Multimodal Models and Affordances

February 12, 2024

Contents

Load data

```r
# setwd("/Users/seantrott/Dropbox/UCSD/Research/NLMs/fmp_research/main_experiment/")
df_natural_gpt = read_csv("gpt4v_results/df_natural_temp0.csv")
```

```
## Rows: 36 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (9): condition, prompt_type, key_verb, afforded_text, non-afforded_text,...
## dbl (4): group_id, gpt4v_result_afforded, gpt4v_result_non_afforded, gpt4v_r...
##
## 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.
```

```r
df_synthetic_gpt = read_csv("gpt4v_results/df_synthetic_temp0.csv")
```

```
## Rows: 36 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (9): condition, prompt_type, key_verb, afforded_text, non-afforded_text,...
## dbl (4): group_id, gpt4v_result_afforded, gpt4v_result_non_afforded, gpt4v_r...
##
## 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.
```

```r
nrow(df_natural_gpt)
```

```
## [1] 36
```

```r
nrow(df_synthetic_gpt)
```

```
## [1] 36
```

Analysis and Visualization of GPT-4V

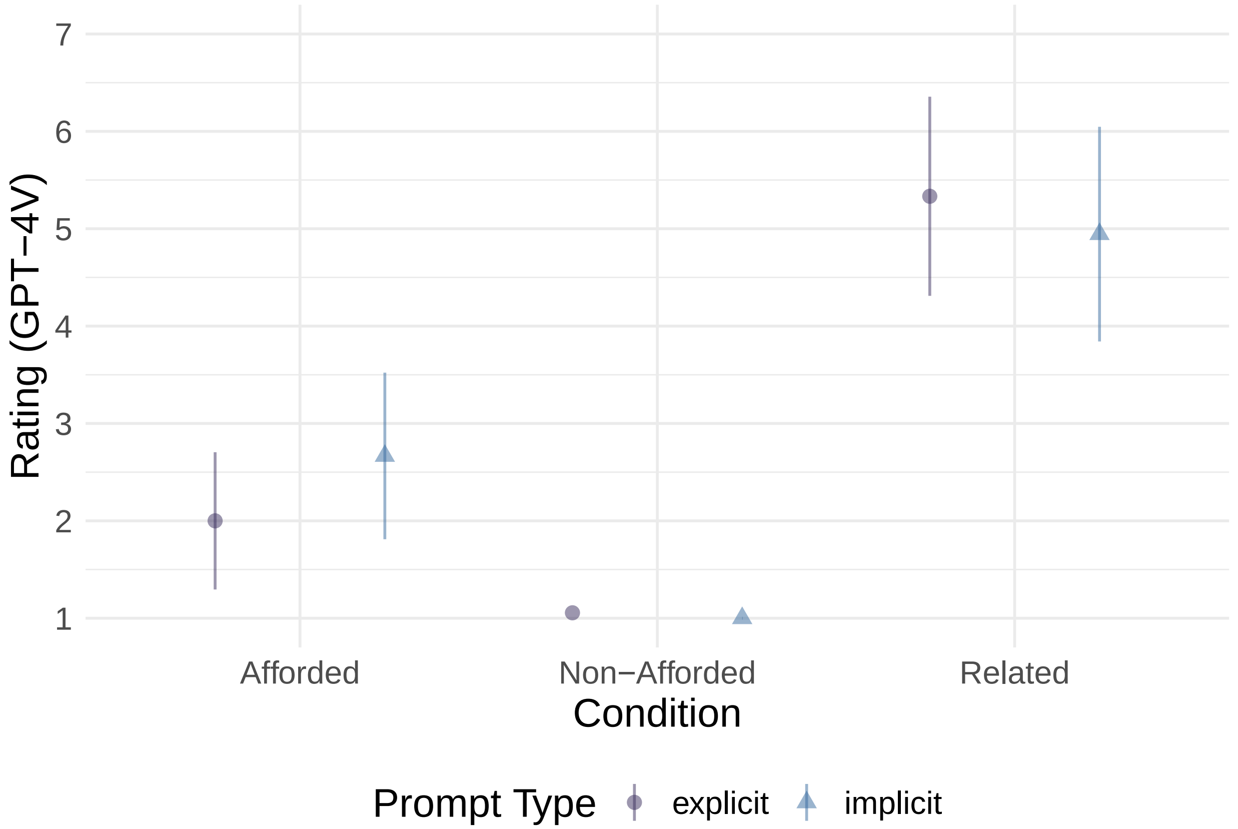
Natural stimuli

```
df_natural_long = df_natural_gpt %>%
  pivot_longer(
    cols = starts_with("gpt4v_result_"), # Select columns starting with 'gpt4v_result_'
    names_to = "Condition", # Name of the new column to create
    names_prefix = "gpt4v_result_", # Remove this prefix from the selected column names
    values_to = "Rating" # The values from these columns go into the 'Result' column
  )

# Optionally, adjust the 'Condition' values if needed
df_natural_long$Condition = case_when(
  df_natural_long$Condition == "afforded" ~ "Afforded",
  df_natural_long$Condition == "non_afforded" ~ "Non-Afforded",
  df_natural_long$Condition == "related" ~ "Related",
  TRUE ~ as.character(df_natural_long$Condition) # Fallback to original value
)

df_natural_long %>%
  ggplot(aes(x = Condition, y = Rating, color = prompt_type, shape = prompt_type)) +
  stat_summary (fun = function(x){mean(x)},
              fun.min = function(x){mean(x) - 2*sd(x)/sqrt(length(x))},
              fun.max = function(x){mean(x) + 2*sd(x)/sqrt(length(x))},
              geom= 'pointrange',
              position=position_dodge(width=0.95),
              size = .5, alpha = .5) +
  labs(x="Condition",
       y="Rating (GPT-4V)",
       color = "Prompt Type",
       shape = "Prompt Type") +
  theme_minimal() +
  theme(text = element_text(size = 15),
        legend.position="bottom") +
  scale_y_continuous(limits = c(1, 7), breaks = seq(1:7)) +
  scale_color_manual(values = my_colors)
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_segment()`).
```

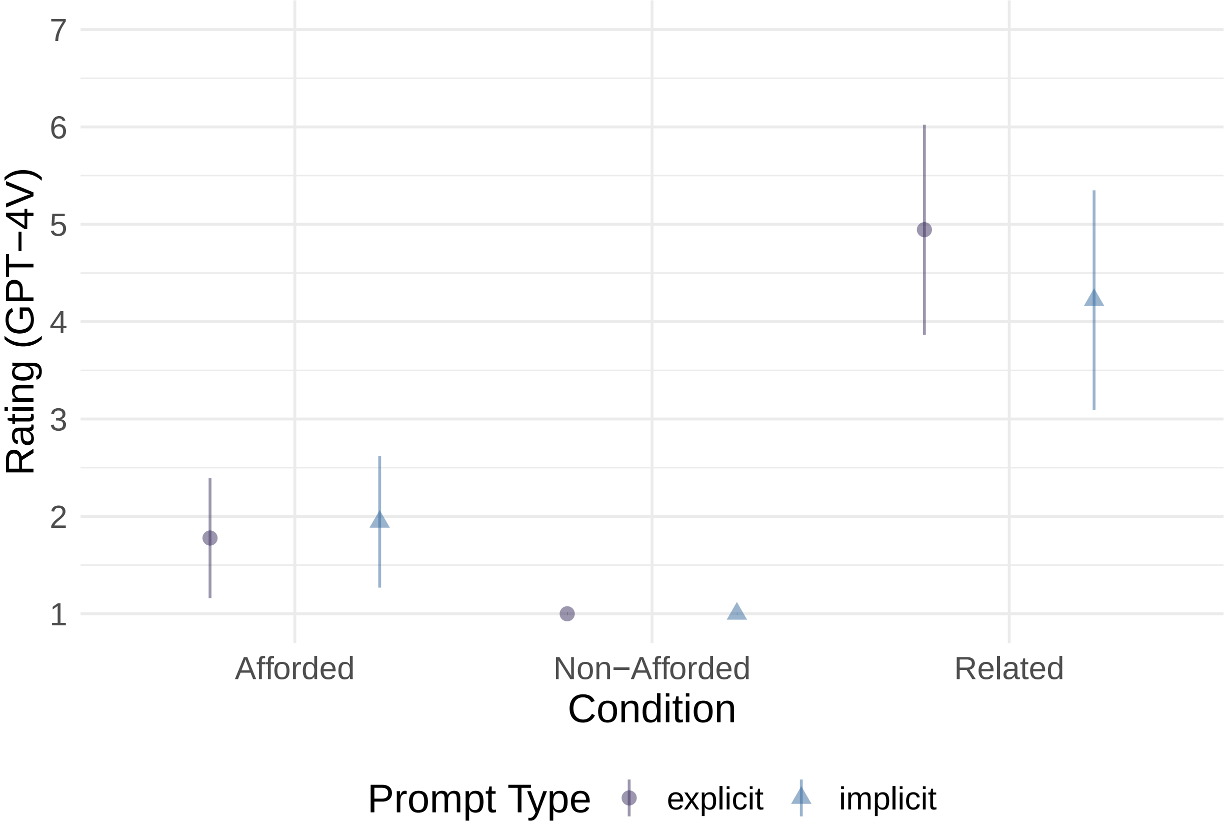


Synthetic stimuli

```
df_synthetic_long = df_synthetic_gpt  %>%
  pivot_longer(
    cols = starts_with("gpt4v_result_"), # Select columns starting with 'gpt4v_result_'
    names_to = "Condition", # Name of the new column to create
    names_prefix = "gpt4v_result_", # Remove this prefix from the selected column names
    values_to = "Rating" # The values from these columns go into the 'Result' column
  )

# Optionally, adjust the 'Condition' values if needed
df_synthetic_long$Condition = case_when(
  df_synthetic_long$Condition == "afforded" ~ "Afforded",
  df_synthetic_long$Condition == "non_afforded" ~ "Non-Afforded",
  df_synthetic_long$Condition == "related" ~ "Related",
  TRUE ~ as.character(df_synthetic_long$Condition) # Fallback to original value
)

df_synthetic_long %>%
  ggplot(aes(x = Condition, y = Rating, color = prompt_type, shape = prompt_type)) +
  stat_summary (fun = function(x){mean(x)},
                fun.min = function(x){mean(x) - 2*sd(x)/sqrt(length(x))},
                fun.max = function(x){mean(x) + 2*sd(x)/sqrt(length(x))},
                geom= 'pointrange',
                position=position_dodge(width=0.95),
                size = .5, alpha = .5) +
  labs(x="Condition",
       y="Rating (GPT-4V)",
       color = "Prompt Type",
       shape = "Prompt Type") +
  theme_minimal() +
  theme(text = element_text(size = 15),
        legend.position="bottom") +
  scale_y_continuous(limits = c(1, 7), breaks = seq(1:7)) +
  scale_color_manual(values = my_colors)
```



All together

```
### First, merge all together
df_natural_long_subset = df_natural_long %>%
  select(Condition, Rating, prompt_type, group_id) %>%
  mutate(Stimuli = "Natural")

df_synthetic_long_subset = df_synthetic_long %>%
  select(Condition, Rating, prompt_type, group_id) %>%
  mutate(Stimuli = "Synthetic")

df_combined = df_synthetic_long_subset %>%
  rbind(df_natural_long_subset)
nrow(df_combined)
```

```
## [1] 216
```

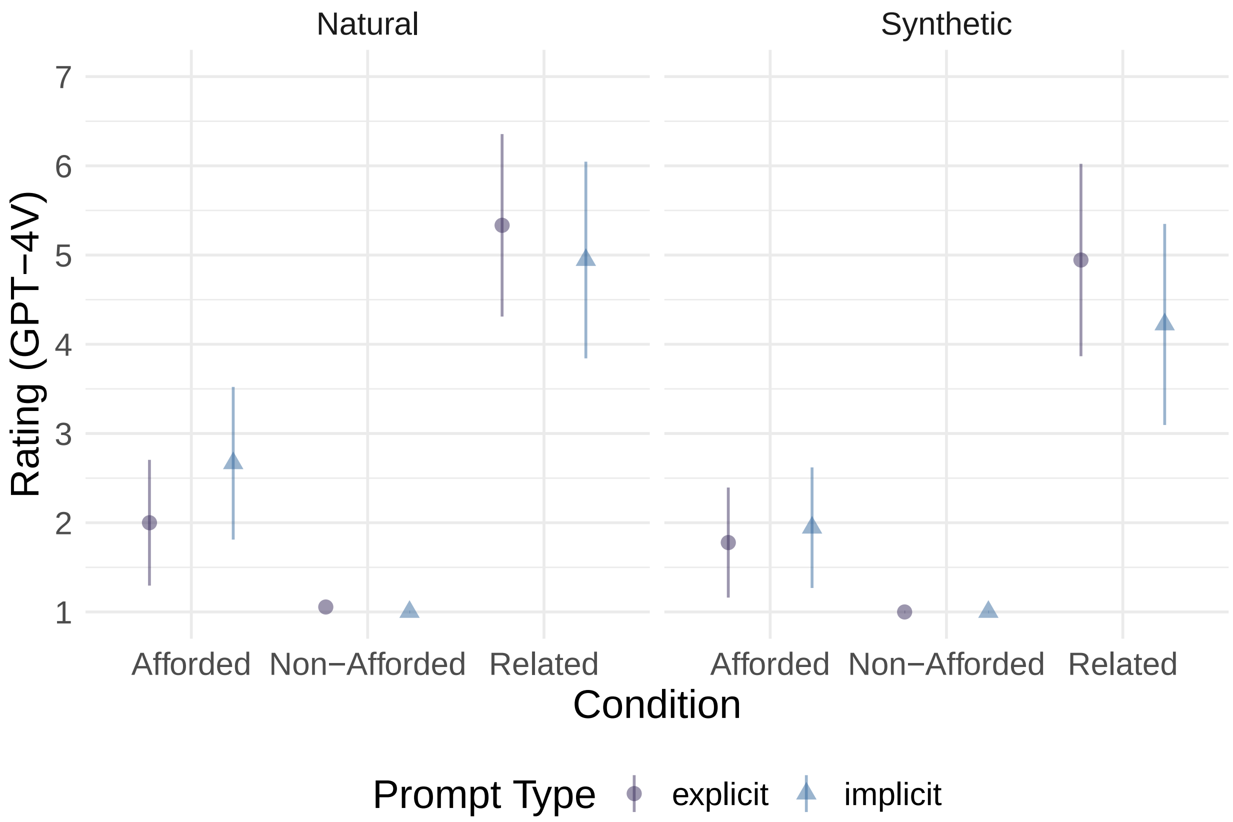
```
df_combined %>%
  group_by(Condition, Stimuli) %>%
  summarise(m_rating = mean(Rating),
            sd_rating = sd(Rating))
```

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

```
## # A tibble: 6 x 4
## # Groups:   Condition [3]
##   Condition    Stimuli   m_rating sd_rating
##   <chr>        <chr>        <dbl>     <dbl>
## 1 Afforded     Natural       2.33     1.67
## 2 Afforded     Synthetic     1.86     1.36
## 3 Non-Afforded Natural       1.03     0.167
## 4 Non-Afforded Synthetic     1        0
## 5 Related      Natural       5.14     2.23
## 6 Related      Synthetic     4.58     2.33
```

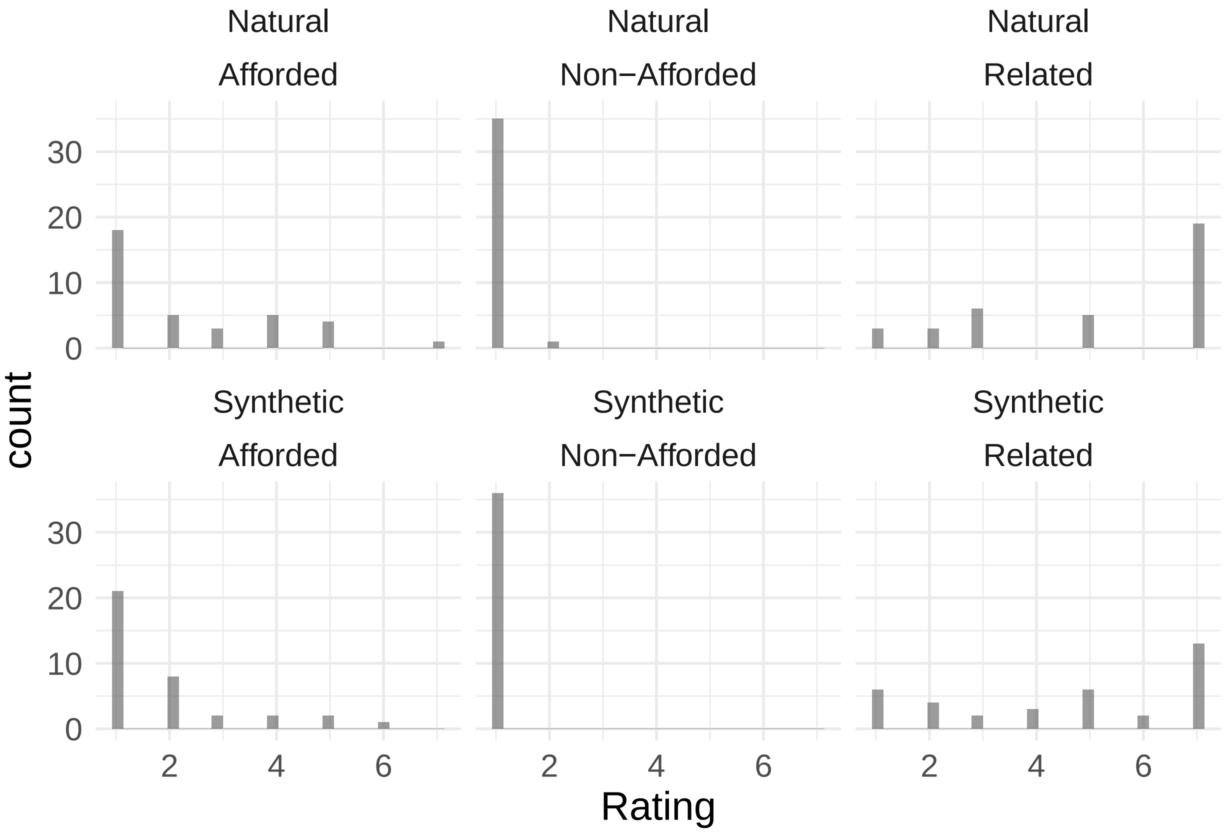
```
df_combined %>%
  ggplot(aes(x = Condition, y = Rating, color = prompt_type, shape = prompt_type)) +
  stat_summary (fun = function(x){mean(x)},
              fun.min = function(x){mean(x) - 2*sd(x)/sqrt(length(x))},
              fun.max = function(x){mean(x) + 2*sd(x)/sqrt(length(x))},
              geom= 'pointrange',
              position=position_dodge(width=0.95),
              size = .5, alpha = .5) +
  labs(x="Condition",
       y="Rating (GPT-4V)",
       color = "Prompt Type",
       shape = "Prompt Type") +
  theme_minimal() +
  theme(text = element_text(size = 15),
        legend.position="bottom") +
  scale_y_continuous(limits = c(1, 7), breaks = seq(1:7)) +
  scale_color_manual(values = my_colors) +
  facet_wrap(~Stimuli)
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_segment()`).
```



```
df_combined %>%
  ggplot(aes(x = Rating)) +
  geom_histogram(alpha = .6) +
  labs(x="Rating") +
  theme_minimal() +
  theme(text = element_text(size = 15),
        legend.position="bottom") +
  facet_wrap(~Stimuli + Condition)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Afforded vs. Non-afforded

```
df_aff_nonaff = df_combined %>%
  filter(Condition != "Related")
nrow(df_aff_nonaff)
```

```
## [1] 144
```

```
mod_full = lmer(data = df_aff_nonaff,
                Rating ~ Condition * prompt_type +
                  (1 | Stimuli) +
                  (1 |group_id), REML = FALSE)

mod_just_fe = lmer(data = df_aff_nonaff,
                Rating ~  Condition + prompt_type +
                  (1 | Stimuli) +
                  (1|group_id), REML = FALSE)

mod_just_prompt_type = lmer(data = df_aff_nonaff,
                Rating ~  prompt_type +
                  (1 | Stimuli) +
                  (1|group_id), REML = FALSE)

mod_just_condition = lmer(data = df_aff_nonaff,
                Rating ~  Condition +
                  (1 | Stimuli) +
                  (1|group_id), REML = FALSE)
```

```
summary(mod_full)
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula: Rating ~ Condition * prompt_type + (1 | Stimuli) + (1 | group_id)
##    Data: df_aff_nonaff
##
##      AIC      BIC   logLik deviance df.resid
##    435.0    455.8   -210.5    421.0      137
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.5323 -0.6141 -0.1098  0.2754  4.5945
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  group_id (Intercept) 0.15873  0.39842
##  Stimuli  (Intercept) 0.00677  0.08228
##  Residual             0.97768  0.98878
## Number of obs: 144, groups:  group_id, 18; Stimuli, 2
##
## Fixed effects:
##                                           Estimate Std. Error       df t value
## (Intercept)                                 1.8889     0.1984  19.9703   9.521
## ConditionNon-Afforded                      -0.8611     0.2331 124.1302  -3.695
## prompt_typeimplicit                         0.4167     0.2331 124.1302   1.788
## ConditionNon-Afforded:prompt_typeimplicit  -0.4444     0.3296 124.1302  -1.348
##                                           Pr(>|t|)
## (Intercept)                               7.28e-09 ***
## ConditionNon-Afforded                     0.000329 ***
## prompt_typeimplicit                       0.076244 .
## ConditionNon-Afforded:prompt_typeimplicit 0.179964
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) CndN-A prmpt_
## CndtnNn-Aff -0.587
## prmpt_typmp -0.587  0.500
## CndtnNn-A:_  0.415 -0.707 -0.707
```

```
anova(mod_full, mod_just_fe)
```

```
## Data: df_aff_nonaff
## Models:
## mod_just_fe: Rating ~ Condition + prompt_type + (1 | Stimuli) + (1 | group_id)
## mod_full: Rating ~ Condition * prompt_type + (1 | Stimuli) + (1 | group_id)
##             npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## mod_just_fe    6 434.79 452.61 -211.40   422.79
## mod_full       7 434.99 455.78 -210.49   420.99 1.8052  1     0.1791
```

```
anova(mod_just_fe, mod_just_prompt_type)
```

```
## Data: df_aff_nonaff
## Models:
```

```
## mod_just_prompt_type: Rating ~ prompt_type + (1 | Stimuli) + (1 | group_id)
## mod_just_fe: Rating ~ Condition + prompt_type + (1 | Stimuli) + (1 | group_id)
##                      npar    AIC    BIC logLik deviance  Chisq Df Pr(>Chisq)
## mod_just_prompt_type    5 469.40 484.25 -229.7   459.40
## mod_just_fe             6 434.79 452.61 -211.4   422.79 36.608  1  1.444e-09
##
## mod_just_prompt_type
## mod_just_fe          ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(mod_just_fe, mod_just_condition)
```

```
## Data: df_aff_nonaff
## Models:
## mod_just_condition: Rating ~ Condition + (1 | Stimuli) + (1 | group_id)
## mod_just_fe: Rating ~ Condition + prompt_type + (1 | Stimuli) + (1 | group_id)
##                    npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## mod_just_condition    5 434.16 449.01 -212.08   424.16
## mod_just_fe           6 434.79 452.61 -211.40   422.79 1.3645  1     0.2428
```

Afforded vs. Related

```
df_aff_rel = df_combined %>%
  filter(Condition != "Non-Afforded")
nrow(df_aff_rel)
```

```
## [1] 144
```

```
mod_full = lmer(data = df_aff_rel,
                Rating ~ Condition * prompt_type +
                  (1 + Condition | Stimuli) +
                  (1|group_id), REML = FALSE)
```

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

```
mod_just_fe = lmer(data = df_aff_rel,
                Rating ~  Condition + prompt_type +
                  (1 + Condition | Stimuli) +
                  (1|group_id), REML = FALSE)
```

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

```
mod_just_prompt_type = lmer(data = df_aff_rel,
                Rating ~  prompt_type +
                  (1 + Condition | Stimuli) +
                  (1|group_id), REML = FALSE)
```

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

```
## Warning: Model failed to converge with 1 negative eigenvalue: -6.8e-04
```

```
mod_just_condition = lmer(data = df_aff_rel,
                Rating ~  Condition +
                  (1 + Condition | Stimuli) +
                  (1|group_id), REML = FALSE)
```

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

```r
summary(mod_full)
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula: Rating ~ Condition * prompt_type + (1 + Condition | Stimuli) +
##     (1 | group_id)
##    Data: df_aff_rel
##
##      AIC      BIC   logLik deviance df.resid
##    607.3    634.1   -294.7    589.3      135
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.37490 -0.64220 -0.09949  0.80546  2.49795
##
## Random effects:
##  Groups   Name             Variance  Std.Dev. Corr
##  group_id (Intercept)      0.5017623 0.70835
##  Stimuli  (Intercept)      0.0342854 0.18516
##           ConditionRelated 0.0006588 0.02567  1.00
##  Residual                  3.1429847 1.77285
## Number of obs: 144, groups:  group_id, 18; Stimuli, 2
##
## Fixed effects:
##                                      Estimate Std. Error       df t value
## (Intercept)                            1.8889     0.3638  12.9863   5.193
## ConditionRelated                       3.2500     0.4183 111.9977   7.770
## prompt_typeimplicit                    0.4167     0.4179 124.2497   0.997
## ConditionRelated:prompt_typeimplicit  -0.9722     0.5909 124.2497  -1.645
##                                      Pr(>|t|)
## (Intercept)                          0.000174 ***
## ConditionRelated                     4.08e-12 ***
## prompt_typeimplicit                  0.320637
## ConditionRelated:prompt_typeimplicit 0.102459
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) CndtnR prmpt_
## ConditnRltd -0.558
## prmpt_typmp -0.574  0.500
## CndtnRltd:_  0.406 -0.706 -0.707
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

```r
anova(mod_full, mod_just_fe)
```

```
## Data: df_aff_rel
## Models:
## mod_just_fe: Rating ~ Condition + prompt_type + (1 + Condition | Stimuli) + (1 | group_id)
## mod_full: Rating ~ Condition * prompt_type + (1 + Condition | Stimuli) + (1 | group_id)
##             npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## mod_just_fe    8 608.03 631.79 -296.01   592.03
## mod_full       9 607.35 634.08 -294.68   589.35 2.6776  1     0.1018
```

```
anova(mod_just_fe, mod_just_prompt_type)
```

```
## Data: df_aff_rel
## Models:
## mod_just_prompt_type: Rating ~ prompt_type + (1 + Condition | Stimuli) + (1 | group_id)
## mod_just_fe: Rating ~ Condition + prompt_type + (1 + Condition | Stimuli) + (1 | group_id)
##                      npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## mod_just_prompt_type    7 615.30 636.08 -300.65   601.30
## mod_just_fe             8 608.03 631.79 -296.01   592.03 9.2676  1   0.002332
##
## mod_just_prompt_type
## mod_just_fe          **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(mod_just_fe, mod_just_condition)
```

```
## Data: df_aff_rel
## Models:
## mod_just_condition: Rating ~ Condition + (1 + Condition | Stimuli) + (1 | group_id)
## mod_just_fe: Rating ~ Condition + prompt_type + (1 + Condition | Stimuli) + (1 | group_id)
##                    npar    AIC    BIC  logLik deviance Chisq Df Pr(>Chisq)
## mod_just_condition    7 606.08 626.87 -296.04   592.08
## mod_just_fe           8 608.03 631.79 -296.01   592.03 0.054  1     0.8162
```

Non-Afforded vs. Related

```
df_nonaff_rel = df_combined %>%
  filter(Condition != "Afforded")
nrow(df_nonaff_rel)
```

```
## [1] 144
```

```
mod_full = lmer(data = df_nonaff_rel,
                Rating ~ Condition * prompt_type +
                  (1 + Condition | Stimuli) +
                  (1|group_id), REML = FALSE)
```

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

```
## Warning: Model failed to converge with 1 negative eigenvalue: -8.9e-02
```

```
mod_just_fe = lmer(data = df_nonaff_rel,
                Rating ~  Condition + prompt_type +
                  (1 + Condition | Stimuli) +
                  (1|group_id), REML = FALSE)
```

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

```
mod_just_prompt_type = lmer(data = df_nonaff_rel,
                Rating ~  prompt_type +
                  (1 + Condition | Stimuli) +
                  (1|group_id), REML = FALSE)
```

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

```
mod_just_condition = lmer(data = df_nonaff_rel,
                Rating ~  Condition +
```

```r
                   (1 + Condition | Stimuli) +
                   (1|group_id), REML = FALSE)
```

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

```r
summary(mod_full)
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
##   method [lmerModLmerTest]
## Formula: Rating ~ Condition * prompt_type + (1 + Condition | Stimuli) +
##     (1 | group_id)
##    Data: df_nonaff_rel
##
##      AIC      BIC   logLik deviance df.resid
##    549.3    576.0   -265.7    531.3      135
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.68574 -0.49008  0.04793  0.70126  1.82546
##
## Random effects:
##  Groups   Name             Variance Std.Dev. Corr
##  group_id (Intercept)      0.49151  0.7011
##  Stimuli  (Intercept)      0.00000  0.0000
##           ConditionRelated 0.03108  0.1763    NaN
##  Residual                  2.03855  1.4278
## Number of obs: 144, groups:  group_id, 18; Stimuli, 2
##
## Fixed effects:
##                                      Estimate Std. Error        df t value
## (Intercept)                           1.02778    0.28971  64.03420   3.548
## ConditionRelated                      4.11111    0.35888  11.59613  11.455
## prompt_typeimplicit                  -0.02778    0.33653 124.15730  -0.083
## ConditionRelated:prompt_typeimplicit  -0.52778    0.47593 124.15730  -1.109
##                                      Pr(>|t|)
## (Intercept)                          0.000734 ***
## ConditionRelated                     1.13e-07 ***
## prompt_typeimplicit                  0.934349
## ConditionRelated:prompt_typeimplicit 0.269595
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) CndtnR prmpt_
## ConditnRltd -0.545
## prmpt_typmp -0.581  0.469
## CndtnRltd:_  0.411 -0.663 -0.707
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

```r
anova(mod_full, mod_just_fe)
```

```
## Data: df_nonaff_rel
## Models:
## mod_just_fe: Rating ~ Condition + prompt_type + (1 + Condition | Stimuli) + (1 | group_id)
## mod_full: Rating ~ Condition * prompt_type + (1 + Condition | Stimuli) + (1 | group_id)
##             npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## mod_just_fe    8 548.53 572.29 -266.27   532.53
## mod_full       9 549.31 576.04 -265.65   531.31 1.2237  1     0.2686
```

```
anova(mod_just_fe, mod_just_prompt_type)
```

```
## Data: df_nonaff_rel
## Models:
## mod_just_prompt_type: Rating ~ prompt_type + (1 + Condition | Stimuli) + (1 | group_id)
## mod_just_fe: Rating ~ Condition + prompt_type + (1 + Condition | Stimuli) + (1 | group_id)
##                      npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## mod_just_prompt_type    7 556.42 577.21 -271.21   542.42
## mod_just_fe             8 548.53 572.29 -266.27   532.53 9.8869  1   0.001665
##
## mod_just_prompt_type
## mod_just_fe          **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(mod_just_fe, mod_just_condition)
```

```
## Data: df_nonaff_rel
## Models:
## mod_just_condition: Rating ~ Condition + (1 + Condition | Stimuli) + (1 | group_id)
## mod_just_fe: Rating ~ Condition + prompt_type + (1 + Condition | Stimuli) + (1 | group_id)
##                    npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## mod_just_condition    7 547.98 568.77 -266.99   533.98
## mod_just_fe           8 548.53 572.29 -266.27   532.53 1.4453  1     0.2293
```