, Group, sh1, sh2, sh3, sh4, sh5, sh6, sh7, sh8, s...
## 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.
## * `` -> `...1`
```

```
rubert_comp <- read_csv("../perplexity/rubert_compatibility_score.csv")
```

```
## New names:
## Rows: 512 Columns: 20
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (10): N1, N2, Pred, Code, Sentence, left, right, masked_left, masked_ful... dbl
## (8): ...1, Prot, Sent, Group, left_bert_is_singular_score, left_bert_is... lgl
## (2): left_bert_is_singular, full_bert_is_singular
## 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.
## * `` -> `...1`
```

```
humans <- read_excel("./data number attr.xlsx")
```

We add columns encoding grammaticality kind of the sentence: `gram`(matical), (ungrammatical with) `distr`(actor), `ungram`(matical)

```
rubert_comp
```

```
## # A tibble: 512 x 20
##     ...1  Prot  Sent Group N1    N2    Pred  Code  Sentence          left  right
##    <dbl> <dbl> <dbl> <dbl> <chr> <chr> <chr> <chr> <chr>             <chr> <chr>
##  1     0     1     1     1 S     S     S     S_S-S           ~    ~    ~
##  2     1     2     1     1 S     S     P     P_S-S           ~    ~    ~
##  3     2     3     1     1 S     P     S     S_S-P           ~    ~    ~
##  4     3     4     1     1 S     P     P     P_S-P           ~    ~    ~
##  5     4     5     1     1 P     S     S     S_P-S           ~    ~    ~
##  6     5     6     1     1 P     S     P     P_P-S           ~    ~    ~
##  7     6     7     1     1 P     P     S     S_P-P           ~    ~    ~
##  8     7     8     1     1 P     P     P     P_P-P           ~    ~    ~
##  9     8     1     2     1 S     S     P     P_S-S           ~    ~    ~
## 10     9     2     2     1 S     P     S     S_S-P           ~    ~    ~
## # i 502 more rows
## # i 9 more variables: masked_left <chr>, masked_full <chr>,
## #   need_to_predict <chr>, left_bert_is_singular <lgl>,
## #   full_bert_is_singular <lgl>, left_bert_is_singular_score <dbl>,
## #   left_bert_is_plural_score <dbl>, full_bert_is_singular_score <dbl>,
## #   full_bert_is_plural_score <dbl>
```

```
add_grammaticality_column <- function(df){
  df %>% mutate(
    kind=as.factor(case_when(
      N1 == Pred ~ "gram",
```

```
      (N1 != Pred) & (N2 == Pred) ~ "distr",
      TRUE ~ "ungram"
    )),
    is_correct = N1 == Pred,
    is_distractor = N2 == Pred,
    distr_like_pl = N2 == "P" | (N2 == "S" & Group==2),
    .after=Pred
  )
}

convert_to_factor_columns <- function(
  df, to_factor_columns = c("N1", "N2", "Pred", "Code")
){
  df %>%
    mutate(across(all_of(to_factor_columns), as.factor))
}
```

```
rubert_comp %>%
  add_grammaticality_column(.) %>%
  convert_to_factor_columns(.) ->
  rubert_comp

rubert_comp %>%
  mutate(is_distr_different = N2 != N1, .after = Pred) ->
  rubert_comp

gpt_comp %>%
  add_grammaticality_column(.) %>%
  convert_to_factor_columns(.) ->
  gpt_comp
```

```
rubert_comp
```

```
## # A tibble: 512 x 25
##     ...1  Prot  Sent Group N1    N2    Pred  is_distr_different kind  is_correct
##    <dbl> <dbl> <dbl> <dbl> <fct> <fct> <fct> <lgl>              <fct> <lgl>
##  1     0     1     1     1 S     S     S     FALSE              gram  TRUE
##  2     1     2     1     1 S     S     P     FALSE              ungr~ FALSE
##  3     2     3     1     1 S     P     S     TRUE               gram  TRUE
##  4     3     4     1     1 S     P     P     TRUE               distr FALSE
##  5     4     5     1     1 P     S     S     TRUE               distr FALSE
##  6     5     6     1     1 P     S     P     TRUE               gram  TRUE
##  7     6     7     1     1 P     P     S     FALSE              ungr~ FALSE
##  8     7     8     1     1 P     P     P     FALSE              gram  TRUE
##  9     8     1     2     1 S     S     P     FALSE              ungr~ FALSE
## 10     9     2     2     1 S     P     S     TRUE               gram  TRUE
## # i 502 more rows
## # i 15 more variables: is_distractor <lgl>, distr_like_pl <lgl>, Code <fct>,
## #   Sentence <chr>, left <chr>, right <chr>, masked_left <chr>,
## #   masked_full <chr>, need_to_predict <chr>, left_bert_is_singular <lgl>,
## #   full_bert_is_singular <lgl>, left_bert_is_singular_score <dbl>,
## #   left_bert_is_plural_score <dbl>, full_bert_is_singular_score <dbl>,
## #   full_bert_is_plural_score <dbl>
```

## Adding RuBERT and GPT predicted number columns

We add columns that show RuBERT's preference of the correct number

```
compute_deltas <- function(df, df_name){
  sing_col = as.symbol(str_glue("left_{df_name}_is_singular_score"))
  plur_col = as.symbol(str_glue("left_{df_name}_is_plural_score"))

  df %>% mutate(
    verb_correct_delta = case_when(
      N1 == "S" ~ !!sing_col - !!plur_col,
      N1 == "P" ~  !!plur_col - !!sing_col            # N1 == "P"
    ),
    verb_correct_delta2 = case_when(
      N1 == "S" ~ !!sing_col,
      N1 == "P" ~  !!plur_col,            # N1 == "P"
    )
  ) %>% mutate(
    is_correct = N1 == Pred,
    is_distr_different = N2 != N1,
    .after = Pred
  )
}

expand_scores <- function(df){
  df %>%
    # we delete dataset-level Pred because it will cause duplicates
    filter(Pred == "S") %>%
    pivot_longer(matches("left_.*_score"), names_to = "Pred2", values_to = "score")  %>%
    mutate(Pred2 = if_else(str_detect(Pred2, "singular"), "S", "P"))
}
```

```
rubert_comp %>%
  compute_deltas(., "bert") ->
  rubert_comp

gpt_comp %>%
  compute_deltas(., "gpt") ->
  gpt_comp
```

These models only saw words before Pred (1 to 3), as such, actual Pred number of the sentences didn't influence them. Thus scores of sentences N1=N2=X, Pred=P and N1=N2=X, Pred=S are exactly the same

```
show_left_scores <- function(df, df_name){
  text_left_col = as.symbol(ifelse(df_name=="gpt", "left", "masked_left"))

  df %>%
    arrange(Sent, N1, N2, Pred) %>%
    select(N1, N2, Pred, Code, !!text_left_col, contains("left") & contains("score"))
}
```

```
rubert_comp %>%
  show_left_scores(., "bert")
```

```
## # A tibble: 512 x 7
##    N1    N2    Pred  Code  masked_left                  left_bert_is_singular~1
##    <fct> <fct> <fct> <fct> <chr>                                          <dbl>
```

```
##  1 P     P     P     P_P-P              [MASK]                -1.36
##  2 P     P     S     S_P-P              [MASK]                -1.36
##  3 P     S     P     P_P-S              [MASK]                -0.349
##  4 P     S     S     S_P-S              [MASK]                -0.349
##  5 S     P     P     P_S-P              [MASK]                 2.11
##  6 S     P     S     S_S-P              [MASK]                 2.11
##  7 S     S     P     P_S-S              [MASK]                 3.07
##  8 S     S     S     S_S-S              [MASK]                 3.07
##  9 P     P     P     P_P-P              [MASK]                -1.26
## 10 P     P     S     S_P-P              [MASK]                -1.26
## # i 502 more rows
## # i abbreviated name: 1: left_bert_is_singular_score
## # i 1 more variable: left_bert_is_plural_score <dbl>
```

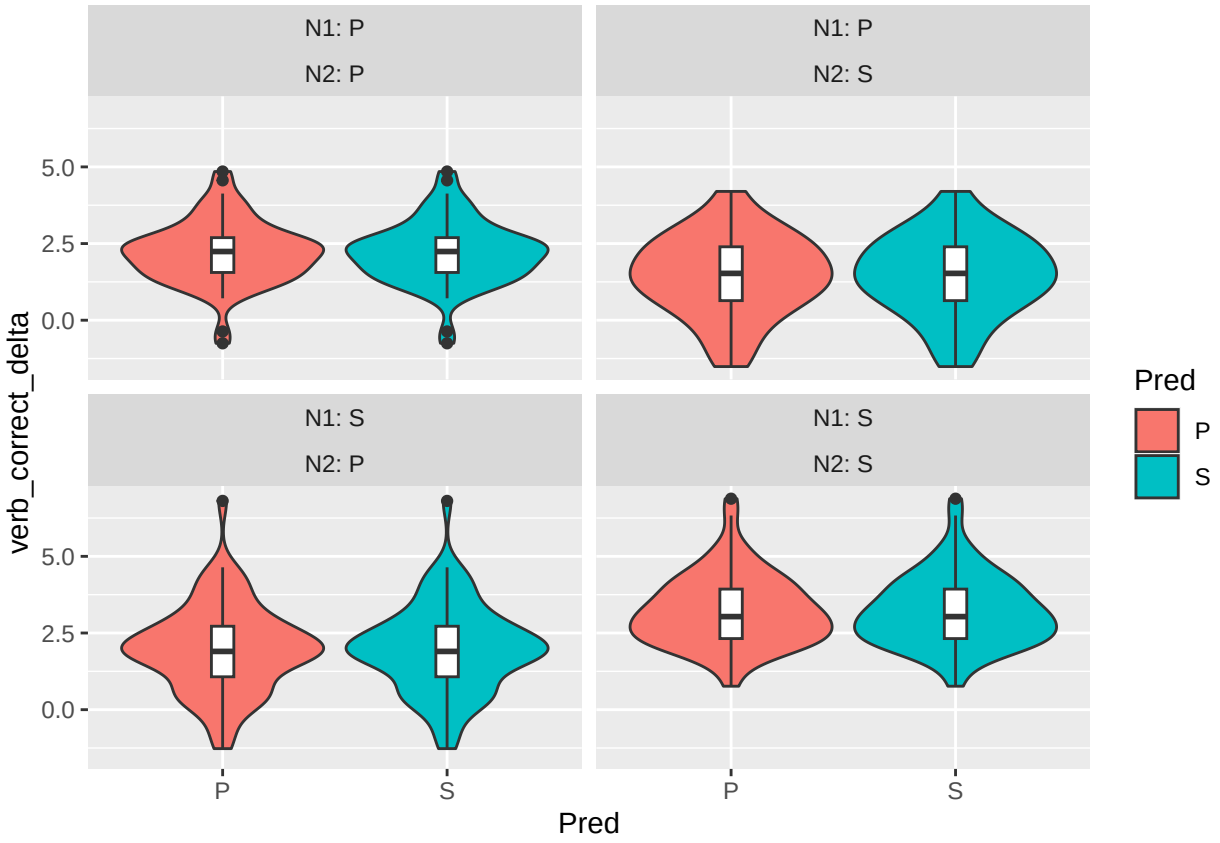
```
summarise_delta <- function(df){
  df %>%
    group_by(N1, N2, Pred) %>%
    summarise(
      across(verb_correct_delta,
             list(mean=mean, median=median)
      )
    )
}
```

```
rubert_comp %>%
  summarise_delta()
```

```
## `summarise()` has grouped output by 'N1', 'N2'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 8 x 5
## # Groups:   N1, N2 [4]
##   N1    N2    Pred  verb_correct_delta_mean verb_correct_delta_median
##   <fct> <fct> <fct>                   <dbl>                     <dbl>
## 1 P     P     P                        2.20                      2.24
## 2 P     P     S                        2.20                      2.24
## 3 P     S     P                        1.46                      1.53
## 4 P     S     S                        1.46                      1.53
## 5 S     P     P                        1.98                      1.90
## 6 S     P     S                        1.98                      1.90
## 7 S     S     P                        3.22                      3.03
## 8 S     S     S                        3.22                      3.03
```

```
rubert_comp %>%
  ggplot(aes(x=Pred, y=verb_correct_delta, fill=Pred)) +
  geom_violin() +
  geom_boxplot(width=0.1, fill="white") +
  facet_wrap(N1 ~ N2, labeller = label_both)
```




Naturally, this is true for GPT too.

```
gpt_comp %>%
    show_left_scores(., "gpt")
```

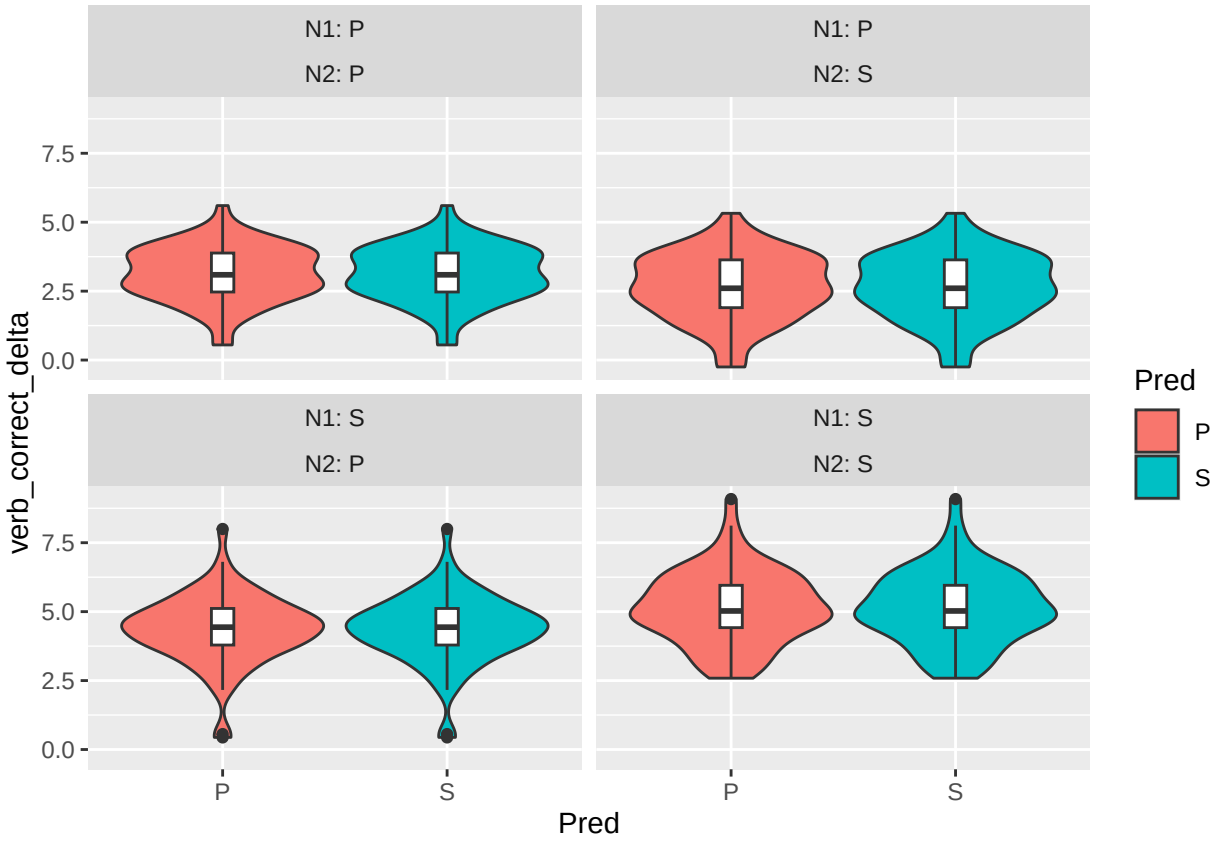
```
## # A tibble: 512 x 7
##    N1    N2    Pred  Code  left    left_gpt_is_singular~1 left_gpt_is_plural_s~2
##    <fct> <fct> <fct> <fct> <chr>                    <dbl>                  <dbl>
##  1 P     P     P     P_P-P     ~                   -3.68                  -1.29
##  2 P     P     S     S_P-P     ~                   -3.68                  -1.29
##  3 P     S     P     P_P-S     ~                   -2.09                  -0.497
##  4 P     S     S     S_P-S     ~                   -2.09                  -0.497
##  5 S     P     P     P_S-P     ~                   -0.139                 -4.03
##  6 S     P     S     S_S-P     ~                   -0.139                 -4.03
##  7 S     S     P     P_S-S     ~                    0.552                 -4.21
##  8 S     S     S     S_S-S     ~                    0.552                 -4.21
##  9 P     P     P     P_P-P     ~                   -2.88                   1.05
## 10 P     P     S     S_P-P     ~                   -2.88                   1.05
## # i 502 more rows
## # i abbreviated names: 1: left_gpt_is_singular_score,
## #   2: left_gpt_is_plural_score
```

```
gpt_comp %>%
  summarise_delta()
```

```
## `summarise()` has grouped output by 'N1', 'N2'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 8 x 5
## # Groups:   N1, N2 [4]
##   N1    N2    Pred  verb_correct_delta_mean verb_correct_delta_median
##   <fct> <fct> <fct>                   <dbl>                     <dbl>
## 1 P     P     P                        3.13                      3.09
## 2 P     P     S                        3.13                      3.09
## 3 P     S     P                        2.64                      2.60
## 4 P     S     S                        2.64                      2.60
## 5 S     P     P                        4.36                      4.44
## 6 S     P     S                        4.36                      4.44
## 7 S     S     P                        5.16                      5.03
## 8 S     S     S                        5.16                      5.03
```

```
gpt_comp %>%
  ggplot(aes(x=Pred, y=verb_correct_delta, fill=Pred)) +
  geom_violin() +
  geom_boxplot(width=0.1, fill="white") +
  facet_wrap(N1 ~ N2, labeller = label_both)
```




So, to avoid using duplicate data, we leave data for only one Pred (it doesn't matter which).

```
nrow(rubert_comp)
```

```
## [1] 512
```

```
rubert_comp %>%
  filter(Pred == "S") ->
  rubert_comp
```

```
nrow(rubert_comp)
```

```
## [1] 256
```

```
rubert_comp %>%
  write_csv(., "rubert_delta_comp.csv")
```

```
colnames(rubert_comp)
```

```
##  [1] "...1"                          "Prot"
##  [3] "Sent"                          "Group"
##  [5] "N1"                            "N2"
##  [7] "Pred"                          "is_distr_different"
##  [9] "kind"                          "is_correct"
## [11] "is_distractor"                 "distr_like_pl"
## [13] "Code"                          "Sentence"
## [15] "left"                          "right"
## [17] "masked_left"                   "masked_full"
## [19] "need_to_predict"               "left_bert_is_singular"
## [21] "full_bert_is_singular"         "left_bert_is_singular_score"
## [23] "left_bert_is_plural_score"     "full_bert_is_singular_score"
## [25] "full_bert_is_plural_score"     "verb_correct_delta"
## [27] "verb_correct_delta2"
```

```
nrow(gpt_comp)
```

```
## [1] 512
```

```
gpt_comp %>%
  filter(Pred == "S") ->
  gpt_comp

nrow(gpt_comp)
```

```
## [1] 256
```

```
gpt_comp %>%
  write_csv(., "gpt_delta_comp.csv")
```

```
colnames(gpt_comp)
```

```
##  [1] "...1"                          "Prot"
##  [3] "Sent"                          "Group"
##  [5] "N1"                            "N2"
##  [7] "Pred"                          "is_distr_different"
##  [9] "kind"                          "is_correct"
## [11] "is_distractor"                 "distr_like_pl"
## [13] "Code"                          "Sentence"
## [15] "left"                          "right"
## [17] "left_gpt_is_singular_score"    "left_gpt_is_plural_score"
## [19] "verb_correct_delta"            "verb_correct_delta2"
```

## Stat Functions

```
test_lmer_signif_KR <- function(model, x_vars, categorial=FALSE){
  res <- data.frame()
```

```
  print(model)
  for (var in x_vars){
    reduced_model = update(model, as.formula(paste("~ . -", var)))

    # print(var)
    # print(reduced_model)

    comp = KRmodcomp(model, reduced_model)
    # print(comp$test)

    var_df <- comp$test[1, c("stat", "ddf", "p.value")]
    var_df$mod_is_singular = isSingular(reduced_model)
    rownames(var_df) <- c(var)

    res <- rbind(res, var_df)
  }

  res %>%
    tibble::rownames_to_column(., "factor") %>%
    pivot_wider(
      names_from = factor,
      values_from = !factor,
      names_glue = "{factor}_KR_{.value}"
    )
}
```

## Reporting functions

### Functions to fetch the relevant part of the dataframe

```
make_subj_col <- function(i){
  paste0("sh", i)
}

make_obj_col <- function(i){
  paste0("oh", i)
}

make_df_for_head <- function(df, head_i){
  # i = 5

  full_attention_cols = c("subj_attention", "obj_attention")

  subject_head_col = paste("sh", head_i, sep="")
  object_head_col = paste("oh", head_i, sep="")
  used_attention_cols = c(subject_head_col, object_head_col)

  return (df %>%
    select(
      !!used_attention_cols
      | (!starts_with("sh") & !starts_with("oh") & !(!!full_attention_cols))
    ))
}
```

```
make_meta_info <- function(i, model_name, ...){
  meta_info <- data.frame(i, model_name)
  names(meta_info) <- c("i", "model")

  meta_info
}
```

**Functions to fetch info**

```
lmer_model_info_to_df <- function(lmer_model, ...){
  call = as.character(lmer_model@call)
  call = paste0(call[1], "(", call[2], ", data=", call[3],  ")")

  optinfo = lmer_model@optinfo
  optimizer = optinfo$optimizer

  if (! ("messages" %in% names(optinfo$conv$lme4) )){
    optimizer_message = optinfo$message
  } else {
    optimizer_message = paste(optinfo$conv$lme4$messages, collapse = "; ")
  }

  n_observations = nobs(lmer_model)

  n_rand_effect_groups = ngrps(lmer_model)
  colnames(n_rand_effect_groups) <- rownames(n_rand_effect_groups)
  rownames(n_rand_effect_groups) <- NULL

  data.frame(cbind(
    call, n_observations, n_rand_effect_groups,
    optimizer, optimizer_message
  ))
}
```

```
coeffs_from_summary_to_df <- function(
  model_summary, categorial=FALSE, ...
){
  model.coeffs = model_summary$coefficients
  df = data.frame(model.coeffs, check.names=FALSE)
  df = tibble::rownames_to_column(df, "factor")

  if (categorial){
    p_value_col = "Pr(>|z|)"
  } else {
    p_value_col = "Pr(>|t|)"
  }
  # print(c(categorial, p_value_col))

  df %>% mutate(
    factor = case_when(
        factor=="(Intercept)" ~ "Intercept",
        str_detect(factor, "^Y") ~ substr(factor, 1, 3),
```

```
        TRUE ~ factor
    )
  ) %>% rename(
    p.value = p_value_col,
    stdError = "Std. Error"
  ) -> df

  df %>% pivot_wider(
    names_from = factor,
    values_from = !factor,
    names_glue = "{factor}_{.value}"
  )
}
```

```
extra_info_from_summary_to_df <- function(model_summary, ...){
  REML_df <- data.frame(model_summary$AICtab)
  colnames(REML_df) <- rownames(REML_df)
  rownames(REML_df) <- NULL

  loglik_df <- data.frame(model_summary$logLik)
  colnames(loglik_df) <- c("logLik")

  bind_cols(REML_df, loglik_df)
}

extra_info_from_summary_to_df_cat <- function(model_summary, ...){
  bind_rows(model_summary$AICtab)
}
```

```
get_full_lmer_info <- function(model, model_summary, categorial=FALSE, ...){

  model.coeffs = coeffs_from_summary_to_df(model_summary, categorial=categorial, ...)
  model_info = lmer_model_info_to_df(model, ...)

  if (categorial){
    extra_fit_info = extra_info_from_summary_to_df_cat(model_summary, ...)
  }
  else {
    extra_fit_info = extra_info_from_summary_to_df(model_summary, ...)
  }

  # print(c(dim(model.coeffs), dim(model_info), dim(extra_fit_info)))

  bind_cols(model_info, extra_fit_info, model.coeffs)
}
```

## Model and human preference of the correct number

### Plots

```
rubert_comp %>%
  # mutate(kind = factor(kind, levels=c("ungram", "distr", "gram"))) %>%
  group_by(N1, distr_like_pl) %>%
  summarise(across(verb_correct_delta, list(mean=mean, median=median, sd=sd)))
```

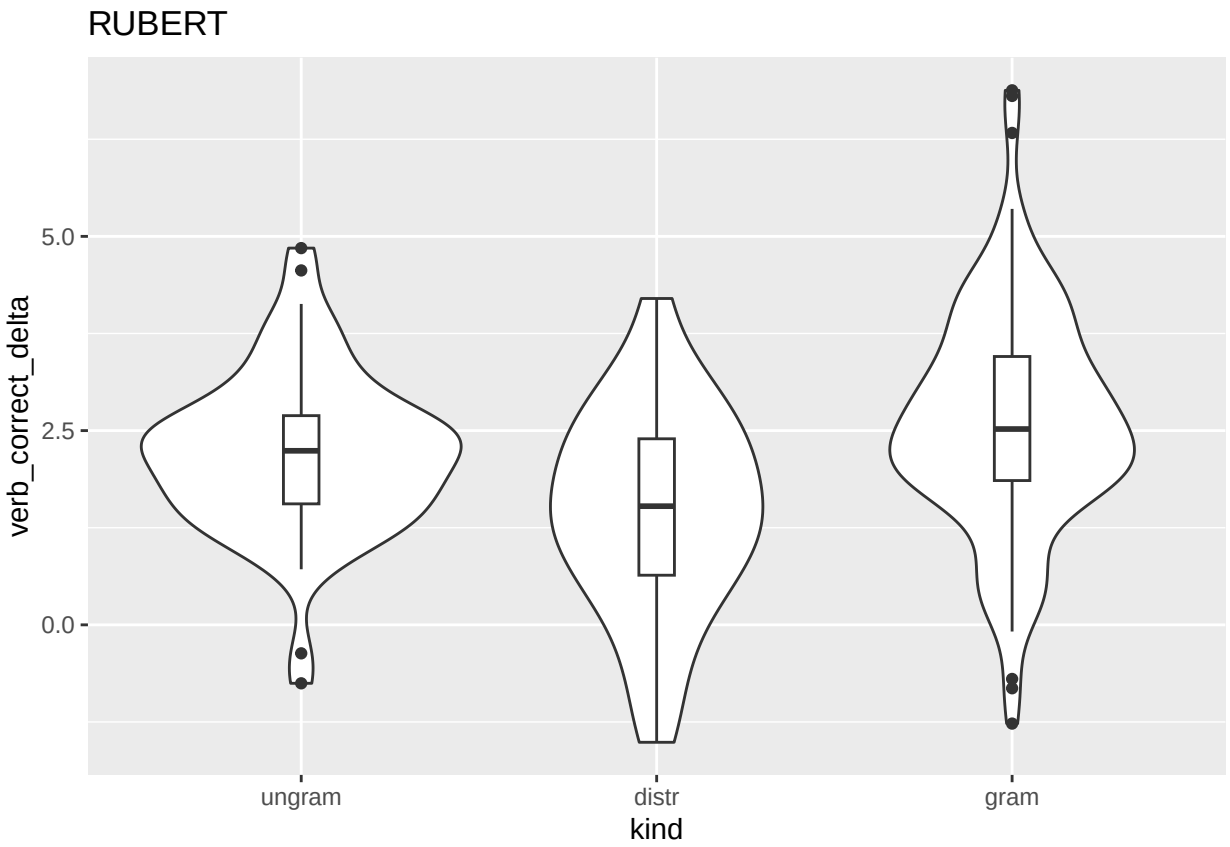
```
## `summarise()` has grouped output by 'N1'. You can override using the `.groups`
## argument.
```

```
## # A tibble: 4 x 5
## # Groups:   N1 [2]
##   N1    distr_like_pl verb_correct_delta_mean verb_correct_delta_median
##   <fct> <lgl>                          <dbl>                     <dbl>
## 1 P     FALSE                          0.972                     0.914
## 2 P     TRUE                           2.12                      2.14
## 3 S     FALSE                          3.36                      3.26
## 4 S     TRUE                           2.35                      2.29
## # i 1 more variable: verb_correct_delta_sd <dbl>
```

```
rubert_comp %>%
  mutate(kind = factor(kind, levels=c("ungram", "distr", "gram"))) %>%
  group_by(kind) %>%
  summarise(across(verb_correct_delta, list(mean=mean, median=median, sd=sd)))
```

```
## # A tibble: 3 x 4
##   kind   verb_correct_delta_mean verb_correct_delta_median verb_correct_delta_sd
##   <fct>                    <dbl>                     <dbl>                 <dbl>
## 1 ungram                    2.20                      2.24                  1.02
## 2 distr                     1.46                      1.53                  1.31
## 3 gram                      2.60                      2.52                  1.44
```

```
rubert_comp %>%
  mutate(kind = factor(kind, levels=c("ungram", "distr", "gram"))) %>%
  ggplot(aes(x=kind, y=verb_correct_delta)) +
    geom_violin() +
    geom_boxplot(width=0.1, fill="white") +
    labs(title="RUBERT")
```




```
gpt_comp %>%
  # mutate(kind = factor(kind, levels=c("ungram", "distr", "gram"))) %>%
  group_by(N1, distr_like_pl) %>%
  summarise(across(verb_correct_delta, list(mean=mean, median=median, sd=sd)))
```

```
## `summarise()` has grouped output by 'N1'. You can override using the `.groups`
## argument.
```

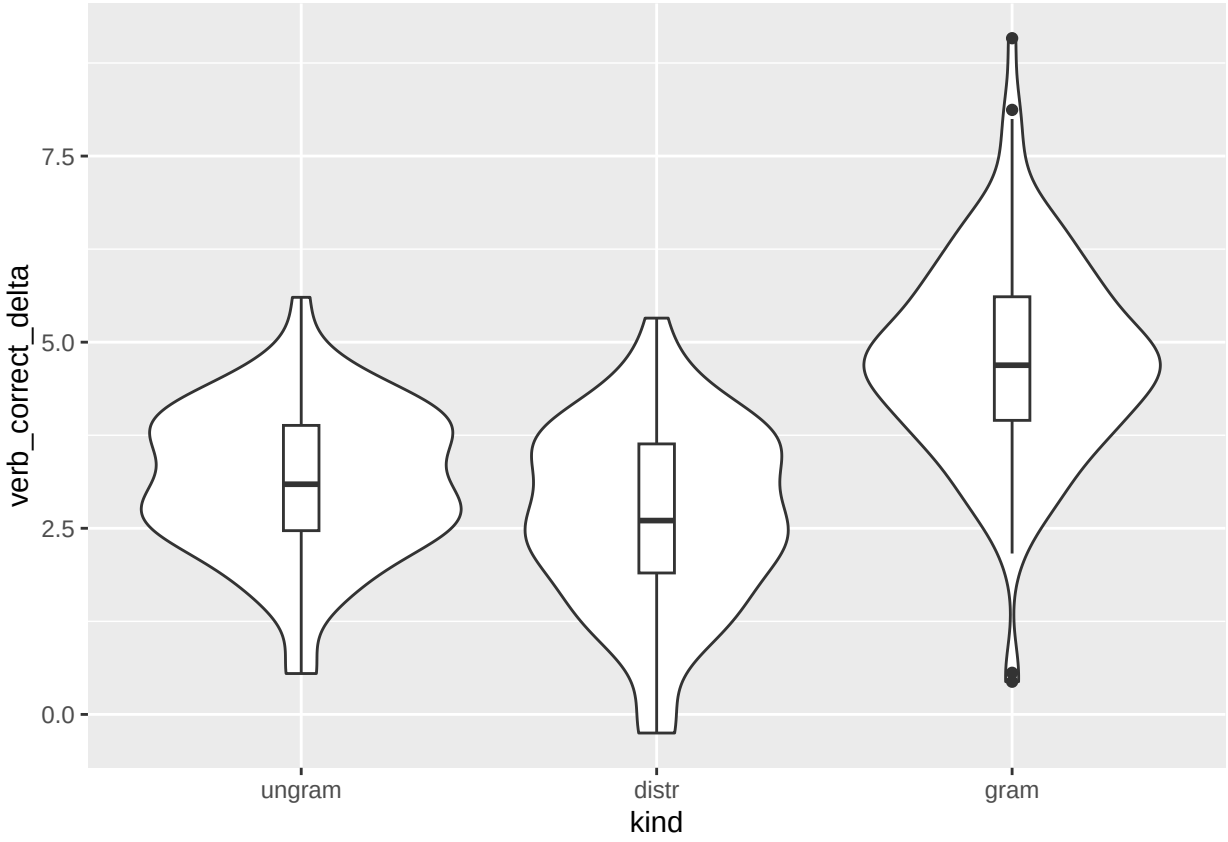
```
## # A tibble: 4 x 5
## # Groups:   N1 [2]
##   N1    distr_like_pl verb_correct_delta_mean verb_correct_delta_median
##   <fct> <lgl>                          <dbl>                     <dbl>
## 1 P     FALSE                           2.43                      2.27
## 2 P     TRUE                            3.04                      3.07
## 3 S     FALSE                           5.29                      5.03
## 4 S     TRUE                            4.58                      4.61
## # i 1 more variable: verb_correct_delta_sd <dbl>
```

```
gpt_comp %>%
  mutate(kind = factor(kind, levels=c("ungram", "distr", "gram"))) %>%
  group_by(kind) %>%
  summarise(across(verb_correct_delta, list(mean=mean, median=median, sd=sd)))
```

```
## # A tibble: 3 x 4
##   kind   verb_correct_delta_mean verb_correct_delta_median verb_correct_delta_sd
##   <fct>                    <dbl>                     <dbl>                 <dbl>
## 1 ungram                    3.13                      3.09                 0.995
## 2 distr                     2.64                      2.60                 1.20
```

```
## 3 gram                    4.76                    4.69                    1.35
```

```
gpt_comp %>%
  mutate(kind = factor(kind, levels=c("ungram", "distr", "gram"))) %>%
  ggplot(aes(x=kind, y=verb_correct_delta)) +
    geom_violin() +
    geom_boxplot(width=0.1, fill="white")
```




**3-valued kind calculation, distinct pred**

```
all_coeffs = tibble()

make_coeffs_table <- function(model_fit, model_name, data_subset){
  stopifnot(model_name %in% c("rubert", "rugpt", "humans"))
  stopifnot(data_subset %in% c("acc", "gen", "both"))

  cbind(list(model=model_name, data=data_subset),  tidy(model_fit, "fixed"))
}
```

```
rubert_comp
```

```
## # A tibble: 256 x 27
##      ...1  Prot  Sent Group N1    N2    Pred  is_distr_different kind  is_correct
##     <dbl> <dbl> <dbl> <dbl> <fct> <fct> <fct> <lgl>              <fct> <lgl>
##  1     0     1     1     1 S     S     S     FALSE              gram  TRUE
##  2     2     3     1     1 S     P     S     TRUE               gram  TRUE
##  3     4     5     1     1 P     S     S     TRUE               distr FALSE
```

```
##  4     6     7     1     1 P     P     S     FALSE             ungr~ FALSE
##  5     9     2     2     1 S     P     S     TRUE              gram  TRUE
##  6    11     4     2     1 P     S     S     TRUE              distr FALSE
##  7    13     6     2     1 P     P     S     FALSE             ungr~ FALSE
##  8    15     8     2     1 S     S     S     FALSE             gram  TRUE
##  9    16     1     3     1 S     P     S     TRUE              gram  TRUE
## 10    18     3     3     1 P     S     S     TRUE              distr FALSE
## # i 246 more rows
## # i 17 more variables: is_distractor <lgl>, distr_like_pl <lgl>, Code <fct>,
## #   Sentence <chr>, left <chr>, right <chr>, masked_left <chr>,
## #   masked_full <chr>, need_to_predict <chr>, left_bert_is_singular <lgl>,
## #   full_bert_is_singular <lgl>, left_bert_is_singular_score <dbl>,
## #   left_bert_is_plural_score <dbl>, full_bert_is_singular_score <dbl>,
## #   full_bert_is_plural_score <dbl>, verb_correct_delta <dbl>, ...
```

```
make_kind2_scores <- function(df){
  df %>%
    expand_scores() %>%
    mutate(
      kind2=as.factor(case_when(
        N1 == Pred2 ~ "gram",
        (N1 != Pred2) & (N2 == Pred2) ~ "distr",
        TRUE ~ "ungram"
      )),
      .after=Pred2
    )
}
```

```
rubert_comp %>%
  make_kind2_scores() ->
  rubert_comp_scores

rubert_comp_scores
```

```
## # A tibble: 512 x 28
##     ...1  Prot  Sent Group N1    N2    Pred  is_distr_different kind  is_correct
##    <dbl> <dbl> <dbl> <dbl> <fct> <fct> <fct> <lgl>              <fct> <lgl>
##  1     0     1     1     1 S     S     S     FALSE              gram  TRUE
##  2     0     1     1     1 S     S     S     FALSE              gram  TRUE
##  3     2     3     1     1 S     P     S     TRUE               gram  TRUE
##  4     2     3     1     1 S     P     S     TRUE               gram  TRUE
##  5     4     5     1     1 P     S     S     TRUE               distr FALSE
##  6     4     5     1     1 P     S     S     TRUE               distr FALSE
##  7     6     7     1     1 P     P     S     FALSE              ungr~ FALSE
##  8     6     7     1     1 P     P     S     FALSE              ungr~ FALSE
##  9     9     2     2     1 S     P     S     TRUE               gram  TRUE
## 10     9     2     2     1 S     P     S     TRUE               gram  TRUE
## # i 502 more rows
## # i 18 more variables: is_distractor <lgl>, distr_like_pl <lgl>, Code <fct>,
## #   Sentence <chr>, left <chr>, right <chr>, masked_left <chr>,
## #   masked_full <chr>, need_to_predict <chr>, left_bert_is_singular <lgl>,
## #   full_bert_is_singular <lgl>, full_bert_is_singular_score <dbl>,
## #   full_bert_is_plural_score <dbl>, verb_correct_delta <dbl>,
## #   verb_correct_delta2 <dbl>, Pred2 <chr>, kind2 <fct>, score <dbl>
```

```
gpt_comp %>%
  make_kind2_scores() ->
  gpt_comp_scores

gpt_comp_scores
```

```
## # A tibble: 512 x 21
##     ...1  Prot  Sent Group N1    N2    Pred  is_distr_different kind  is_correct
##    <dbl> <dbl> <dbl> <dbl> <fct> <fct> <fct> <lgl>              <fct> <lgl>
##  1     0     1     1     1 S     S     S     FALSE              gram  TRUE
##  2     0     1     1     1 S     S     S     FALSE              gram  TRUE
##  3     2     3     1     1 S     P     S     TRUE               gram  TRUE
##  4     2     3     1     1 S     P     S     TRUE               gram  TRUE
##  5     4     5     1     1 P     S     S     TRUE               distr FALSE
##  6     4     5     1     1 P     S     S     TRUE               distr FALSE
##  7     6     7     1     1 P     P     S     FALSE              ungr~ FALSE
##  8     6     7     1     1 P     P     S     FALSE              ungr~ FALSE
##  9     9     2     2     1 S     P     S     TRUE               gram  TRUE
## 10     9     2     2     1 S     P     S     TRUE               gram  TRUE
## # i 502 more rows
## # i 11 more variables: is_distractor <lgl>, distr_like_pl <lgl>, Code <fct>,
## #   Sentence <chr>, left <chr>, right <chr>, verb_correct_delta <dbl>,
## #   verb_correct_delta2 <dbl>, Pred2 <chr>, kind2 <fct>, score <dbl>
```

```
mod5_rubert = lmer(score ~ N1 + N2 + kind2 + (1 | Sent), data=rubert_comp_scores)
summary(mod5_rubert)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: score ~ N1 + N2 + kind2 + (1 | Sent)
##    Data: rubert_comp_scores
##
## REML criterion at convergence: 1626.1
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.1744 -0.6029  0.0144  0.6318  2.9579
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Sent     (Intercept) 2.1876   1.4790
##  Residual             0.9532   0.9763
## Number of obs: 512, groups:  Sent, 64
##
## Fixed effects:
##              Estimate Std. Error        df t value Pr(>|t|)
## (Intercept)   0.19926    0.21296  98.80736   0.936    0.352
## N1S          -0.80477    0.08629 444.00000  -9.326  < 2e-16 ***
## N2S          -0.46206    0.08629 444.00000  -5.354 1.38e-07 ***
## kind2gram     1.87799    0.10569 444.00000  17.769  < 2e-16 ***
## kind2ungram  -0.67751    0.12204 444.00000  -5.552 4.88e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Correlation of Fixed Effects:
##             (Intr) N1S    N2S    knd2gr
## N1S         -0.203
## N2S         -0.203  0.000
## kind2gram   -0.331  0.000  0.000
## kind2ungram -0.287  0.000  0.000  0.577
```

```
mod5_rubert_coeffs = make_coeffs_table(mod5_rubert, "rubert", "both")
all_coeffs = rbind(all_coeffs, mod5_rubert_coeffs)

info5_rubert <- get_full_lmer_info(mod5_rubert, summary(mod5_rubert))

mod5KR_rubert <- test_lmer_signif_KR(mod5_rubert, c("N1", "N2", "kind2"))
```

```
## Linear mixed model fit by REML ['lmerModLmerTest']
## Formula: score ~ N1 + N2 + kind2 + (1 | Sent)
##    Data: rubert_comp_scores
## REML criterion at convergence: 1626.12
## Random effects:
##  Groups   Name        Std.Dev.
##  Sent     (Intercept) 1.4790
##  Residual             0.9763
## Number of obs: 512, groups:  Sent, 64
## Fixed Effects:
## (Intercept)          N1S          N2S    kind2gram  kind2ungram
##      0.1993      -0.8048      -0.4621       1.8780      -0.6775
```

```
mod5KR_rubert
```

```
## # A tibble: 1 x 12
##   N1_KR_stat N2_KR_stat kind2_KR_stat N1_KR_ddf N2_KR_ddf kind2_KR_ddf
##        <dbl>      <dbl>         <dbl>     <dbl>     <dbl>        <dbl>
## 1       87.0       28.7          345.      444.      444.         444.
## # i 6 more variables: N1_KR_p.value <dbl>, N2_KR_p.value <dbl>,
## #   kind2_KR_p.value <dbl>, N1_KR_mod_is_singular <lgl>,
## #   N2_KR_mod_is_singular <lgl>, kind2_KR_mod_is_singular <lgl>
```

```
mod5_gpt = lmer(score ~ N1 + N2 + kind2 + (1 | Sent), data=gpt_comp_scores)
summary(mod5_gpt)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: score ~ N1 + N2 + kind2 + (1 | Sent)
##    Data: gpt_comp_scores
##
## REML criterion at convergence: 1623.1
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.2091 -0.6049  0.0837  0.6230  2.6997
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Sent     (Intercept) 1.064    1.032
##  Residual             1.039    1.019
## Number of obs: 512, groups:  Sent, 64
```

```
##
## Fixed effects:
##              Estimate Std. Error        df t value Pr(>|t|)
## (Intercept)  -1.82950    0.16973 144.12278 -10.779  < 2e-16 ***
## N1S          -0.55459    0.09011 444.00000  -6.155 1.68e-09 ***
## N2S          -0.04194    0.09011 444.00000  -0.465    0.642
## kind2gram     3.55264    0.11036 444.00000  32.191  < 2e-16 ***
## kind2ungram  -0.53859    0.12743 444.00000  -4.226 2.88e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) N1S    N2S    knd2gr
## N1S         -0.265
## N2S         -0.265  0.000
## kind2gram   -0.433  0.000  0.000
## kind2ungram -0.375  0.000  0.000  0.577
```

```
mod5_gpt_coeffs = make_coeffs_table(mod5_gpt, "rugpt", "both")
all_coeffs = rbind(all_coeffs, mod5_gpt_coeffs)

info5_gpt <- get_full_lmer_info(mod5_gpt, summary(mod5_gpt))

mod5KR_gpt <- test_lmer_signif_KR(mod5_gpt, c("N1", "N2", "kind2"))
```

```
## Linear mixed model fit by REML ['lmerModLmerTest']
## Formula: score ~ N1 + N2 + kind2 + (1 | Sent)
##    Data: gpt_comp_scores
## REML criterion at convergence: 1623.071
## Random effects:
##  Groups   Name        Std.Dev.
##  Sent     (Intercept) 1.032
##  Residual             1.019
## Number of obs: 512, groups:  Sent, 64
## Fixed Effects:
## (Intercept)          N1S          N2S    kind2gram  kind2ungram
##    -1.82950     -0.55459     -0.04194      3.55264     -0.53859
```

```
mod5KR_gpt
```

```
## # A tibble: 1 x 12
##   N1_KR_stat N2_KR_stat kind2_KR_stat N1_KR_ddf N2_KR_ddf kind2_KR_ddf
##        <dbl>      <dbl>         <dbl>     <dbl>     <dbl>        <dbl>
## 1       37.9      0.217          908.      444.      444.         444.
## # i 6 more variables: N1_KR_p.value <dbl>, N2_KR_p.value <dbl>,
## #   kind2_KR_p.value <dbl>, N1_KR_mod_is_singular <lgl>,
## #   N2_KR_mod_is_singular <lgl>, kind2_KR_mod_is_singular <lgl>
```

```
rubert_comp_scores %>% filter(Group == 1) -> rubert_comp_scores_acc

mod5_rubert_acc = lmer(score ~ N1 + N2 + kind2 + (1 | Sent), data=rubert_comp_scores_acc)
summary(mod5_rubert_acc)
```

**by Group**

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: score ~ N1 + N2 + kind2 + (1 | Sent)
##    Data: rubert_comp_scores_acc
##
## REML criterion at convergence: 783.6
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.68086 -0.58109  0.03767  0.66305  2.70958
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Sent     (Intercept) 1.8235   1.3504
##  Residual             0.8528   0.9235
## Number of obs: 256, groups:  Sent, 32
##
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)   0.1265     0.2774  49.9484   0.456     0.65
## N1S          -0.6882     0.1154 220.0000  -5.962 9.82e-09 ***
## N2S          -0.5438     0.1154 220.0000  -4.711 4.37e-06 ***
## kind2gram     1.4827     0.1414 220.0000  10.488  < 2e-16 ***
## kind2ungram  -1.0590     0.1632 220.0000  -6.487 5.70e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) N1S    N2S    knd2gr
## N1S         -0.208
## N2S         -0.208  0.000
## kind2gram   -0.340  0.000  0.000
## kind2ungram -0.294  0.000  0.000  0.577
```

```
mod5_rubert_acc_coeffs = make_coeffs_table(mod5_rubert_acc, "rubert", "acc")
all_coeffs = rbind(all_coeffs, mod5_rubert_acc_coeffs)

info5_rubert_acc <- get_full_lmer_info(mod5_rubert_acc, summary(mod5_rubert_acc))

mod5KR_rubert_acc <- test_lmer_signif_KR(mod5_rubert_acc, c("N1", "N2", "kind2"))
```

```
## Linear mixed model fit by REML ['lmerModLmerTest']
## Formula: score ~ N1 + N2 + kind2 + (1 | Sent)
##    Data: rubert_comp_scores_acc
## REML criterion at convergence: 783.604
## Random effects:
##  Groups   Name        Std.Dev.
##  Sent     (Intercept) 1.3504
##  Residual             0.9235
## Number of obs: 256, groups:  Sent, 32
## Fixed Effects:
## (Intercept)          N1S          N2S    kind2gram  kind2ungram
##      0.1265      -0.6882      -0.5438       1.4827      -1.0590
```

```
mod5KR_rubert_acc
```

```
## # A tibble: 1 x 12
##   N1_KR_stat N2_KR_stat kind2_KR_stat N1_KR_ddf N2_KR_ddf kind2_KR_ddf
##        <dbl>      <dbl>         <dbl>     <dbl>     <dbl>        <dbl>
## 1       35.5       22.2          173.      220.      220.         220.
## # i 6 more variables: N1_KR_p.value <dbl>, N2_KR_p.value <dbl>,
## #   kind2_KR_p.value <dbl>, N1_KR_mod_is_singular <lgl>,
## #   N2_KR_mod_is_singular <lgl>, kind2_KR_mod_is_singular <lgl>
```

```
rubert_comp_scores %>%
  filter(Group == 2) ->
  rubert_comp_scores_gen

mod5_rubert_gen = lmer(score ~ N1 + N2 + kind2 + (1 | Sent), data=rubert_comp_scores_gen)
summary(mod5_rubert_gen)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: score ~ N1 + N2 + kind2 + (1 | Sent)
##    Data: rubert_comp_scores_gen
##
## REML criterion at convergence: 825.1
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.74049 -0.58231 -0.01764  0.58483  2.53483
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Sent     (Intercept) 2.3832   1.5438
##  Residual             0.9924   0.9962
## Number of obs: 256, groups:  Sent, 32
##
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)   0.2720     0.3126  47.8225   0.870  0.38863
## N1S          -0.9213     0.1245 220.0000  -7.399 2.87e-12 ***
## N2S          -0.3804     0.1245 220.0000  -3.054  0.00253 **
## kind2gram     2.2733     0.1525 220.0000  14.906  < 2e-16 ***
## kind2ungram  -0.2960     0.1761 220.0000  -1.681  0.09421 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) N1S    N2S    knd2gr
## N1S         -0.199
## N2S         -0.199  0.000
## kind2gram   -0.325  0.000  0.000
## kind2ungram -0.282  0.000  0.000  0.577
```

```
mod5_rubert_gen_coeffs = make_coeffs_table(mod5_rubert_gen, "rubert", "gen")
all_coeffs = rbind(all_coeffs, mod5_rubert_gen_coeffs)

info5_rubert_gen <- get_full_lmer_info(mod5_rubert_gen, summary(mod5_rubert_gen))
```

```
mod5KR_rubert_gen <- test_lmer_signif_KR(mod5_rubert_gen, c("N1", "N2", "kind2"))
```

```
## Linear mixed model fit by REML ['lmerModLmerTest']
## Formula: score ~ N1 + N2 + kind2 + (1 | Sent)
##    Data: rubert_comp_scores_gen
## REML criterion at convergence: 825.0781
## Random effects:
##  Groups   Name        Std.Dev.
##  Sent     (Intercept) 1.5438
##  Residual             0.9962
## Number of obs: 256, groups:  Sent, 32
## Fixed Effects:
## (Intercept)          N1S          N2S    kind2gram  kind2ungram
##      0.2720      -0.9213      -0.3804       2.2733      -0.2960
```

```
mod5KR_rubert_gen
```

```
## # A tibble: 1 x 12
##   N1_KR_stat N2_KR_stat kind2_KR_stat N1_KR_ddf N2_KR_ddf kind2_KR_ddf
##        <dbl>      <dbl>         <dbl>     <dbl>     <dbl>        <dbl>
## 1       54.7       9.33          190.      220.      220.         220.
## # i 6 more variables: N1_KR_p.value <dbl>, N2_KR_p.value <dbl>,
## #   kind2_KR_p.value <dbl>, N1_KR_mod_is_singular <lgl>,
## #   N2_KR_mod_is_singular <lgl>, kind2_KR_mod_is_singular <lgl>
```

Group 2 is genitive, which is syncretic like Nom.Pl = Gen.Sg (!= Acc.Pl)

```
rubert_comp_scores %>%
  filter(Group == 2) %>%
  mutate(
    kind3=as.factor(case_when(
      N1 == Pred2 ~ "gram",
      (N1 != Pred2) & (N2 != Pred2) ~ "distr",
      TRUE ~ "ungram"
    )),
    .after=Pred2
  ) ->
  rubert_comp_scores_gen_syncr

mod5_rubert_gen_syncr = lmer(score ~ N1 + N2 + kind3 + (1 | Sent), data=rubert_comp_scores_gen_syncr)
summary(mod5_rubert_gen_syncr)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: score ~ N1 + N2 + kind3 + (1 | Sent)
##    Data: rubert_comp_scores_gen_syncr
##
## REML criterion at convergence: 825.1
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.74049 -0.58231 -0.01764  0.58483  2.53483
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
```

```
##  Sent     (Intercept) 2.3832   1.5438
##  Residual             0.9924   0.9962
## Number of obs: 256, groups:  Sent, 32
##
## Fixed effects:
##             Estimate Std. Error        df t value Pr(>|t|)
## (Intercept)  -0.02402    0.31263  47.82255  -0.077  0.93909
## N1S          -0.92134    0.12453 220.00000  -7.399 2.87e-12 ***
## N2S          -0.38035    0.12453 220.00000  -3.054  0.00253 **
## kind3gram     2.56928    0.15251 220.00000  16.846  < 2e-16 ***
## kind3ungram   0.29601    0.17611 220.00000   1.681  0.09421 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) N1S    N2S    knd3gr
## N1S         -0.199
## N2S         -0.199  0.000
## kind3gram   -0.325  0.000  0.000
## kind3ungram -0.282  0.000  0.000  0.577
```

```
info5_rubert_gen_syncr <- get_full_lmer_info(mod5_rubert_gen_syncr, summary(mod5_rubert_gen_syncr))

mod5KR_rubert_gen_syncr <- test_lmer_signif_KR(mod5_rubert_gen_syncr, c("N1", "N2", "kind3"))
```

```
## Linear mixed model fit by REML ['lmerModLmerTest']
## Formula: score ~ N1 + N2 + kind3 + (1 | Sent)
##    Data: rubert_comp_scores_gen_syncr
## REML criterion at convergence: 825.0781
## Random effects:
##  Groups   Name        Std.Dev.
##  Sent     (Intercept) 1.5438
##  Residual             0.9962
## Number of obs: 256, groups:  Sent, 32
## Fixed Effects:
## (Intercept)          N1S          N2S    kind3gram  kind3ungram
##    -0.02402     -0.92134     -0.38035      2.56928      0.29601
```

```
mod5KR_rubert_gen_syncr
```

```
## # A tibble: 1 x 12
##   N1_KR_stat N2_KR_stat kind3_KR_stat N1_KR_ddf N2_KR_ddf kind3_KR_ddf
##        <dbl>      <dbl>         <dbl>     <dbl>     <dbl>        <dbl>
## 1       54.7       9.33          190.      220.      220.         220.
## # i 6 more variables: N1_KR_p.value <dbl>, N2_KR_p.value <dbl>,
## #   kind3_KR_p.value <dbl>, N1_KR_mod_is_singular <lgl>,
## #   N2_KR_mod_is_singular <lgl>, kind3_KR_mod_is_singular <lgl>
```

```
gpt_comp_scores %>%
  filter(Group==1) ->
  gpt_comp_scores_acc

mod5_gpt_acc = lmer(score ~ N1 + N2 + kind2 + (1 | Sent), data=gpt_comp_scores_acc)
summary(mod5_gpt_acc)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
```

```
## lmerModLmerTest]
## Formula: score ~ N1 + N2 + kind2 + (1 | Sent)
##    Data: gpt_comp_scores_acc
##
## REML criterion at convergence: 839.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.9518 -0.6429  0.1400  0.5528  2.4902
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Sent     (Intercept) 1.341    1.158
##  Residual             1.142    1.069
## Number of obs: 256, groups:  Sent, 32
##
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)  -1.3492     0.2620  65.8381  -5.149 2.57e-06 ***
## N1S          -0.6499     0.1336 220.0000  -4.866 2.17e-06 ***
## N2S          -0.3351     0.1336 220.0000  -2.509   0.0128 *
## kind2gram     3.3563     0.1636 220.0000  20.517  < 2e-16 ***
## kind2ungram  -0.8146     0.1889 220.0000  -4.313 2.44e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) N1S    N2S    knd2gr
## N1S         -0.255
## N2S         -0.255  0.000
## kind2gram   -0.416  0.000  0.000
## kind2ungram -0.360  0.000  0.000  0.577
```

```
mod5_gpt_acc_coeffs = make_coeffs_table(mod5_gpt_acc, "rugpt", "acc")
all_coeffs = rbind(all_coeffs, mod5_gpt_acc_coeffs)

info5_gpt_acc <- get_full_lmer_info(mod5_gpt_acc, summary(mod5_gpt_acc))

mod5KR_gpt_acc <- test_lmer_signif_KR(mod5_gpt_acc, c("N1", "N2", "kind2"))
```

```
## Linear mixed model fit by REML ['lmerModLmerTest']
## Formula: score ~ N1 + N2 + kind2 + (1 | Sent)
##    Data: gpt_comp_scores_acc
## REML criterion at convergence: 839.6458
## Random effects:
##  Groups   Name        Std.Dev.
##  Sent     (Intercept) 1.158
##  Residual             1.069
## Number of obs: 256, groups:  Sent, 32
## Fixed Effects:
## (Intercept)          N1S          N2S    kind2gram  kind2ungram
##     -1.3492      -0.6499      -0.3351       3.3563      -0.8146
```

```
mod5KR_gpt_acc
```

```
## # A tibble: 1 x 12
##   N1_KR_stat N2_KR_stat kind2_KR_stat N1_KR_ddf N2_KR_ddf kind2_KR_ddf
##        <dbl>      <dbl>         <dbl>     <dbl>     <dbl>        <dbl>
## 1       23.7       6.30          406.      220.      220.         220.
## # i 6 more variables: N1_KR_p.value <dbl>, N2_KR_p.value <dbl>,
## #   kind2_KR_p.value <dbl>, N1_KR_mod_is_singular <lgl>,
## #   N2_KR_mod_is_singular <lgl>, kind2_KR_mod_is_singular <lgl>
```

```
gpt_comp_scores %>%
  filter(Group==2) ->
  gpt_comp_scores_gen

mod5_gpt_gen = lmer(score ~ N1 + N2 + kind2 + (1 | Sent), data=gpt_comp_scores_gen)
summary(mod5_gpt_gen)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: score ~ N1 + N2 + kind2 + (1 | Sent)
##    Data: gpt_comp_scores_gen
##
## REML criterion at convergence: 766.4
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.68803 -0.58288  0.04961  0.63648  2.27890
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Sent     (Intercept) 0.8044   0.8969
##  Residual             0.8764   0.9362
## Number of obs: 256, groups:  Sent, 32
##
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)  -2.3098     0.2137  75.4708 -10.807  < 2e-16 ***
## N1S          -0.4593     0.1170 220.0000  -3.925 0.000116 ***
## N2S           0.2513     0.1170 220.0000   2.147 0.032871 *
## kind2gram     3.7490     0.1433 220.0000  26.158  < 2e-16 ***
## kind2ungram  -0.2626     0.1655 220.0000  -1.587 0.114007
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) N1S    N2S    knd2gr
## N1S         -0.274
## N2S         -0.274  0.000
## kind2gram   -0.447  0.000  0.000
## kind2ungram -0.387  0.000  0.000  0.577
```

```
mod5_gpt_gen_coeffs = make_coeffs_table(mod5_gpt_gen, "rugpt", "gen")
all_coeffs = rbind(all_coeffs, mod5_gpt_gen_coeffs)

info5_gpt_gen <- get_full_lmer_info(mod5_gpt_gen, summary(mod5_gpt_gen))
```

```
mod5KR_gpt_gen <- test_lmer_signif_KR(mod5_gpt_gen, c("N1", "N2", "kind2"))
```

```
## Linear mixed model fit by REML ['lmerModLmerTest']
## Formula: score ~ N1 + N2 + kind2 + (1 | Sent)
##    Data: gpt_comp_scores_gen
## REML criterion at convergence: 766.4437
## Random effects:
##  Groups   Name        Std.Dev.
##  Sent     (Intercept) 0.8969
##  Residual             0.9362
## Number of obs: 256, groups:  Sent, 32
## Fixed Effects:
## (Intercept)          N1S          N2S    kind2gram  kind2ungram
##     -2.3098      -0.4593       0.2513       3.7490      -0.2626
```

```
mod5KR_gpt_gen
```

```
## # A tibble: 1 x 12
##   N1_KR_stat N2_KR_stat kind2_KR_stat N1_KR_ddf N2_KR_ddf kind2_KR_ddf
##        <dbl>      <dbl>         <dbl>     <dbl>     <dbl>        <dbl>
## 1       15.4       4.61          551.      220.      220.         220.
## # i 6 more variables: N1_KR_p.value <dbl>, N2_KR_p.value <dbl>,
## #   kind2_KR_p.value <dbl>, N1_KR_mod_is_singular <lgl>,
## #   N2_KR_mod_is_singular <lgl>, kind2_KR_mod_is_singular <lgl>
```

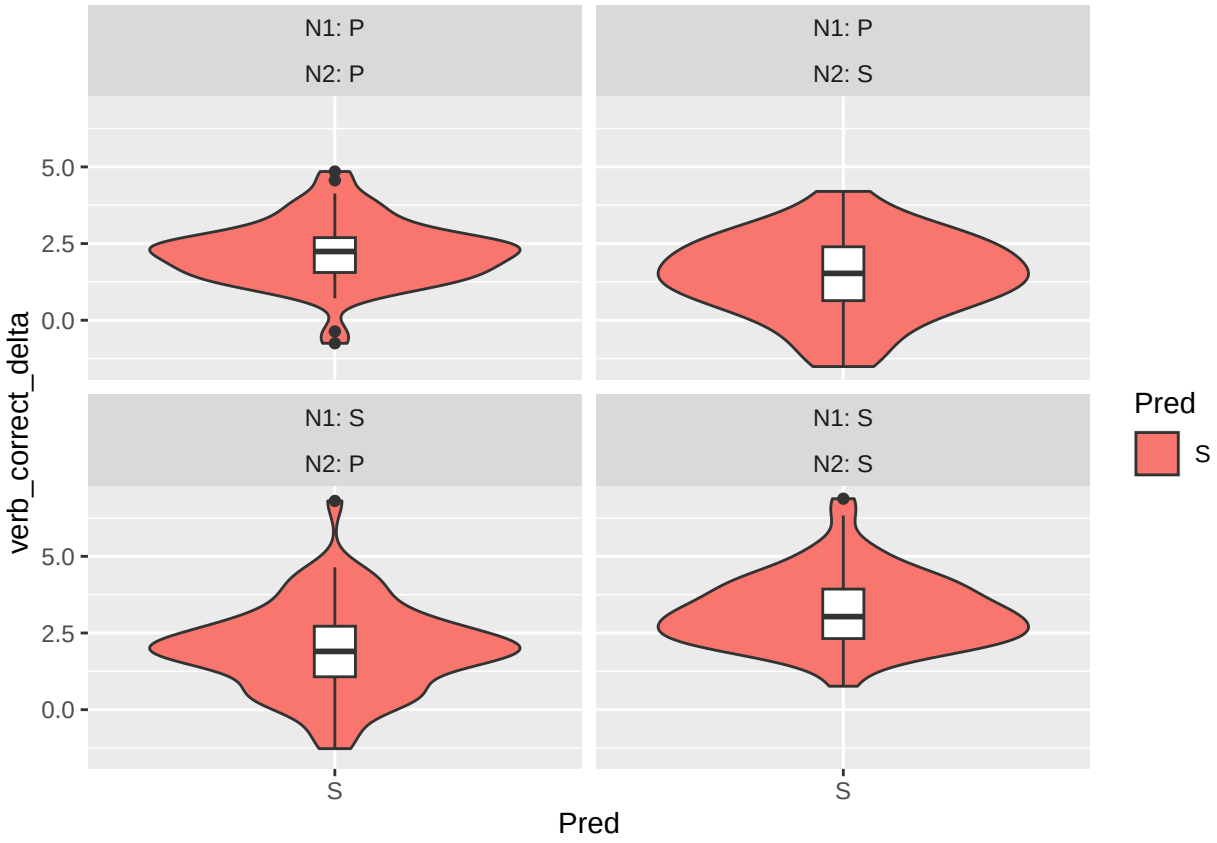
**other model plots**

```
rubert_comp %>%
  group_by(N1, N2, Pred) %>%
  summarise(
    across(verb_correct_delta,
           list(mean=mean, median=median)
    )
  )
```

```
## `summarise()` has grouped output by 'N1', 'N2'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 4 x 5
## # Groups:   N1, N2 [4]
##   N1    N2    Pred  verb_correct_delta_mean verb_correct_delta_median
##   <fct> <fct> <fct>                   <dbl>                     <dbl>
## 1 P     P     S                        2.20                      2.24
## 2 P     S     S                        1.46                      1.53
## 3 S     P     S                        1.98                      1.90
## 4 S     S     S                        3.22                      3.03
```

```
rubert_comp %>%
  ggplot(aes(x=Pred, y=verb_correct_delta, fill=Pred)) +
  geom_violin() +
  geom_boxplot(width=0.1, fill="white") +
  facet_wrap(N1 ~ N2, labeller = label_both)
```




**accuracy**

```
rubert_comp %>%
  mutate(
    verb_correct = verb_correct_delta > 0,
    # particip_correct = particip_correct_delta > 0,
    # pred_correct = pred_correct_delta > 0
  ) %>%
  summarise(
    n_verb_correct = sum(verb_correct),
    n_verb_wrong = length(verb_correct) - sum(verb_correct),
    # n_particip_correct = sum(particip_correct),
    # n_particip_wrong = length(particip_correct) - sum(particip_correct),
    # n_pred_correct = sum(pred_correct),
    # n_pred_wrong = length(pred_correct) - sum(pred_correct)
  )
```

```
## # A tibble: 1 x 2
##   n_verb_correct n_verb_wrong
##            <int>        <int>
## 1            241           15
```

```
rubert_comp %>%
  mutate(
    verb_correct = verb_correct_delta > 0,
  ) %>%
```

```
  group_by(Code) %>%
  summarise(
    n_verb_correct = sum(verb_correct),
    n_verb_wrong = length(verb_correct) - sum(verb_correct),
  )
```

```
## # A tibble: 4 x 3
##   Code  n_verb_correct n_verb_wrong
##   <fct>          <int>        <int>
## 1 S_P-P             62            2
## 2 S_P-S             55            9
## 3 S_S-P             60            4
## 4 S_S-S             64            0
```

```
rubert_comp %>%
  mutate(verb_correct = verb_correct_delta > 0) %>%
  group_by(N1, N2) %>%
  summarise(ratio_correct = mean(verb_correct))
```

```
## `summarise()` has grouped output by 'N1'. You can override using the `.groups`
## argument.
```

```
## # A tibble: 4 x 3
## # Groups:   N1 [2]
##   N1    N2    ratio_correct
##   <fct> <fct>         <dbl>
## 1 P     P             0.969
## 2 P     S             0.859
## 3 S     P             0.938
## 4 S     S             1
```

```
gpt_comp %>%
  mutate(verb_correct = verb_correct_delta > 0) %>%
  group_by(N1, N2) %>%
  summarise(ratio_correct = mean(verb_correct))
```

```
## `summarise()` has grouped output by 'N1'. You can override using the `.groups`
## argument.
```

```
## # A tibble: 4 x 3
## # Groups:   N1 [2]
##   N1    N2    ratio_correct
##   <fct> <fct>         <dbl>
## 1 P     P             1
## 2 P     S             0.953
## 3 S     P             1
## 4 S     S             1
```

```

mod <- lmer(verb_correct_delta ~ N1 + N2 +  (1 | Sent), data=rubert_comp)
summary(mod)
```

### Other regression models

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: verb_correct_delta ~ N1 + N2 + (1 | Sent)
```

```
##    Data: rubert_comp
##
## REML criterion at convergence: 856.5
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.58818 -0.51114  0.06534  0.67702  2.32133
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Sent     (Intercept) 0.4729   0.6877
##  Residual             1.3101   1.1446
## Number of obs: 256, groups:  Sent, 64
##
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)   1.7064     0.1508 170.4510  11.315  < 2e-16 ***
## N1S           0.7713     0.1431 190.0000   5.391 2.07e-07 ***
## N2S           0.2494     0.1431 190.0000   1.743   0.0829 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##     (Intr) N1S
## N1S -0.474
## N2S -0.474  0.000
```

```
test_lmer_signif_KR(mod, c("N1", "N2"))
```

```
## Linear mixed model fit by REML ['lmerModLmerTest']
## Formula: verb_correct_delta ~ N1 + N2 + (1 | Sent)
##    Data: rubert_comp
## REML criterion at convergence: 856.4772
## Random effects:
##  Groups   Name        Std.Dev.
##  Sent     (Intercept) 0.6877
##  Residual             1.1446
## Number of obs: 256, groups:  Sent, 64
## Fixed Effects:
## (Intercept)          N1S          N2S
##      1.7064       0.7713       0.2494
```

```
## # A tibble: 1 x 8
##   N1_KR_stat N2_KR_stat N1_KR_ddf N2_KR_ddf N1_KR_p.value N2_KR_p.value
##        <dbl>      <dbl>     <dbl>     <dbl>         <dbl>         <dbl>
## 1       29.1       3.04      190.      190.   0.000000207        0.0829
## # i 2 more variables: N1_KR_mod_is_singular <lgl>, N2_KR_mod_is_singular <lgl>
```

```
rubert_comp %>%
  filter(Group == 1) ->
  rubert_comp_acc

mod2 <- lmer(verb_correct_delta ~ N1 + N2 + Group + is_distr_different + (1 | Sent), data=rubert_comp)
summary(mod2)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
```

```
## lmerModLmerTest]
## Formula: verb_correct_delta ~ N1 + N2 + Group + is_distr_different + (1 |
##     Sent)
##    Data: rubert_comp
##
## REML criterion at convergence: 801.4
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.46944 -0.54193  0.02022  0.53707  2.48033
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Sent     (Intercept) 0.5239   0.7238
##  Residual             0.9851   0.9925
## Number of obs: 256, groups:  Sent, 64
##
## Fixed effects:
##                         Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)               1.5878     0.3632  74.2420   4.372 3.94e-05 ***
## N1S                       0.7713     0.1241 189.0000   6.217 3.18e-09 ***
## N2S                       0.2494     0.1241 189.0000   2.010   0.0458 *
## Group                     0.4090     0.2194  62.0000   1.864   0.0670 .
## is_distr_differentTRUE  -0.9900     0.1241 189.0000  -7.979 1.38e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) N1S    N2S    Group
## N1S         -0.171
## N2S         -0.171  0.000
## Group       -0.906  0.000  0.000
## is_dst_TRUE -0.171  0.000  0.000  0.000
```

```
info2 <- get_full_lmer_info(mod2, summary(mod2))
```

```
mod2KR = test_lmer_signif_KR(mod2, c("N1", "N2", "is_distr_different"))
```

```
## Linear mixed model fit by REML ['lmerModLmerTest']
## Formula: verb_correct_delta ~ N1 + N2 + Group + is_distr_different + (1 |
##     Sent)
##    Data: rubert_comp
## REML criterion at convergence: 801.4191
## Random effects:
##  Groups   Name        Std.Dev.
##  Sent     (Intercept) 0.7238
##  Residual             0.9925
## Number of obs: 256, groups:  Sent, 64
## Fixed Effects:
##            (Intercept)                    N1S                    N2S
##                 1.5878                 0.7713                 0.2494
##                  Group  is_distr_differentTRUE
##                 0.4090                -0.9900
```

```
mod2KR
```

```
## # A tibble: 1 x 12
##   N1_KR_stat N2_KR_stat is_distr_different_KR_stat N1_KR_ddf N2_KR_ddf
##        <dbl>      <dbl>                      <dbl>     <dbl>     <dbl>
## 1       38.6       4.04                       63.7      189.      189.
## # i 7 more variables: is_distr_different_KR_ddf <dbl>, N1_KR_p.value <dbl>,
## #   N2_KR_p.value <dbl>, is_distr_different_KR_p.value <dbl>,
## #   N1_KR_mod_is_singular <lgl>, N2_KR_mod_is_singular <lgl>,
## #   is_distr_different_KR_mod_is_singular <lgl>
```

```
cbind(info2, mod2KR) %>%
  write_csv("bert_lmer_results.csv")
```

```
mod2_gpt = lmer(verb_correct_delta ~ N1 + N2 + is_distr_different + (1 | Sent), data=gpt_comp)
summary(mod2_gpt)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: verb_correct_delta ~ N1 + N2 + is_distr_different + (1 | Sent)
##    Data: gpt_comp
##
## REML criterion at convergence: 813.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.9904 -0.5366  0.0123  0.6202  2.7580
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Sent     (Intercept) 0.271    0.5206
##  Residual             1.168    1.0807
## Number of obs: 256, groups:  Sent, 64
##
## Fixed effects:
##                        Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)              3.1296     0.1499 227.7609  20.872  < 2e-16 ***
## N1S                      1.8719     0.1351 189.0000  13.857  < 2e-16 ***
## N2S                      0.1566     0.1351 189.0000   1.159    0.248
## is_distr_differentTRUE  -0.6437     0.1351 189.0000  -4.765 3.75e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) N1S    N2S
## N1S         -0.450
## N2S         -0.450  0.000
## is_dst_TRUE -0.450  0.000  0.000
```

```
info2_gpt <- get_full_lmer_info(mod2_gpt, summary(mod2_gpt))

mod2KR_gpt <- test_lmer_signif_KR(mod2_gpt, c("N1", "N2", "is_distr_different"))
```

```
## Linear mixed model fit by REML ['lmerModLmerTest']
## Formula: verb_correct_delta ~ N1 + N2 + is_distr_different + (1 | Sent)
##    Data: gpt_comp
```

```
## REML criterion at convergence: 813.6286
## Random effects:
##  Groups   Name        Std.Dev.
##  Sent     (Intercept) 0.5206
##  Residual             1.0807
## Number of obs: 256, groups:  Sent, 64
## Fixed Effects:
##            (Intercept)                     N1S                     N2S
##                 3.1296                  1.8719                  0.1566
## is_distr_differentTRUE
##                -0.6437
```

```
## boundary (singular) fit: see help('isSingular')
```

```
mod2KR_gpt
```

```
## # A tibble: 1 x 12
##   N1_KR_stat N2_KR_stat is_distr_different_KR_stat N1_KR_ddf N2_KR_ddf
##        <dbl>      <dbl>                      <dbl>     <dbl>     <dbl>
## 1       192.       1.34                       22.7      189.      189.
## # i 7 more variables: is_distr_different_KR_ddf <dbl>, N1_KR_p.value <dbl>,
## #   N2_KR_p.value <dbl>, is_distr_different_KR_p.value <dbl>,
## #   N1_KR_mod_is_singular <lgl>, N2_KR_mod_is_singular <lgl>,
## #   is_distr_different_KR_mod_is_singular <lgl>
```

```
cbind(info2_gpt, mod2KR_gpt) %>%
  write_csv("gpt_lmer_results.csv")
```

```
mod3 <- lmer(verb_correct_delta ~ N1 * N2 +  (1 | Sent), data=rubert_comp)
summary(mod3)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: verb_correct_delta ~ N1 * N2 + (1 | Sent)
##    Data: rubert_comp
##
## REML criterion at convergence: 802.3
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.47555 -0.56918 -0.00154  0.55265  2.50507
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Sent     (Intercept) 0.5542   0.7444
##  Residual             0.9851   0.9925
## Number of obs: 256, groups:  Sent, 64
##
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)   2.2014     0.1551 181.4481  14.195  < 2e-16 ***
## N1S          -0.2187     0.1755 189.0000  -1.246    0.214
## N2S          -0.7405     0.1755 189.0000  -4.221 3.78e-05 ***
## N1S:N2S       1.9800     0.2481 189.0000   7.979 1.38e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Correlation of Fixed Effects:
##         (Intr) N1S    N2S
## N1S     -0.566
## N2S     -0.566  0.500
## N1S:N2S  0.400 -0.707 -0.707
```

```
mod4 <- lmer(verb_correct_delta2 ~ N1 + N2 + is_distr_different + (1 | Sent), data=rubert_comp)
summary(mod4)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: verb_correct_delta2 ~ N1 + N2 + is_distr_different + (1 | Sent)
##    Data: rubert_comp
##
## REML criterion at convergence: 845.7
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.99313 -0.57707  0.01487  0.65701  2.27658
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Sent     (Intercept) 3.0632   1.750
##  Residual             0.7708   0.878
## Number of obs: 256, groups:  Sent, 64
##
## Fixed effects:
##                         Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)               1.9783     0.2448  86.4498   8.083 3.49e-12 ***
## N1S                      -0.4191     0.1097 189.0000  -3.819 0.000181 ***
## N2S                      -0.3373     0.1097 189.0000  -3.074 0.002425 **
## is_distr_differentTRUE   -0.3125     0.1097 189.0000  -2.847 0.004897 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) N1S    N2S
## N1S         -0.224
## N2S         -0.224  0.000
## is_dst_TRUE -0.224  0.000  0.000
```

### Humans

```
humans
```

```
## # A tibble: 28,796 x 10
##     part    rt sent  group pred  subj    reg  snum  anum   err
##    <dbl> <dbl> <chr> <dbl> <chr> <chr> <dbl> <dbl> <dbl> <dbl>
##  1    33  180. S1        1 S     S-S       1     0     0     0
##  2    20  184  S1        1 S     P-S       1     1     0     1
##  3    15  188. S1        1 P     S-P       1     0     1     1
##  4     4  195. S1        1 S     S-S       1     0     0     0
##  5    27  203. S1        1 S     P-P       1     1     1     1
##  6     1  205. S1        1 S     S-S       1     0     0     0
```

```
##  7     2  205. S1        1 S     S-S       1     0     0     0
##  8    14  221. S1        1 P     S-P       1     0     1     1
##  9    39  234. S1        1 S     P-P       1     1     1     1
## 10    36  242. S1        1 P     S-P       1     0     1     1
## # i 28,786 more rows
```

```
humans %>% pull(part) %>% unique() %>% length()
```

```
## [1] 40
```

```
humans_word4 = humans %>% filter(reg == 4)
humans_word4
```

```
## # A tibble: 3,200 x 10
##     part    rt sent  group pred  subj    reg  snum  anum   err
##    <dbl> <dbl> <chr> <dbl> <chr> <chr> <dbl> <dbl> <dbl> <dbl>
##  1    33  163. S1        1 S     S-S       4     0     0     0
##  2     4  178. S1        1 S     S-S       4     0     0     0
##  3    15  190. S1        1 P     S-P       4     0     1     1
##  4    26  192. S1        1 S     P-P       4     1     1     1
##  5     2  197. S1        1 S     S-S       4     0     0     0
##  6    28  198. S1        1 S     P-P       4     1     1     1
##  7    27  203. S1        1 S     P-P       4     1     1     1
##  8    20  205. S1        1 S     P-S       4     1     0     1
##  9    35  218. S1        1 S     S-P       4     0     1     0
## 10    11  233. S1        1 S     S-P       4     0     1     0
## # i 3,190 more rows
```

```
humans_word5 = humans %>% filter(reg == 5)
humans_word5
```

```
## # A tibble: 3,200 x 10
##     part    rt sent  group pred  subj    reg  snum  anum   err
##    <dbl> <dbl> <chr> <dbl> <chr> <chr> <dbl> <dbl> <dbl> <dbl>
##  1    33  196. S1        1 S     S-S       5     0     0     0
##  2     2  204. S1        1 S     S-S       5     0     0     0
##  3     4  210. S1        1 S     S-S       5     0     0     0
##  4    20  216. S1        1 S     P-S       5     1     0     1
##  5    15  227. S1        1 P     S-P       5     0     1     1
##  6    32  230. S1        1 P     P-P       5     1     1     0
##  7    26  232. S1        1 S     P-P       5     1     1     1
##  8    37  240  S1        1 S     P-S       5     1     0     1
##  9    27  242. S1        1 S     P-P       5     1     1     1
## 10    39  251. S1        1 S     P-P       5     1     1     1
## # i 3,190 more rows
```

There are 80 sentences in the experiment where response time was measured

```
humans %>% pull(sent) %>% unique()
```

```
##  [1] "S1"  "S10" "S11" "S12" "S13" "S14" "S15" "S16" "S17" "S18" "S19" "S2"
## [13] "S20" "S21" "S22" "S23" "S24" "S25" "S26" "S27" "S28" "S29" "S3"  "S30"
## [25] "S31" "S32" "S33" "S34" "S35" "S36" "S37" "S38" "S39" "S4"  "S40" "S5"
## [37] "S6"  "S7"  "S8"  "S9"  "S41" "S42" "S43" "S44" "S45" "S46" "S47" "S48"
## [49] "S49" "S50" "S51" "S52" "S53" "S54" "S55" "S56" "S57" "S58" "S59" "S60"
## [61] "S61" "S62" "S63" "S64" "S65" "S66" "S67" "S68" "S69" "S70" "S71" "S72"
## [73] "S73" "S74" "S75" "S76" "S77" "S78" "S79" "S80"
```

```r
humans_word4 %>%
  filter(sent=="S1") %>%
  select(part, sent, subj, pred, rt) %>%
  arrange(part, sent,  subj, pred)
```

```
## # A tibble: 40 x 5
##     part sent  subj  pred     rt
##    <dbl> <chr> <chr> <chr> <dbl>
##  1     1 S1    S-S   S      305.
##  2     2 S1    S-S   S      197.
##  3     3 S1    S-S   S      272.
##  4     4 S1    S-S   S      178.
##  5     5 S1    S-S   P      474.
##  6     6 S1    S-S   P      358.
##  7     7 S1    S-S   P      488.
##  8     8 S1    S-S   P      368.
##  9     9 S1    S-P   S      362.
## 10    10 S1    S-P   S      376.
## # i 30 more rows
```

```r
humans_word4 %>%
  select(part, sent, subj, pred, rt) %>%
  # group_by(part, sent, subj) %>%
  arrange(part, sent,  subj) %>%
  pivot_wider(names_from = pred, values_from = rt)
```

```
## # A tibble: 3,200 x 5
##     part sent  subj      S     P
##    <dbl> <chr> <chr> <dbl> <dbl>
##  1     1 S1    S-S    305.   NA
##  2     1 S10   S-S     NA   277.
##  3     1 S11   S-P    178    NA
##  4     1 S12   S-P     NA   445.
##  5     1 S13   P-S    282.   NA
##  6     1 S14   P-S     NA   317
##  7     1 S15   P-P    261.   NA
##  8     1 S16   P-P     NA   429.
##  9     1 S17   S-S    332    NA
## 10     1 S18   S-S     NA   259.
## # i 3,190 more rows
```

```r
humans %>%
  separate_wider_delim(
      "subj", "-", names = c("N1", "N2")
  ) %>%
  mutate(is_distr_different = N2 != N1) %>%
  mutate(
    kind=as.factor(case_when(
      N1 == pred ~ "gram",
      (N1 != pred) & (N2 == pred) ~ "distr",
      TRUE ~ "ungram"
    )),
    is_correct = N1 == pred,
    is_distractor = N2 == pred,
    .after=pred
```

```
  ) -> humans

humans
```

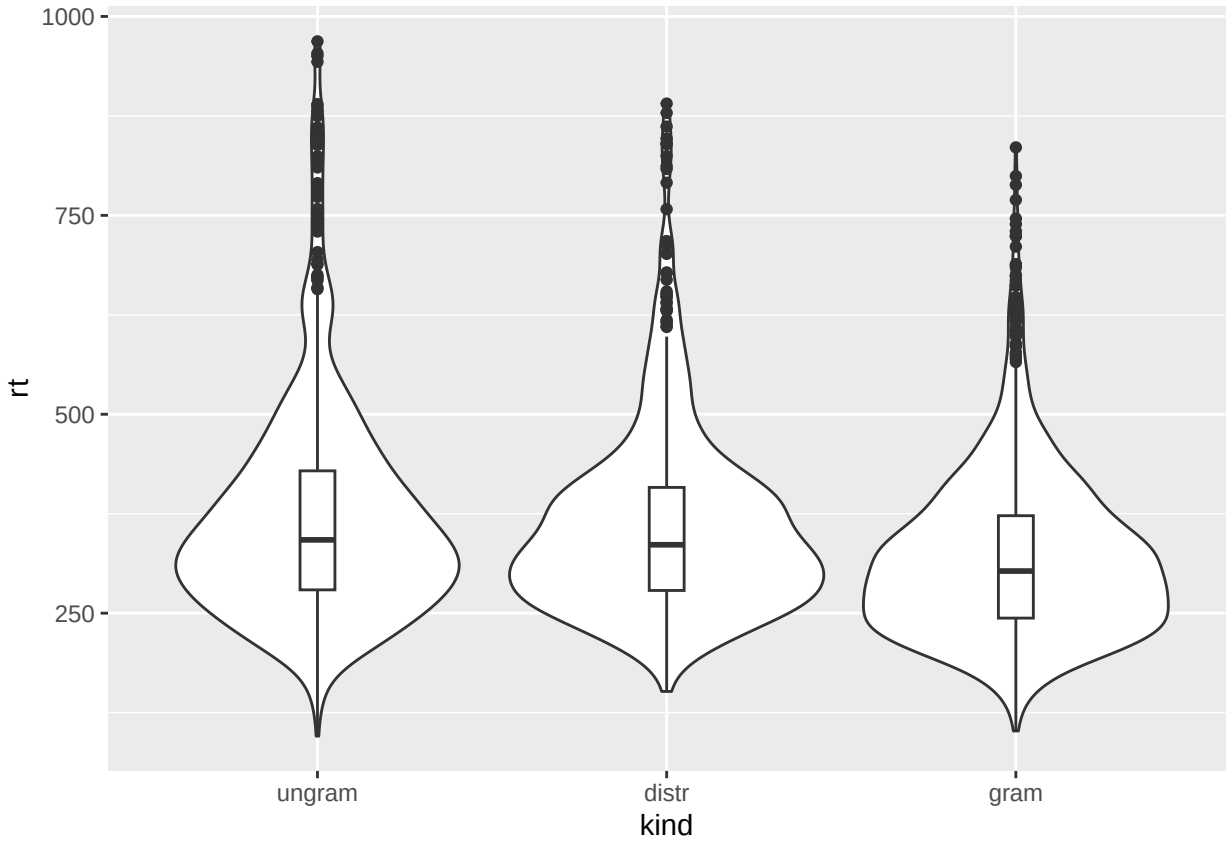
```
## # A tibble: 28,796 x 15
##     part    rt sent  group pred  kind   is_correct is_distractor N1    N2
##    <dbl> <dbl> <chr> <dbl> <chr> <fct>  <lgl>      <lgl>         <chr> <chr>
##  1    33  180. S1        1 S     gram   TRUE       TRUE          S     S
##  2    20  184  S1        1 S     distr  FALSE      TRUE          P     S
##  3    15  188. S1        1 P     distr  FALSE      TRUE          S     P
##  4     4  195. S1        1 S     gram   TRUE       TRUE          S     S
##  5    27  203. S1        1 S     ungram FALSE      FALSE         P     P
##  6     1  205. S1        1 S     gram   TRUE       TRUE          S     S
##  7     2  205. S1        1 S     gram   TRUE       TRUE          S     S
##  8    14  221. S1        1 P     distr  FALSE      TRUE          S     P
##  9    39  234. S1        1 S     ungram FALSE      FALSE         P     P
## 10    36  242. S1        1 P     distr  FALSE      TRUE          S     P
## # i 28,786 more rows
## # i 5 more variables: reg <dbl>, snum <dbl>, anum <dbl>, err <dbl>,
## #   is_distr_different <lgl>
```

The effect is visible on word 5

```
humans %>%
  filter(reg == 5) %>%
  mutate(kind = factor(kind, levels=c("ungram", "distr", "gram"))) %>%
  group_by(kind) %>%
  summarise(mean=mean(rt, na.rm=TRUE), median=median(rt, na.rm=TRUE), sd=sd(rt, na.rm=TRUE))
```

```
## # A tibble: 3 x 4
##   kind    mean median    sd
##   <fct>  <dbl>  <dbl> <dbl>
## 1 ungram  372.   342.  138.
## 2 distr   360.   336.  118.
## 3 gram    319.   303.  104.
```

```
humans %>%
  filter(reg == 5) %>%
  mutate(kind = factor(kind, levels=c("ungram", "distr", "gram"))) %>%
  ggplot(aes(x=kind, y=rt)) +
    geom_violin() +
    geom_boxplot(width=0.1, fill="white")
```




```
    # facet_wrap(N1 ~ N2, labeller = label_both)
```

```
humans_word5 = humans %>% filter(reg == 5)

# humans_word5_acc = humans_word5 %>% filter(group==1)

lmer(rt ~ N1 + N2 + kind + (1 | sent) + (1|part),
     data=humans_word5) ->
  mod5_humans

summary(mod5_humans)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: rt ~ N1 + N2 + kind + (1 | sent) + (1 | part)
##    Data: humans_word5
##
## REML criterion at convergence: 37806.7
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.2797 -0.5642 -0.1679  0.3041  6.1699
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  sent     (Intercept)  250.8   15.84
##  part     (Intercept) 4023.5   63.43
```

```
##  Residual             9402.3   96.97
## Number of obs: 3140, groups:  sent, 80; part, 40
##
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)  367.834     11.031   53.709  33.346  < 2e-16 ***
## N1S           -9.529      3.462 3018.920  -2.752  0.00595 **
## N2S           -3.673      3.462 3019.435  -1.061  0.28877
## kindgram     -41.281      4.239 3019.113  -9.739  < 2e-16 ***
## kindungram    11.097      4.892 3018.883   2.268  0.02339 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) N1S    N2S    kndgrm
## N1S         -0.157
## N2S         -0.156 -0.001
## kindgram    -0.255 -0.003 -0.002
## kindungram -0.222  0.002  0.000  0.577
```

```
mod5_humans_coeffs = make_coeffs_table(mod5_humans, "humans", "both")
all_coeffs = rbind(all_coeffs, mod5_humans_coeffs)

info5_humans <- get_full_lmer_info(mod5_humans, summary(mod5_humans))

mod5KR_humans <- test_lmer_signif_KR(mod5_humans, c("N1", "N2", "kind"))
```

```
## Linear mixed model fit by REML ['lmerModLmerTest']
## Formula: rt ~ N1 + N2 + kind + (1 | sent) + (1 | part)
##    Data: humans_word5
## REML criterion at convergence: 37806.73
## Random effects:
##  Groups   Name        Std.Dev.
##  sent     (Intercept) 15.84
##  part     (Intercept) 63.43
##  Residual             96.97
## Number of obs: 3140, groups:  sent, 80; part, 40
## Fixed Effects:
## (Intercept)          N1S          N2S     kindgram   kindungram
##     367.834       -9.529       -3.673      -41.281       11.097
```

```
mod5KR_humans
```

```
## # A tibble: 1 x 12
##   N1_KR_stat N2_KR_stat kind_KR_stat N1_KR_ddf N2_KR_ddf kind_KR_ddf
##        <dbl>      <dbl>        <dbl>     <dbl>     <dbl>       <dbl>
## 1       7.58       1.13         94.0     3019.     3019.       3019.
## # i 6 more variables: N1_KR_p.value <dbl>, N2_KR_p.value <dbl>,
## #   kind_KR_p.value <dbl>, N1_KR_mod_is_singular <lgl>,
## #   N2_KR_mod_is_singular <lgl>, kind_KR_mod_is_singular <lgl>
```

```
humans_word5_acc = humans %>% filter(reg == 5) %>% filter(group == 1)

lmer(rt ~ N1 + N2 + kind + (1 | sent) + (1|part),
     data=humans_word5_acc) ->
  mod5_humans_acc
```

```r
summary(mod5_humans_acc)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: rt ~ N1 + N2 + kind + (1 | sent) + (1 | part)
##    Data: humans_word5_acc
##
## REML criterion at convergence: 18934.4
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.5700 -0.5697 -0.1662  0.2795  6.0623
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  sent     (Intercept)  137.8   11.74
##  part     (Intercept) 3847.8   62.03
##  Residual             9581.1   97.88
## Number of obs: 1569, groups:  sent, 40; part, 40
##
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)  356.352     11.678   68.580  30.515  < 2e-16 ***
## N1S           -9.217      4.944 1486.797  -1.864   0.0625 .
## N2S            1.805      4.944 1487.083   0.365   0.7151
## kindgram     -33.745      6.060 1487.553  -5.569 3.04e-08 ***
## kindungram    30.316      6.998 1487.726   4.332 1.58e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) N1S    N2S    kndgrm
## N1S         -0.212
## N2S         -0.211 -0.001
## kindgram    -0.346 -0.002 -0.001
## kindungram -0.300  0.001 -0.001  0.578
```

```r
mod5_humans_acc_coeffs = make_coeffs_table(mod5_humans_acc, "humans", "acc")
all_coeffs = rbind(all_coeffs, mod5_humans_acc_coeffs)

info5_humans_acc <- get_full_lmer_info(mod5_humans_acc, summary(mod5_humans_acc))

mod5KR_humans_acc <- test_lmer_signif_KR(mod5_humans_acc, c("N1", "N2", "kind"))
```

```
## Linear mixed model fit by REML ['lmerModLmerTest']
## Formula: rt ~ N1 + N2 + kind + (1 | sent) + (1 | part)
##    Data: humans_word5_acc
## REML criterion at convergence: 18934.37
## Random effects:
##  Groups   Name        Std.Dev.
##  sent     (Intercept) 11.74
##  part     (Intercept) 62.03
##  Residual             97.88
## Number of obs: 1569, groups:  sent, 40; part, 40
```

```
## Fixed Effects:
## (Intercept)          N1S          N2S     kindgram   kindungram
##     356.352       -9.217        1.805      -33.745       30.316
```

```
mod5KR_humans_acc
```

```
## # A tibble: 1 x 12
##   N1_KR_stat N2_KR_stat kind_KR_stat N1_KR_ddf N2_KR_ddf kind_KR_ddf
##        <dbl>      <dbl>        <dbl>     <dbl>     <dbl>       <dbl>
## 1       3.48      0.133         58.3     1487.     1487.       1488.
## # i 6 more variables: N1_KR_p.value <dbl>, N2_KR_p.value <dbl>,
## #   kind_KR_p.value <dbl>, N1_KR_mod_is_singular <lgl>,
## #   N2_KR_mod_is_singular <lgl>, kind_KR_mod_is_singular <lgl>
```

```
humans_word5_gen = humans %>% filter(reg == 5) %>% filter(group == 2)

# humans_word5_acc = humans_word5 %>% filter(group==1)

lmer(rt ~ N1 + N2 + kind + (1 | sent) + (1|part),
     data=humans_word5_gen) ->
  mod5_humans_gen

summary(mod5_humans_gen)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: rt ~ N1 + N2 + kind + (1 | sent) + (1 | part)
##    Data: humans_word5_gen
##
## REML criterion at convergence: 18907.3
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.2013 -0.5369 -0.1729  0.2861  5.6083
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  sent     (Intercept)  383.8   19.59
##  part     (Intercept) 4216.2   64.93
##  Residual             9122.2   95.51
## Number of obs: 1571, groups:  sent, 40; part, 40
##
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)  379.326     12.236   69.223  31.000  < 2e-16 ***
## N1S           -9.888      4.822 1488.995  -2.050   0.0405 *
## N2S           -9.199      4.822 1489.151  -1.908   0.0566 .
## kindgram     -48.871      5.898 1488.614  -8.286 2.57e-16 ***
## kindungram    -7.920      6.802 1488.313  -1.164   0.2444
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##            (Intr) N1S    N2S    kndgrm
## N1S        -0.197
```

```
## N2S        -0.196 -0.001
## kindgram   -0.319 -0.003 -0.003
## kindungram -0.278  0.003  0.001  0.576
```

```
mod5_humans_gen_coeffs = make_coeffs_table(mod5_humans_gen, "humans", "gen")
all_coeffs = rbind(all_coeffs, mod5_humans_gen_coeffs)

info5_humans_gen <- get_full_lmer_info(mod5_humans_gen, summary(mod5_humans_gen))

mod5KR_humans_gen <- test_lmer_signif_KR(mod5_humans_gen, c("N1", "N2", "kind"))
```

```
## Linear mixed model fit by REML ['lmerModLmerTest']
## Formula: rt ~ N1 + N2 + kind + (1 | sent) + (1 | part)
##    Data: humans_word5_gen
## REML criterion at convergence: 18907.35
## Random effects:
##  Groups   Name        Std.Dev.
##  sent     (Intercept) 19.59
##  part     (Intercept) 64.93
##  Residual             95.51
## Number of obs: 1571, groups:  sent, 40; part, 40
## Fixed Effects:
## (Intercept)          N1S          N2S     kindgram   kindungram
##     379.326       -9.888       -9.199      -48.871       -7.920
```

```
mod5KR_humans_gen
```

```
## # A tibble: 1 x 12
##   N1_KR_stat N2_KR_stat kind_KR_stat N1_KR_ddf N2_KR_ddf kind_KR_ddf
##        <dbl>      <dbl>        <dbl>     <dbl>     <dbl>       <dbl>
## 1       4.20       3.64         44.1     1489.     1489.       1489.
## # i 6 more variables: N1_KR_p.value <dbl>, N2_KR_p.value <dbl>,
## #   kind_KR_p.value <dbl>, N1_KR_mod_is_singular <lgl>,
## #   N2_KR_mod_is_singular <lgl>, kind_KR_mod_is_singular <lgl>
```

There is no effect on word 4 however (these results are like Slioussar's)

```
humans_word4 = humans %>% filter(reg == 4)

# humans_word5_acc = humans_word5 %>% filter(group==1)

lmer(rt ~ N1 + N2 + kind + (1 | sent) + (1|part),
     data=humans_word4) ->
  mod5_humans_word4

summary(mod5_humans_word4)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: rt ~ N1 + N2 + kind + (1 | sent) + (1 | part)
##    Data: humans_word4
##
## REML criterion at convergence: 36897.5
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.8362 -0.6079 -0.1438  0.3912  5.8149
```

```
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  sent     (Intercept)  302.7   17.40
##  part     (Intercept) 3029.9   55.04
##  Residual             6774.0   82.30
## Number of obs: 3148, groups:  sent, 80; part, 40
##
## Fixed effects:
##              Estimate Std. Error        df t value Pr(>|t|)
## (Intercept)  315.6727     9.6178   54.8321  32.822   <2e-16 ***
## N1S           -2.3076     2.9348 3026.2195  -0.786    0.432
## N2S           -0.9150     2.9347 3025.9793  -0.312    0.755
## kindgram       0.5244     3.5981 3025.8050   0.146    0.884
## kindungram     4.9255     4.1584 3026.0617   1.184    0.236
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) N1S    N2S    kndgrm
## N1S         -0.153
## N2S         -0.152  0.001
## kindgram    -0.250  0.001 -0.002
## kindungram -0.217  0.003  0.000  0.579
```

```
info5_humans_word4 <- get_full_lmer_info(mod5_humans_word4, summary(mod5_humans_word4))

mod5KR_humans_word4 <- test_lmer_signif_KR(mod5_humans_word4, c("N1", "N2", "kind"))
```

```
## Linear mixed model fit by REML ['lmerModLmerTest']
## Formula: rt ~ N1 + N2 + kind + (1 | sent) + (1 | part)
##    Data: humans_word4
## REML criterion at convergence: 36897.49
## Random effects:
##  Groups   Name        Std.Dev.
##  sent     (Intercept) 17.40
##  part     (Intercept) 55.04
##  Residual             82.30
## Number of obs: 3148, groups:  sent, 80; part, 40
## Fixed Effects:
## (Intercept)          N1S          N2S     kindgram   kindungram
##    315.6727      -2.3076      -0.9150       0.5244       4.9255
```

```
mod5KR_humans_word4
```

```
## # A tibble: 1 x 12
##   N1_KR_stat N2_KR_stat kind_KR_stat N1_KR_ddf N2_KR_ddf kind_KR_ddf
##        <dbl>      <dbl>        <dbl>     <dbl>     <dbl>       <dbl>
## 1      0.618     0.0972        0.920     3027.     3026.       3026.
## # i 6 more variables: N1_KR_p.value <dbl>, N2_KR_p.value <dbl>,
## #   kind_KR_p.value <dbl>, N1_KR_mod_is_singular <lgl>,
## #   N2_KR_mod_is_singular <lgl>, kind_KR_mod_is_singular <lgl>
```

**models coefficients plot**

```
all_coeffs
```

```
##     model data effect        term    estimate   std.error   statistic
## 1  rubert both  fixed (Intercept)   0.1992566  0.21295677   0.9356671
## 2  rubert both  fixed         N1S  -0.8047714  0.08629360  -9.3259686
## 3  rubert both  fixed         N2S  -0.4620590  0.08629360  -5.3544990
## 4  rubert both  fixed   kind2gram   1.8779932  0.10568765  17.7692775
## 5  rubert both  fixed kind2ungram  -0.6775075  0.12203758  -5.5516300
## 6   rugpt both  fixed (Intercept)  -1.8294957  0.16973442 -10.7785779
## 7   rugpt both  fixed         N1S  -0.5545857  0.09010952  -6.1545735
## 8   rugpt both  fixed         N2S  -0.0419371  0.09010952  -0.4654014
## 9   rugpt both  fixed   kind2gram   3.5526376  0.11036118  32.1910083
## 10  rugpt both  fixed kind2ungram  -0.5385901  0.12743411  -4.2264203
## 11 rubert  acc  fixed (Intercept)   0.1265183  0.27743523   0.4560281
## 12 rubert  acc  fixed         N1S  -0.6882075  0.11543216  -5.9620085
## 13 rubert  acc  fixed         N2S  -0.5437673  0.11543216  -4.7107091
## 14 rubert  acc  fixed   kind2gram   1.4827230  0.14137494  10.4878770
## 15 rubert  acc  fixed kind2ungram  -1.0590033  0.16324572  -6.4871732
## 16 rubert  gen  fixed (Intercept)   0.2719950  0.31262724   0.8700297
## 17 rubert  gen  fixed         N1S  -0.9213354  0.12452520  -7.3987865
## 18 rubert  gen  fixed         N2S  -0.3803507  0.12452520  -3.0544075
## 19 rubert  gen  fixed   kind2gram   2.2732633  0.15251160  14.9055108
## 20 rubert  gen  fixed kind2ungram  -0.2960118  0.17610523  -1.6808801
## 21  rugpt  acc  fixed (Intercept)  -1.3491739  0.26204868  -5.1485622
## 22  rugpt  acc  fixed         N1S  -0.6498985  0.13356304  -4.8658563
## 23  rugpt  acc  fixed         N2S  -0.3351400  0.13356304  -2.5092272
## 24  rugpt  acc  fixed   kind2gram   3.3562510  0.16358064  20.5174092
## 25  rugpt  acc  fixed kind2ungram  -0.8145876  0.18888666  -4.3125736
## 26  rugpt  gen  fixed (Intercept)  -2.3098174  0.21372696 -10.8073284
## 27  rugpt  gen  fixed         N1S  -0.4592728  0.11702021  -3.9247309
## 28  rugpt  gen  fixed         N2S   0.2512658  0.11702021   2.1472001
## 29  rugpt  gen  fixed   kind2gram   3.7490242  0.14331990  26.1584336
## 30  rugpt  gen  fixed kind2ungram  -0.2625926  0.16549157  -1.5867432
## 31 humans both  fixed (Intercept) 367.8343369 11.03092557  33.3457365
## 32 humans both  fixed         N1S  -9.5289733  3.46194176  -2.7524938
## 33 humans both  fixed         N2S  -3.6730287  3.46178563  -1.0610214
## 34 humans both  fixed    kindgram -41.2814471  4.23890192  -9.7387125
## 35 humans both  fixed  kindungram  11.0967324  4.89228902   2.2682087
## 36 humans  acc  fixed (Intercept) 356.3522001 11.67803595  30.5147374
## 37 humans  acc  fixed         N1S  -9.2166965  4.94351488  -1.8644015
## 38 humans  acc  fixed         N2S   1.8047004  4.94360990   0.3650572
## 39 humans  acc  fixed    kindgram -33.7446457  6.05983510  -5.5685749
## 40 humans  acc  fixed  kindungram  30.3163129  6.99849232   4.3318349
## 41 humans  gen  fixed (Intercept) 379.3264545 12.23622360  31.0002879
## 42 humans  gen  fixed         N1S  -9.8882462  4.82236122  -2.0504989
## 43 humans  gen  fixed         N2S  -9.1985264  4.82151781  -1.9078072
## 44 humans  gen  fixed    kindgram -48.8705226  5.89769530  -8.2863763
## 45 humans  gen  fixed  kindungram  -7.9203762  6.80170357  -1.1644695
##            df       p.value
## 1    98.80736  3.517258e-01
## 2   444.00000  5.233865e-19
## 3   444.00000  1.379138e-07
```

```
## 4   444.00000  9.501329e-54
## 5   444.00000  4.878210e-08
## 6   144.12278  3.105766e-20
## 7   444.00000  1.684046e-09
## 8   444.00000  6.418722e-01
## 9   444.00000 3.621921e-118
## 10  444.00000  2.883642e-05
## 11   49.94842  6.503464e-01
## 12  220.00000  9.821753e-09
## 13  220.00000  4.372956e-06
## 14  220.00000  3.940727e-21
## 15  220.00000  5.699739e-10
## 16   47.82255  3.886303e-01
## 17  220.00000  2.869005e-12
## 18  220.00000  2.533360e-03
## 19  220.00000  3.380525e-35
## 20  220.00000  9.420535e-02
## 21   65.83810  2.571972e-06
## 22  220.00000  2.172052e-06
## 23  220.00000  1.282024e-02
## 24  220.00000  5.436279e-53
## 25  220.00000  2.436381e-05
## 26   75.47085  5.505423e-17
## 27  220.00000  1.161258e-04
## 28  220.00000  3.287105e-02
## 29  220.00000  1.837304e-69
## 30  220.00000  1.140070e-01
## 31   53.70897  1.422981e-37
## 32 3018.92039  5.949685e-03
## 33 3019.43548  2.887650e-01
## 34 3019.11320  4.342398e-22
## 35 3018.88267  2.338684e-02
## 36   68.58038  1.242613e-41
## 37 1486.79695  6.246228e-02
## 38 1487.08333  7.151206e-01
## 39 1487.55304  3.042930e-08
## 40 1487.72637  1.577756e-05
## 41   69.22332  2.548503e-42
## 42 1488.99522  4.049052e-02
## 43 1489.15093  5.660831e-02
## 44 1488.61434  2.574191e-16
## 45 1488.31271  2.444204e-01
```

```
all_coeffs %>%
  count(model, data)
```

```
##    model data n
## 1 humans  acc 5
## 2 humans both 5
## 3 humans  gen 5
## 4 rubert  acc 5
## 5 rubert both 5
## 6 rubert  gen 5
## 7  rugpt  acc 5
## 8  rugpt both 5
```

```
## 9  rugpt  gen 5
```

```
all_coeffs %>%
  count(term)
```

```
##          term n
## 1 (Intercept) 9
## 2         N1S 9
## 3         N2S 9
## 4   kind2gram 6
## 5 kind2ungram 6
## 6    kindgram 3
## 7  kindungram 3
```

```
all_coeffs %>%
  mutate(term = case_when(
    term == "kindungram" ~  "kind2ungram",
    term == "kindgram" ~  "kind2gram",
    TRUE ~ term,
  )) ->
  all_coeffs


all_coeffs %>%
  count(term)
```

```
##          term n
## 1 (Intercept) 9
## 2         N1S 9
## 3         N2S 9
## 4   kind2gram 9
## 5 kind2ungram 9
```

```
get_humans <- function(df){
  df %>% filter(model == "humans")
}

get_models <- function(df){
  df %>% filter(model != "humans")
}


round_any = function(x, accuracy, f=round){f(x/ accuracy) * accuracy}

find_offset <- function(variable, round_to=6){
  max_ = max(variable)
  min_ = min(variable)
  print(c(max_, min_))

  round_any((max_ + abs(min_) - 1), round_to, f=ceiling)
}

get_list_vals <- function(l){
  unname(unlist(l))
}
```

```
make_minor_breaks <- function(variable, offsets, step=0.5){
  max_ = ceiling(max(variable))
  print(max_)

  minor_breaks = c()
  for (offset in offsets){
    minor_breaks = c(minor_breaks, seq(offset - max_, offset + max_, step))
  }

  minor_breaks
}


make_coeffs_to_y_center <- function(df, coeffs_order, round_to=6){
  coeffs_y_step = find_offset(df$estimate, round_to = round_to)
  print(coeffs_y_step)

  coeffs_to_y_center = list()
  for (i in seq_along(coeffs_order)){
     coeffs_to_y_center = c(coeffs_to_y_center, (i - 1) * coeffs_y_step)
  }
  coeffs_to_y_center = setNames(coeffs_to_y_center, coeffs_order)
  coeffs_to_y_center
}
```

```
scale_human_subset <- function(df, scale_factor){
  df %>%
    mutate(estimate = case_when(
      model == "humans" ~ estimate / scale_factor,
      TRUE ~ estimate
    ))
}
```

```
all_coeffs
```

```
##     model data effect        term    estimate   std.error   statistic
## 1  rubert both  fixed (Intercept)   0.1992566  0.21295677   0.9356671
## 2  rubert both  fixed         N1S  -0.8047714  0.08629360  -9.3259686
## 3  rubert both  fixed         N2S  -0.4620590  0.08629360  -5.3544990
## 4  rubert both  fixed   kind2gram   1.8779932  0.10568765  17.7692775
## 5  rubert both  fixed kind2ungram  -0.6775075  0.12203758  -5.5516300
## 6   rugpt both  fixed (Intercept)  -1.8294957  0.16973442 -10.7785779
## 7   rugpt both  fixed         N1S  -0.5545857  0.09010952  -6.1545735
## 8   rugpt both  fixed         N2S  -0.0419371  0.09010952  -0.4654014
## 9   rugpt both  fixed   kind2gram   3.5526376  0.11036118  32.1910083
## 10  rugpt both  fixed kind2ungram  -0.5385901  0.12743411  -4.2264203
## 11 rubert  acc  fixed (Intercept)   0.1265183  0.27743523   0.4560281
## 12 rubert  acc  fixed         N1S  -0.6882075  0.11543216  -5.9620085
## 13 rubert  acc  fixed         N2S  -0.5437673  0.11543216  -4.7107091
## 14 rubert  acc  fixed   kind2gram   1.4827230  0.14137494  10.4878770
## 15 rubert  acc  fixed kind2ungram  -1.0590033  0.16324572  -6.4871732
## 16 rubert  gen  fixed (Intercept)   0.2719950  0.31262724   0.8700297
## 17 rubert  gen  fixed         N1S  -0.9213354  0.12452520  -7.3987865
## 18 rubert  gen  fixed         N2S  -0.3803507  0.12452520  -3.0544075
```

```
## 19 rubert  gen  fixed   kind2gram   2.2732633  0.15251160  14.9055108
## 20 rubert  gen  fixed kind2ungram  -0.2960118  0.17610523  -1.6808801
## 21  rugpt  acc  fixed (Intercept)  -1.3491739  0.26204868  -5.1485622
## 22  rugpt  acc  fixed         N1S  -0.6498985  0.13356304  -4.8658563
## 23  rugpt  acc  fixed         N2S  -0.3351400  0.13356304  -2.5092272
## 24  rugpt  acc  fixed   kind2gram   3.3562510  0.16358064  20.5174092
## 25  rugpt  acc  fixed kind2ungram  -0.8145876  0.18888666  -4.3125736
## 26  rugpt  gen  fixed (Intercept)  -2.3098174  0.21372696 -10.8073284
## 27  rugpt  gen  fixed         N1S  -0.4592728  0.11702021  -3.9247309
## 28  rugpt  gen  fixed         N2S   0.2512658  0.11702021   2.1472001
## 29  rugpt  gen  fixed   kind2gram   3.7490242  0.14331990  26.1584336
## 30  rugpt  gen  fixed kind2ungram  -0.2625926  0.16549157  -1.5867432
## 31 humans both  fixed (Intercept) 367.8343369 11.03092557  33.3457365
## 32 humans both  fixed         N1S  -9.5289733  3.46194176  -2.7524938
## 33 humans both  fixed         N2S  -3.6730287  3.46178563  -1.0610214
## 34 humans both  fixed   kind2gram -41.2814471  4.23890192  -9.7387125
## 35 humans both  fixed kind2ungram  11.0967324  4.89228902   2.2682087
## 36 humans  acc  fixed (Intercept) 356.3522001 11.67803595  30.5147374
## 37 humans  acc  fixed         N1S  -9.2166965  4.94351488  -1.8644015
## 38 humans  acc  fixed         N2S   1.8047004  4.94360990   0.3650572
## 39 humans  acc  fixed   kind2gram -33.7446457  6.05983510  -5.5685749
## 40 humans  acc  fixed kind2ungram  30.3163129  6.99849232   4.3318349
## 41 humans  gen  fixed (Intercept) 379.3264545 12.23622360  31.0002879
## 42 humans  gen  fixed         N1S  -9.8882462  4.82236122  -2.0504989
## 43 humans  gen  fixed         N2S  -9.1985264  4.82151781  -1.9078072
## 44 humans  gen  fixed   kind2gram -48.8705226  5.89769530  -8.2863763
## 45 humans  gen  fixed kind2ungram  -7.9203762  6.80170357  -1.1644695
##            df       p.value
## 1    98.80736  3.517258e-01
## 2   444.00000  5.233865e-19
## 3   444.00000  1.379138e-07
## 4   444.00000  9.501329e-54
## 5   444.00000  4.878210e-08
## 6   144.12278  3.105766e-20
## 7   444.00000  1.684046e-09
## 8   444.00000  6.418722e-01
## 9   444.00000 3.621921e-118
## 10  444.00000  2.883642e-05
## 11   49.94842  6.503464e-01
## 12  220.00000  9.821753e-09
## 13  220.00000  4.372956e-06
## 14  220.00000  3.940727e-21
## 15  220.00000  5.699739e-10
## 16   47.82255  3.886303e-01
## 17  220.00000  2.869005e-12
## 18  220.00000  2.533360e-03
## 19  220.00000  3.380525e-35
## 20  220.00000  9.420535e-02
## 21   65.83810  2.571972e-06
## 22  220.00000  2.172052e-06
## 23  220.00000  1.282024e-02
## 24  220.00000  5.436279e-53
## 25  220.00000  2.436381e-05
## 26   75.47085  5.505423e-17
```

```
## 27  220.00000  1.161258e-04
## 28  220.00000  3.287105e-02
## 29  220.00000  1.837304e-69
## 30  220.00000  1.140070e-01
## 31   53.70897  1.422981e-37
## 32 3018.92039  5.949685e-03
## 33 3019.43548  2.887650e-01
## 34 3019.11320  4.342398e-22
## 35 3018.88267  2.338684e-02
## 36   68.58038  1.242613e-41
## 37 1486.79695  6.246228e-02
## 38 1487.08333  7.151206e-01
## 39 1487.55304  3.042930e-08
## 40 1487.72637  1.577756e-05
## 41   69.22332  2.548503e-42
## 42 1488.99522  4.049052e-02
## 43 1489.15093  5.660831e-02
## 44 1488.61434  2.574191e-16
## 45 1488.31271  2.444204e-01
```

```
all_coeffs %>%
  write_csv("regression_coefficients.csv")
```

```
all_coeffs <- read_csv("regression_coefficients.csv")
```

```
## Rows: 45 Columns: 9
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (4): model, data, effect, term
## dbl (5): estimate, std.error, statistic, df, p.value
##
## 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.
```

```
all_coeffs
```

```
## # A tibble: 45 x 9
##    model  data  effect term        estimate std.error statistic    df   p.value
##    <chr>  <chr> <chr>  <chr>          <dbl>     <dbl>     <dbl> <dbl>     <dbl>
##  1 rubert both  fixed  (Intercept)   0.199     0.213      0.936  98.8 3.52e-  1
##  2 rubert both  fixed  N1S          -0.805     0.0863    -9.33  444.  5.23e- 19
##  3 rubert both  fixed  N2S          -0.462     0.0863    -5.35  444.  1.38e-  7
##  4 rubert both  fixed  kind2gram     1.88      0.106     17.8   444.  9.50e- 54
##  5 rubert both  fixed  kind2ungram  -0.678     0.122     -5.55  444.  4.88e-  8
##  6 rugpt  both  fixed  (Intercept)  -1.83      0.170    -10.8   144.  3.11e- 20
##  7 rugpt  both  fixed  N1S          -0.555     0.0901    -6.15  444.  1.68e-  9
##  8 rugpt  both  fixed  N2S          -0.0419    0.0901    -0.465 444.  6.42e-  1
##  9 rugpt  both  fixed  kind2gram     3.55      0.110     32.2   444.  3.62e-118
## 10 rugpt  both  fixed  kind2ungram  -0.539     0.127     -4.23  444.  2.88e-  5
## # i 35 more rows
```

```
alpha = 0.05

coeffs_order = c(
  "(Intercept)",
  "kind2ungram",
```

```
  "kind2gram",
  "N2S",
  "N1S"
)
final_names = c(
  "(Intercept)",
  "Kind=ungram-\nmatical_feats",
  "Kind=gram-\nmatical_feats",
  "Attractor\n Number = S",
  "Subject\n Number = S"
)
model_order = c("rubert", "rugpt", "humans")
coeffs_y_step = 6
y_limits = c(-3, 24)


all_coeffs_humans <- get_humans(all_coeffs)
all_coeffs_models <- get_models(all_coeffs)

scale_factor = round_any(max(all_coeffs_humans$estimate) / max(all_coeffs_models$estimate), 5)
all_coeffs_2 = scale_human_subset(all_coeffs, scale_factor)

coeffs_to_y_center = make_coeffs_to_y_center(all_coeffs_2, coeffs_order, round_to = coeffs_y_step)
```

```
## [1]  3.793265 -2.309817
## [1] 6
```

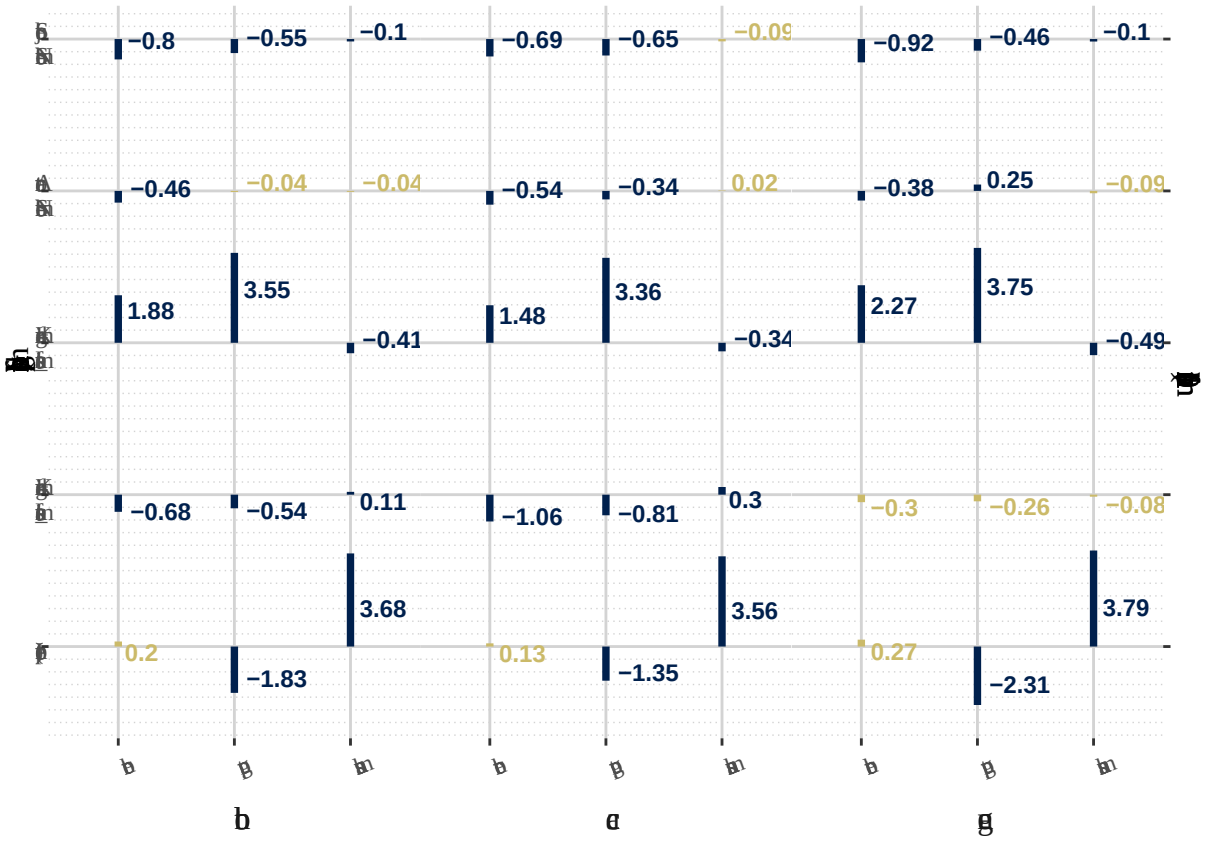
```
get_y_center <- function(coeff){
  coeffs_to_y_center[[coeff]]
}

make_range <- function(df){
  df %>%
    mutate(
      is_significant = if_else(p.value < alpha, TRUE, FALSE),
      coeff_y = sapply(term, get_y_center),
      estimate2 = coeff_y + estimate,
      estimate_lower = pmin(estimate2, coeff_y),
      estimate_upper = pmax(estimate2, coeff_y),
      estimate_center = (estimate_lower + estimate_upper) / 2,
    )
}


all_coeffs_2 %>%
  make_range() %>%
  mutate(
    is_significant=factor(is_significant, levels = c(TRUE, FALSE)),
    data=factor(data, levels = c("both", "acc", "gen")),
    model=factor(model, levels = model_order)
  ) ->
  all_coeffs_3

all_coeffs_3_humans = get_humans(all_coeffs_3)
```

```r
all_coeffs_3_models = get_models(all_coeffs_3)
coeffs_to_y_center <- setNames(coeffs_to_y_center, final_names)

all_coeffs_3 %>%
  {
    ggplot(., aes(
      y=estimate_center, x=model, color=is_significant, linetype=is_significant)
    ) +
    geom_linerange(
      data = all_coeffs_3_models,
      mapping = aes(y=estimate_center, ymin=estimate_lower, ymax=estimate_upper), linewidth=1.3
    ) +
    geom_linerange(
      data = all_coeffs_3_humans,
      mapping = aes(
        y=estimate_center, ymin=estimate_lower, ymax=estimate_upper
      ), linewidth=1.3
    ) +
    geom_text(aes(label=round(estimate, 2)), vjust="outward", hjust=-0.2, size=3, fontface="bold") +
    # scale_y_continuous(name="estimate_center", sec.axis=sec_axis(name="humans estimate_center (s)")) +
    # scale_color_discrete("cvidis") +
    scale_color_viridis(discrete = TRUE, option="E", begin=0, end=0.8, direction = 1) +
    facet_wrap(~data, strip.position = "bottom") +
    theme(
      # plot.margin = margin(r = 15, l = 15, unit = "pt"),
      panel.background = element_rect(
        fill = "white", colour = "white", size = 0.5, linetype = "solid"
      ),
      panel.grid.major = element_line(size = 0.5, linetype = 'solid', colour = "lightgray"),
      panel.grid.minor = element_line(size = 0.25, linetype = 'dotted', colour = "lightgray"),
      panel.spacing = unit(0, "lines"),

      strip.background = element_blank(),
      strip.clip="off",
      strip.placement = "outside",
      strip.text = element_text(size = 12),

      # base_size=10,
      text=element_text(family="Times New Roman"),
      axis.text.x = element_text(angle = 20, hjust = 0.5, vjust = 0.8),
      axis.title.y = element_text(size=12),

      legend.position = "none",
    ) +
    scale_y_continuous(
      breaks = get_list_vals(coeffs_to_y_center), labels = names(coeffs_to_y_center),
      minor_breaks = make_minor_breaks(.$estimate, get_list_vals(coeffs_to_y_center)),
      sec.axis = dup_axis(name="humans' coefficients / 100 (ms)", labels=NULL)
    ) +
    scale_x_discrete(
      limits = model_order,
      expand = expansion(add = c(0.6, 0.6)),
      # guide = guide_axis(n.dodge = 2)
```

```
    ) +
    labs(x = NULL, y="models' coefficients (log-prob units)")
  } ->
  all_coeffs_plot
```

```
## [1] 4
```

```
all_coeffs_plot
```




```
cairo_pdf("regressions_coefficients_06.pdf", width=8, height=4)

all_coeffs_plot

dev.off()
```

```
## pdf
##   2
```

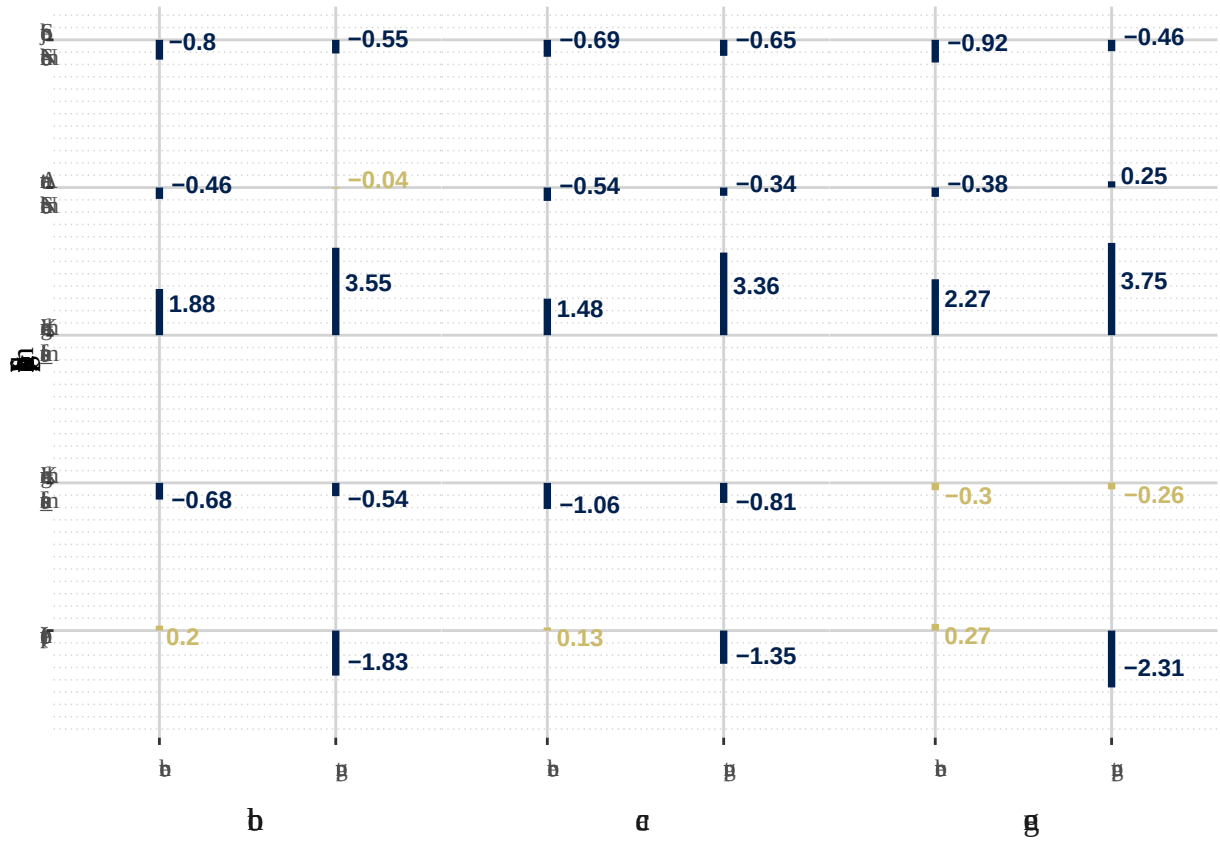
```
all_coeffs_3_models %>%
  mutate(model=as.character(model)) %>%
  {
    ggplot(., aes(y=estimate_center, x=model, color=is_significant, linetype=is_significant)) +
    geom_linerange(
      mapping = aes(y=estimate_center, ymin=estimate_lower, ymax=estimate_upper), linewidth=1.3
    ) +
    geom_text(aes(label=round(estimate, 2)), vjust="outward", hjust=-0.2, size=3, fontface="bold") +
    scale_color_viridis(discrete = TRUE, option="E", begin=0, end=0.8, direction = 1) +
```

```
    facet_wrap(~data, strip.position = "bottom") +
    theme(
      # plot.margin = margin(r = 15, l = 15, unit = "pt"),
      panel.background = element_rect(
        fill = "white", colour = "white", size = 0.5, linetype = "solid"
      ),
      panel.grid.major = element_line(size = 0.5, linetype = 'solid', colour = "lightgray"),
      panel.grid.minor = element_line(size = 0.25, linetype = 'dotted', colour = "lightgray"),
      panel.spacing = unit(0, "lines"),

      strip.background = element_blank(),
      strip.clip="off",
      strip.placement = "outside",
      strip.text = element_text(size = 12),

      # base_size=10,
      text=element_text(family="Times New Roman"),
      # axis.text.x = element_text(angle = 20, hjust = 0.5, vjust = 0.8),
      axis.title.y = element_text(size=12),

      legend.position = "none",
    ) +
    ylim(-6, 27) +
    scale_y_continuous(
      breaks = get_list_vals(coeffs_to_y_center), labels = names(coeffs_to_y_center),
      minor_breaks = make_minor_breaks(.$estimate, get_list_vals(coeffs_to_y_center)),
      # sec.axis = dup_axis(name="humans' coefficients / 100 (ms)", labels=NULL)
      limits = y_limits
    ) +
    scale_x_discrete(
      limits = c("rubert", "rugpt"),
      # expand = expansion(add = c(0.3, 0.3)),
      # guide = guide_axis(n.dodge = 2)
    ) +
    labs(x = NULL, y="models' coefficients (log-prob units)")
  } ->
  all_coeffs_models_plot
```

```
## [1] 4
```

```
## Scale for y is already present.
## Adding another scale for y, which will replace the existing scale.
```

```
all_coeffs_models_plot
```




```
cairo_pdf("regressions_coefficients_07_models.pdf", width=6, height=4)

all_coeffs_models_plot

dev.off()
```

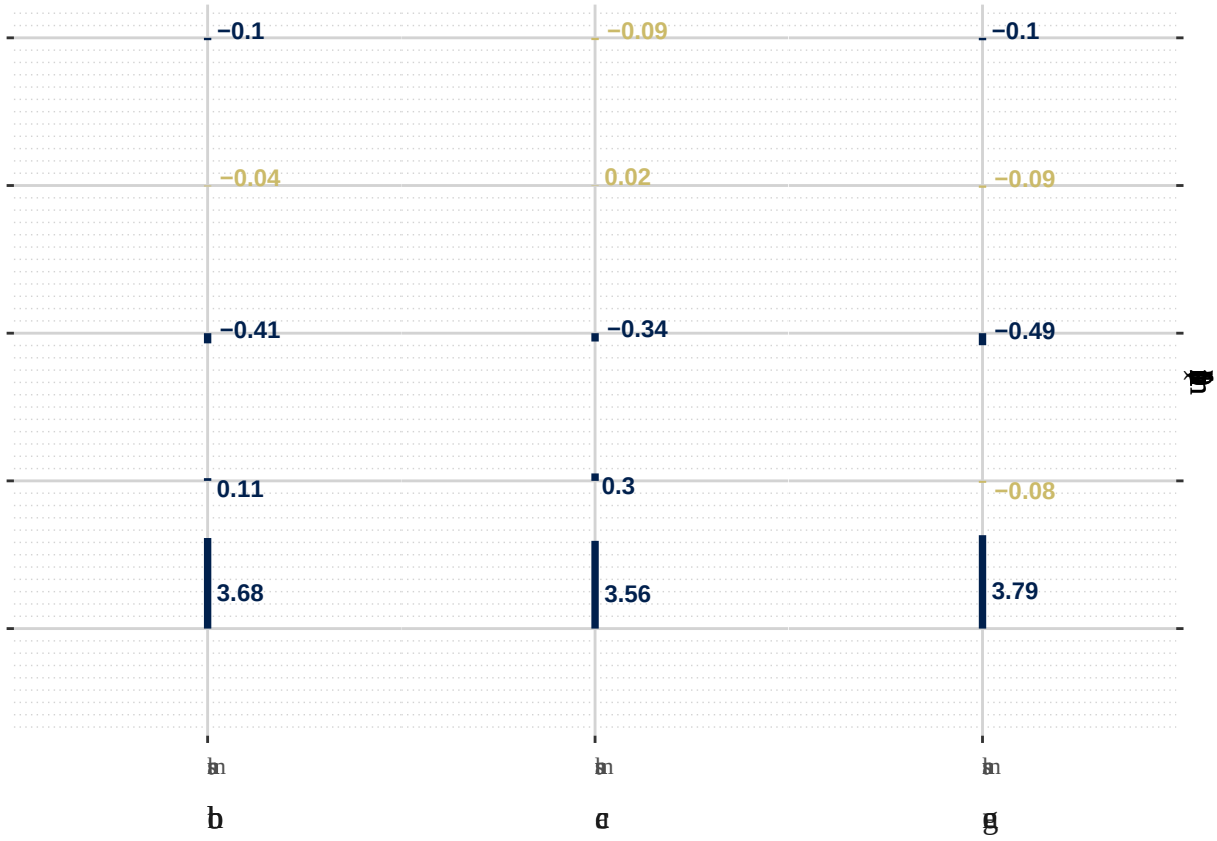
```
## pdf
##   2
```

```
all_coeffs_3_humans %>%
  mutate(model=as.character(model)) %>%
  {
    ggplot(., aes(y=estimate_center, x=model, color=is_significant, linetype=is_significant)) +
    geom_linerange(
      mapping = aes(y=estimate_center, ymin=estimate_lower, ymax=estimate_upper), linewidth=1.3
    ) +
    geom_text(aes(label=round(estimate, 2)), vjust="outward", hjust=-0.2, size=3, fontface="bold") +
    scale_color_viridis(discrete = TRUE, option="E", begin=0, end=0.8, direction = 1) +
    facet_wrap(~data, strip.position = "bottom") +
    theme(
      # plot.margin = margin(r = 15, l = 15, unit = "pt"),
      panel.background = element_rect(
        fill = "white", colour = "white", size = 0.5, linetype = "solid"
      ),
      panel.grid.major = element_line(size = 0.5, linetype = 'solid', colour = "lightgray"),
      panel.grid.minor = element_line(size = 0.25, linetype = 'dotted', colour = "lightgray"),
      panel.spacing = unit(0, "lines"),
```

```
      strip.background = element_blank(),
      strip.clip="off",
      strip.placement = "outside",
      strip.text = element_text(size = 12),

      text=element_text(family="Times New Roman"),
      # axis.text.x = element_text(size=12),
      axis.title.y = element_text(size=12),

      legend.position = "none",
    ) +
    scale_y_continuous(
      breaks = get_list_vals(coeffs_to_y_center),
      # labels = names(coeffs_to_y_center),
      labels = NULL,
      minor_breaks = make_minor_breaks(.$estimate, get_list_vals(coeffs_to_y_center)),
      sec.axis = dup_axis(name="humans' coefficients / 100 (ms)", labels=NULL),
      limits = y_limits
    ) +
    scale_x_discrete(
      # limits = c("rubert", "rugpt"),
      # expand = expansion(add = c(0.3, 0.3)),
      # guide = guide_axis(n.dodge = 2)
    ) +
    labs(x = NULL, y=NULL)
  } ->
  all_coeffs_humans_plot
```

```
## [1] 4
```

```
all_coeffs_humans_plot
```




```
cairo_pdf("regressions_coefficients_06_humans.pdf", width=3, height=4)

all_coeffs_humans_plot

dev.off()
```

```
## pdf
##   2
```