Supplemental file

Last updated 2021-07-30

Contents

Cleaning

Workspace

```
library(here) # for working with files
library(tidyverse) # for cleaning
library(janitor) # for variable names
library(lme4) # for multilevel modeling
library(lmerTest) # for p-values
library(sjPlot) # for figures
library(ggpubr) # for prettier plots
library(kableExtra) # for nicer tables
library(stringdist) # for scoring memory task
library(papaja) # for pretty numbers
```

```
data_path = here("data/Wording_July 13, 2021_20.00.text.csv")

data_labels = read_csv(data_path)

data = read_csv(data_path,
                skip = 3,
                col_names = names(data_labels))
rm(data_labels)
data = clean_names(data)
```

Remove the following columns.

```
data = data %>%
  select(-end_date,
         -ip_address,
         -progress,
         -finished,
         -recorded_date,
         -external_reference,
         -distribution_channel,
         -user_language,
         -starts_with("recipient"),
         -starts_with("location"),
         -starts_with("meta_info"))
```

Recode personality item responses to numeric

We recode the responses to personality items, which we downloaded as text strings.

```
p_items = str_extract(names(data), "^[[:alpha:]]*_[abcd](_2)?$")
p_items = p_items[!is.na(p_items)]

personality_items = select(data, proid, all_of(p_items))
```

Next we write a simple function to recode values.

```r
recode_p = function(x){
  y = case_when(
    x == "Very inaccurate" ~ 1,
    x == "Moderately inaccurate" ~ 2,
    x == "Slightly inaccurate" ~ 3,
    x == "Slightly accurate" ~ 4,
    x == "Moderately accurate" ~ 5,
    x == "Very accurate" ~ 6,
    TRUE ~ NA_real_)
  return(y)
}
```

Finally, we apply this function to all personality items.

```r
personality_items = personality_items %>%
  mutate(
    across(!c(proid), recode_p))
```

Now we merge this back into the data.

```r
data = select(data, -all_of(p_items))
data = full_join(data, personality_items)
```

Drop bots

Based on ID

We removed 5 participants without valid Prolific IDs.

```r
data = data %>%
  mutate(proid = str_remove(proid, "Value will be set from panel or URL"),
         proid = str_remove(proid, "Value will be set from panel or UR"),
         proid = str_remove(proid, "TEST")) %>%
  filter(proid != "")
```

We removed 0 participants that do not speak english well or very well.

Based on patterns

We remove any participant who provides the same response to over half of the items (17 or more items) from a given block in a row.

```r
# first, identify unique adjectives, in order
adjectives = p_items %>%
  str_remove_all("_.") %>%
  unique()

# extract block 1 questions
block1 = data %>%
  select(proid, matches("^[[:alpha:]]+_[abcd]$"))
```

```r
#rename variables
n = 0
for(i in adjectives){
  n = n+1
  names(block1) = str_replace(names(block1), i, paste0("trait", str_pad(n, 2, pad = "0")))
}


block1 = block1 %>%
  gather(item, response, -proid) %>%
  filter(!is.na(response)) %>%
  separate(item, into = c("item", "format")) %>%
  select(-format) %>%
  spread(item, response)

block1_runs = numeric(length = nrow(block1))

#   working on this!!!
for(i in 1:nrow(block1)){
  run = 0
  maxrun = 0
  for(j in 3:ncol(block1)){
    if(block1[i,j] == block1[i, j-1]){
      run = run+1
      if(run > maxrun) maxrun = run
      } else{ run = 0}
  }
  block1_runs[i] = maxrun
}

#add to data frame
block1$block1_runs = block1_runs
```

```r
# extract block 2 questions
block2 = data %>%
  select(proid, matches("^[[:alpha:]]+_[abcd]_2$"))

#rename variables
n = 0
for(i in adjectives){
  n = n+1
  names(block2) = str_replace(names(block2), i, paste0("trait", str_pad(n, 2, pad = "0")))
}

block2 = block2 %>%
  gather(item, response, -proid) %>%
  filter(!is.na(response)) %>%
  mutate(item = str_remove(item, "_2")) %>%
  separate(item, into = c("item", "format")) %>%
  select(-format) %>%
  spread(item, response)

block2_runs = numeric(length = nrow(block2))
```

```
#   working on this!!!
for(i in 1:nrow(block2)){
  run = 0
  maxrun = 0
  for(j in 3:ncol(block2)){
    if(block2[i,j] == block2[i, j-1]){
      run = run+1
      if(run > maxrun) maxrun = run
      } else{ run = 0}
  }
  block2_runs[i] = maxrun
}

#add to data frame
block2$block2_runs = block2_runs
```

```
#combine results
runs_data = block1 %>%
  select(proid, block1_runs) %>%
  full_join(select(block2, proid, block2_runs)) %>%
  mutate(
    remove = case_when(
      block1_runs >= 17 ~ "Remove",
      block2_runs >= 17 ~ "Remove",
      TRUE ~ "Keep"
    ))
```

```
#visualize
runs_data %>%
  ggplot(aes(block1_runs, block2_runs)) +
  geom_point(aes(color = remove)) +
  scale_color_manual(values = c("black", "red")) +
  guides(color = "none") +
  labs(
    x = "block 1 runs",
    y = "block 2 runs"
  ) +
  theme_pubr()
```

There were 2 participants who provided the same answer 17 or more times in a row. These participants were removed from the analyses.

```
data = data %>%
  full_join(select(runs_data, proid, remove)) %>%
  filter(remove != "Remove") %>%
  select(-remove)

rm(runs_data)
```

Based on inattentive responding

We expect to exclude any participant who has an average response of 4 ("slightly agree") or greater to the attention check items. Two items from the Inattentive and Deviant Responding Inventory for Adjectives

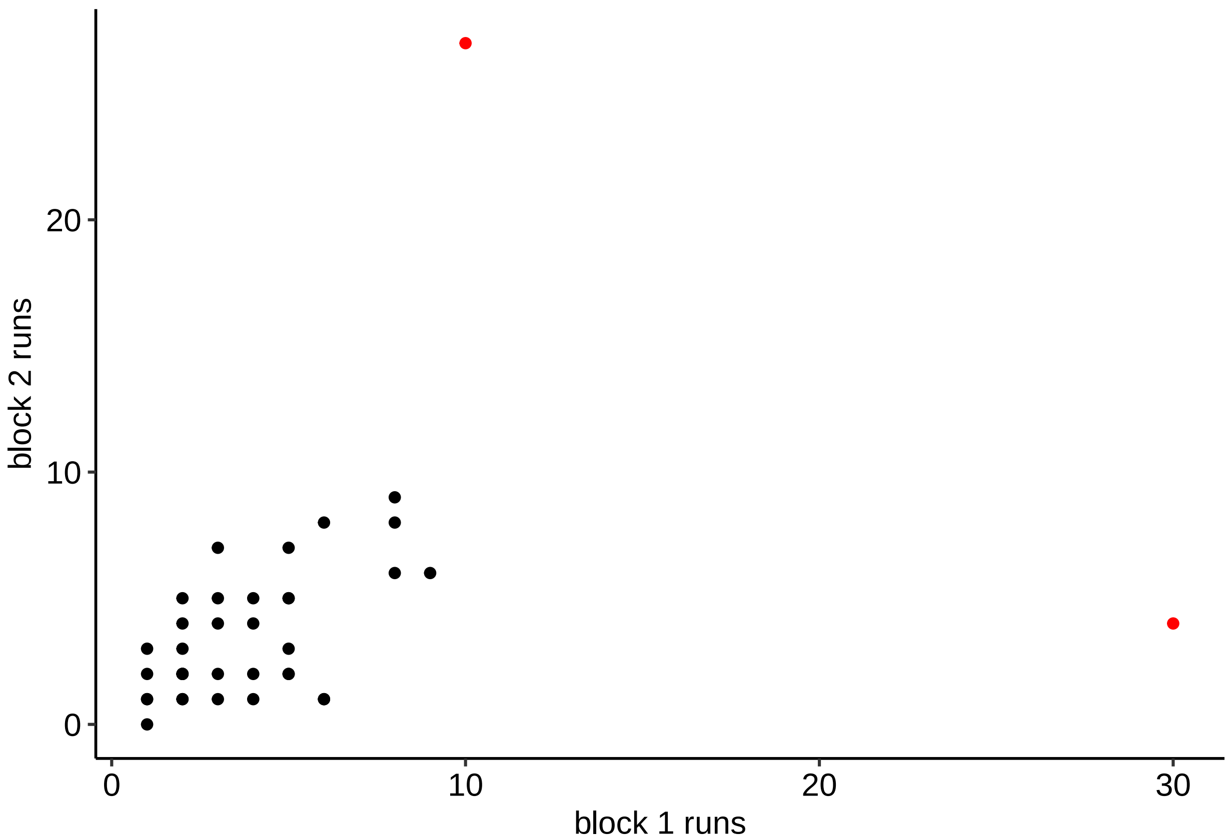


Figure 1: Maximum number of same consecutive responses in personality blocks.

(IDRIA) scale (Kay & Saucier, in prep) have been included here, in part to help evaluate the extent of inattentive responding but also to consider the effect of item wording on these items. The two items used here (i.e., "Asleep", "Human") were chosen to be as inconspicuous as possible, so as to not to inflate item response durations. The frequency item (i.e., "human") will be reverse-scored, so that higher scores on both the infrequency and frequency items reflect greater inattentive responding.

```
in_average = data %>%
  # reverse score human
  mutate(across(matches("^human"),  ~(.x*-1)+7)) %>%
  # select id and attention check items
  select(proid, matches("^human"), matches("^asleep")) %>%
  gather(item, response, -proid) %>%
  filter(!is.na(response)) %>%
  group_by(proid) %>%
  summarise(avg = mean(response)) %>%
  mutate(
    remove = case_when(
      avg >= 4 ~ "Remove",
      TRUE ~ "Keep"))
```

```
in_average %>%
  ggplot(aes(x = avg, fill = remove)) +
  geom_histogram(bins = 20, color = "white") +
  geom_vline(aes(xintercept = 4)) +
  guides(fill = "none") +
  labs(x = "Average response to inattention check items") +
  theme_pubr()
```

We remove 1 participants whose responses suggest inattention.

```
data = data %>%
  full_join(select(in_average, proid, remove)) %>%
  filter(remove != "Remove") %>%
  select(-remove)
```

Based on average time to respond to personality items

First, select just the timing of the personality items. We do this by searching for specific strings: "t_[someword]*[a or b or c or d]*(maybe 2_)_page_submit."

```
timing_data = data %>%
  select(proid, matches("t_[[:alpha:]]*_[abcd](_2)?_page_submit"))
```

Next we gather into long form and remove missing timing values

```
timing_data = timing_data %>%
  gather(variable, timing, -proid) %>%
  filter(!is.na(timing))

timing_data
```

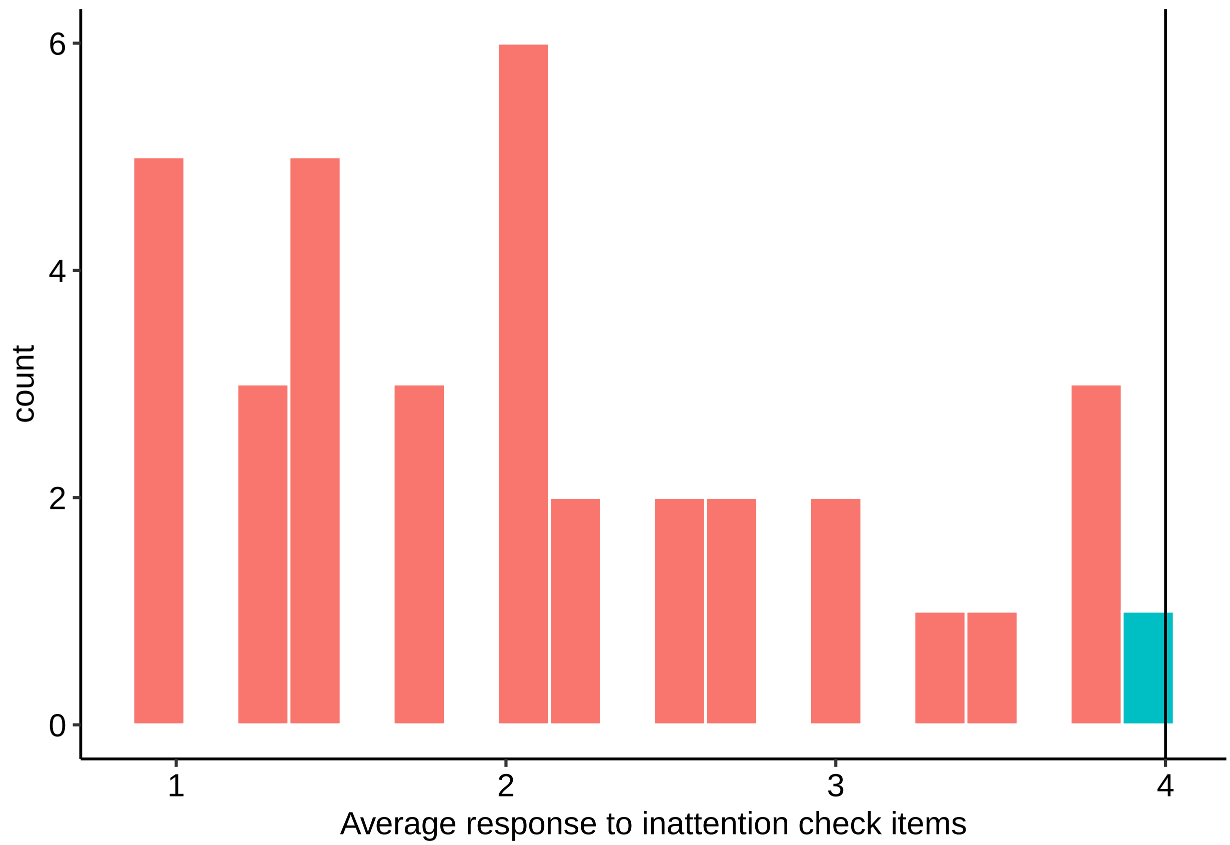


Figure 2: Average response to inattention check items

```
## # A tibble: 2,170 x 3
##    proid                    variable                 timing
##    <chr>                    <chr>                     <dbl>
##  1 5f0227ae81bf2f3a4618c8c7 t_outgoing_a_page_submit   4.47
##  2 60eb3434de863fc43f563b0e t_outgoing_a_page_submit   4.52
##  3 60eb31222e8b2fb8dc904432 t_outgoing_a_page_submit   4.86
##  4 60e49d66c9c4f08ce7a3b789 t_outgoing_a_page_submit   4.16
##  5 60ea7b0e32a76a57b4a34664 t_outgoing_a_page_submit   4.20
##  6 60e950d879a14636c5fc286d t_outgoing_a_page_submit   4.49
##  7 60e781742748ec6401b79f86 t_outgoing_a_page_submit   4.5
##  8 60e9521961c670e718bcc4df t_outgoing_a_page_submit  85.7
##  9 60e777356e13630d745eeb49 t_outgoing_a_page_submit   2.47
## 10 60e99999b101ef725cb0b8a2 t_outgoing_a_page_submit   4.45
## # ... with 2,160 more rows
```

To check, each participant should have the same number of responses: 62.

```
timing_data %>%
  group_by(proid) %>%
  count() %>%
  ungroup() %>%
  summarise(min(n), max(n))
```

```
## # A tibble: 1 x 2
##   `min(n)` `max(n)`
##      <int>    <int>
## 1       62       62
```

Excellent! Now we calculate the average response time per item for each participant. We mark a participant for removal if their average time is less than 1 second or greater than 30. See Figure @ref(fig:timing_dist) for a distribution of average response time.

```
timing_data = timing_data %>%
  group_by(proid) %>%
  summarise(m_time = mean(timing)) %>%
  mutate(remove = case_when(
    m_time < 1 ~ "Remove",
    m_time > 30 ~ "Remove",
    TRUE ~ "Keep"
  ))
```

```
timing_data %>%
  ggplot(aes(x = m_time, fill = remove)) +
  geom_histogram(color = "white") +
  labs(x = "Average response time (seconds)", y = "Number of participants") +
  theme_pubr()
```

```
data = inner_join(data, filter(timing_data, remove == "Keep")) %>%
  select(-remove)
```

Based on timing, we removed 0 participants.

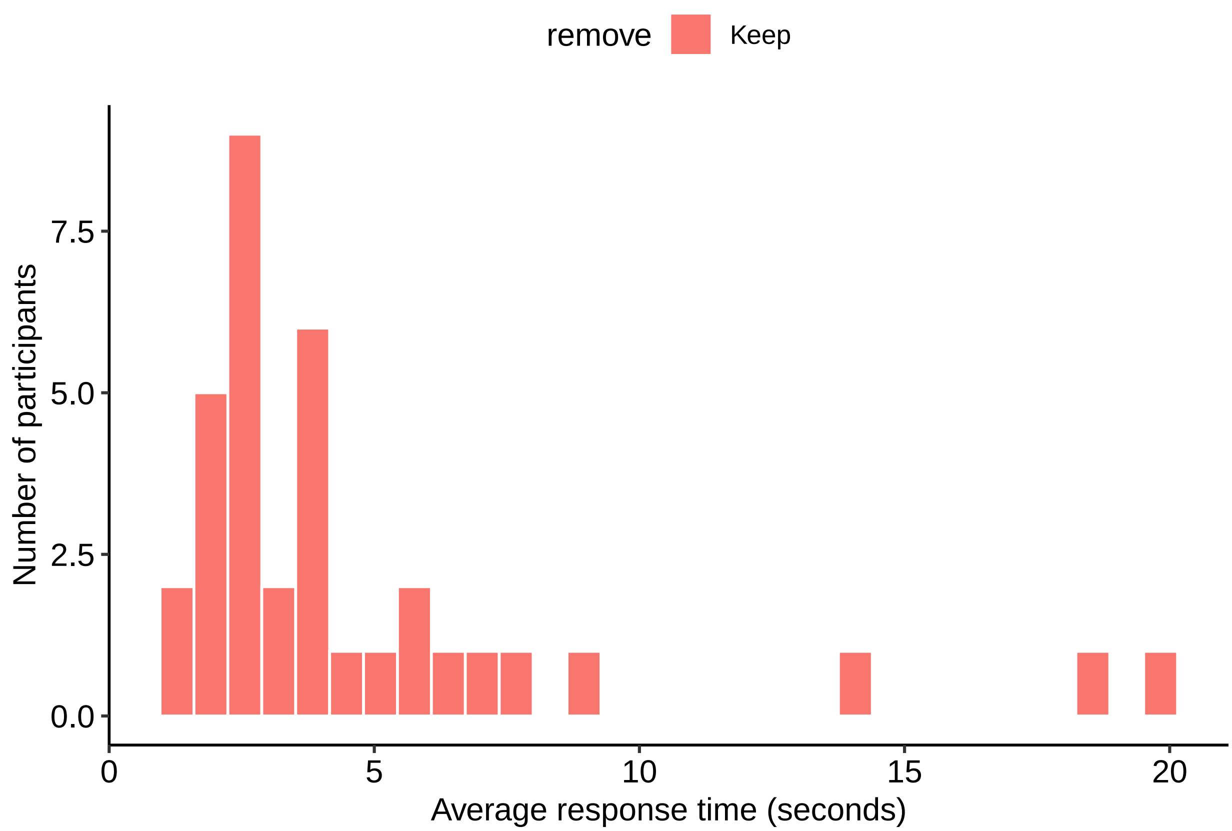


Figure 3: Distribution of average time to respond to personality items.

Reverse score personality items

The following items are (typically) negatively correlated with the others: reckless, moody, worrying, nervous, careless, impulsive. We reverse-score them to ease interpretation of associations and means in the later sections. In short, all traits will be scored such that larger numbers are indicative of the more socially desirable end of the specturm.

```
data = data %>%
  mutate(
    across(matches("^reckless"),  ~(.x*-1)+7),
    across(matches("^moody"),     ~(.x*-1)+7),
    across(matches("^worrying"),  ~(.x*-1)+7),
    across(matches("^nervous"),   ~(.x*-1)+7),
    across(matches("^careless"),  ~(.x*-1)+7),
    across(matches("^impulsive"), ~(.x*-1)+7))
```

Score memory task

Now we score the memory task. We start by creating vectors of the correct responses.

```
correct1 = c("book", "child", "gold", "hotel", "king",
             "market", "paper", "river", "skin", "tree")

correct2 = c("butter", "college", "dollar", "earth", "flag",
             "home", "machine", "ocean", "sky", "wife")

correct3 = c("blood", "corner", "engine", "girl", "house",
             "letter", "rock", "shoes", "valley", "woman")

correct4 = c("baby", "church", "doctor", "fire", "garden",
             "palace", "sea", "table", "village", "water")
```

Next we convert all responses to lowercase. Then we break the string of responses into a vector containing many strings.

```
data = data %>%
  mutate(
    across(matches("recall"),tolower), # convert to lower
    #replace carriage return with space
    across(matches("recall"), str_replace_all, pattern = "\\n", replacement = ","),
    # remove spaces
    across(matches("recall"), str_replace_all, pattern = " ", replacement = ","),
    # remove doubles
    across(matches("recall"), str_replace_all, pattern = ",,", replacement = ","),
    #remove last comma
    across(matches("recall"), str_remove, pattern = ",$"),
    # split the strings based on the spaces
    across(matches("recall"), str_split, pattern = ","))
```

Immediate recall

Now we use the `amatch` function in the `stringdist` package to look for exact (or close) matches to the target words. This function returns for each word either the position of the key in which you can find the target word or NA to indicate the word or a close match does not exist in the string.

```
distance = 1 #maximum distance between target word and correct response
data = data %>%
  mutate(
    memory1 = map(recall1, ~sapply(., amatch, correct1, maxDist = distance)),
    memory2 = map(recall2, ~sapply(., amatch, correct2, maxDist = distance)),
    memory3 = map(recall3, ~sapply(., amatch, correct3, maxDist = distance)),
    memory4 = map(recall4, ~sapply(., amatch, correct4, maxDist = distance))
    )
```

We count the number of correct answers. This gets complicated...

```
data = data %>%
  mutate(
    across(starts_with("memory"),
      #replace position with 1
      ~map(., sapply, FUN = function(x) ifelse(x >0, 1, 0))),
    across(starts_with("recall"),
          # are there non-missing values in the original response?
          ~map_dbl(.,
                   .f = function(x) sum(!is.na(x))),
          .names = "{.col}_miss"),
    across(starts_with("memory"),
      #replace position with 1
      # count the number of correct answers
      ~map_dbl(., sum, na.rm=T))) %>%
  mutate(
    memory1 = case_when(
      # if there were no resposes, make the answer NA
      recall1_miss == 0 ~ NA_real_,
      # otherwise, the number of correct guesses
      TRUE ~ memory1),
    memory2 = case_when(
      recall2_miss == 0 ~ NA_real_,
      TRUE ~ memory2),
    memory3 = case_when(
      recall3_miss == 0 ~ NA_real_,
      TRUE ~ memory3),
    memory4 = case_when(
      recall4_miss == 0 ~ NA_real_,
      TRUE ~ memory4)) %>%
  # no longer need the missing count variables
  select(-ends_with("miss"))
```

Finally, we want to go from 4 columns (one for each recall test), to two: one that has the number of correct resposnes, and one that indicates which version they saw.

```
data = data %>%
  select(proid, starts_with("memory")) %>%
  gather(mem_condition, memory, -proid) %>%
  filter(!is.na(memory)) %>%
  mutate(mem_condition = str_remove(mem_condition, "memory")) %>%
  full_join(data)
```

Participants remember on average 5.80 words correctly ($SD = 2.73$),

```r
data %>%
  ggplot(aes(x = memory)) +
  geom_histogram(bins = 11, color = "white") +
  labs(x = "Number of correct responses") +
  scale_x_continuous(breaks = 0:10) +
  theme_pubr()
```

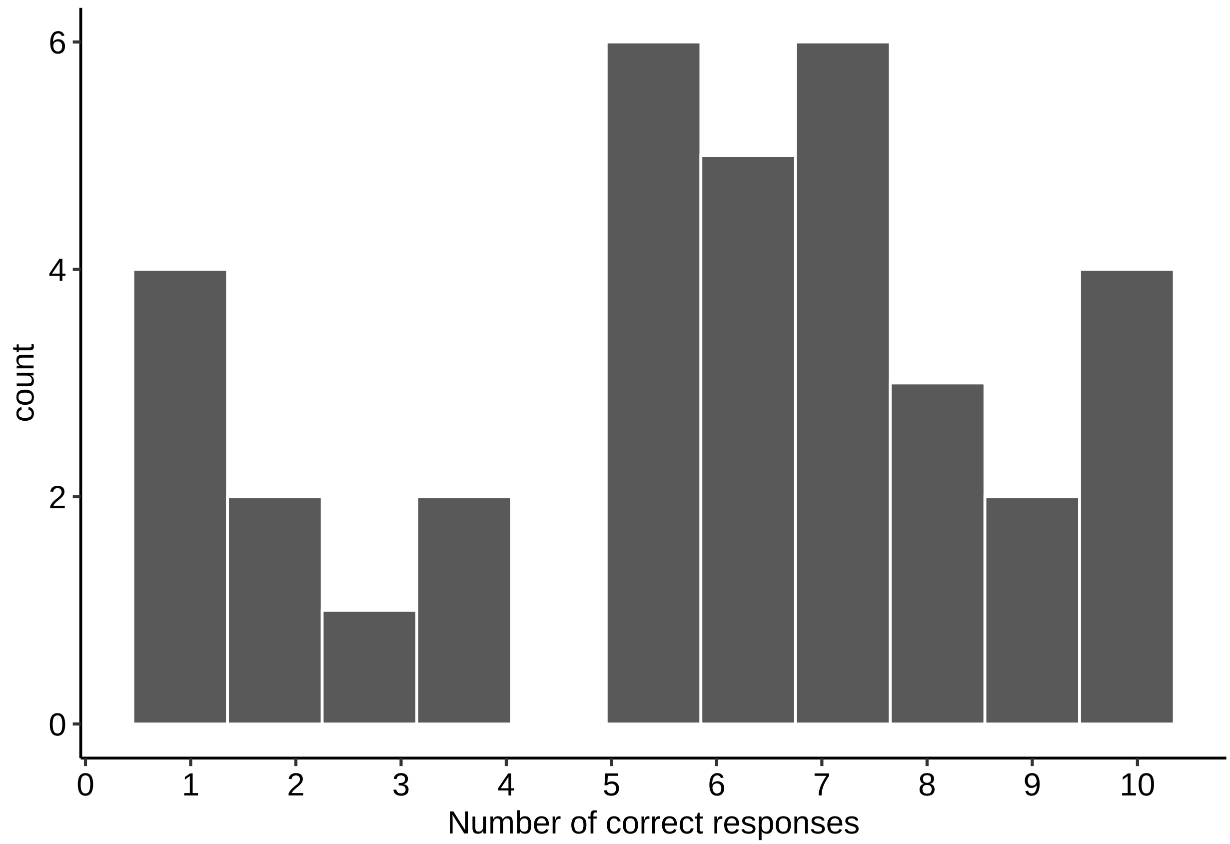


Figure 4: Correct responses on the memory task

```r
data %>%
  group_by(mem_condition) %>%
  summarise(
    m = mean(memory),
    s = sd(memory),
    min = min(memory),
    max = max(memory),
    n = n()
  ) %>%
  kable(booktabs = T,
        col.names = c("Condition", "Mean", "SD", "Min", "Max", "N"),
        digits = c(0, 2, 2, 1, 1, 1),
        caption = "Memory responses by condition") %>%
  kable_styling()
```

Table 1: Memory responses by condition

Condition	Mean	SD	Min	Max	N
1	5.50	2.56	1	10	8
2	4.62	3.20	1	10	8
3	6.40	2.72	1	10	10
4	6.44	2.51	2	10	9

Delayed recall

A challenge with the delayed recall task is identifying the memory condition that participants were assigned to, but this is made easier by the work done above.

```
mem2 = data %>%
  select(proid, mem_condition, delayed_recall) %>%
  mutate(newid = 1:nrow(.))

mem2 = mem2 %>%
  mutate(
    delayed_recall1 = map(delayed_recall, ~sapply(., amatch, correct1, maxDist = distance)),
    delayed_recall2 = map(delayed_recall, ~sapply(., amatch, correct2, maxDist = distance)),
    delayed_recall3 = map(delayed_recall, ~sapply(., amatch, correct3, maxDist = distance)),
    delayed_recall4 = map(delayed_recall, ~sapply(., amatch, correct4, maxDist = distance))
    ) %>%
  gather(variable, delayed_memory, delayed_recall1:delayed_recall4)

mem2 = mem2 %>%
  mutate(
      delayed_memory = map(delayed_memory, sapply,
                  FUN = function(x) ifelse(x >0, 1, 0)),
      # count the number of correct answers
      delayed_memory = map_dbl(delayed_memory, sum, na.rm=T))

mem2 = mem2 %>%
  group_by(proid) %>%
  filter(delayed_memory == max(delayed_memory)) %>%
  filter(row_number() == 1 ) %>%
  select(-delayed_recall, -variable)

data = inner_join(data, mem2)
```

```
data %>%
  ggplot(aes(x = delayed_memory)) +
  geom_histogram(color = "white", bins = 11) +
  scale_x_continuous("Number correct", breaks = c(0:10)) +
  labs(y = "Number of participants") +
  theme_pubr()
```

```
data %>%
  ggplot(aes(x = memory, y = delayed_memory)) +
  geom_point() +
```

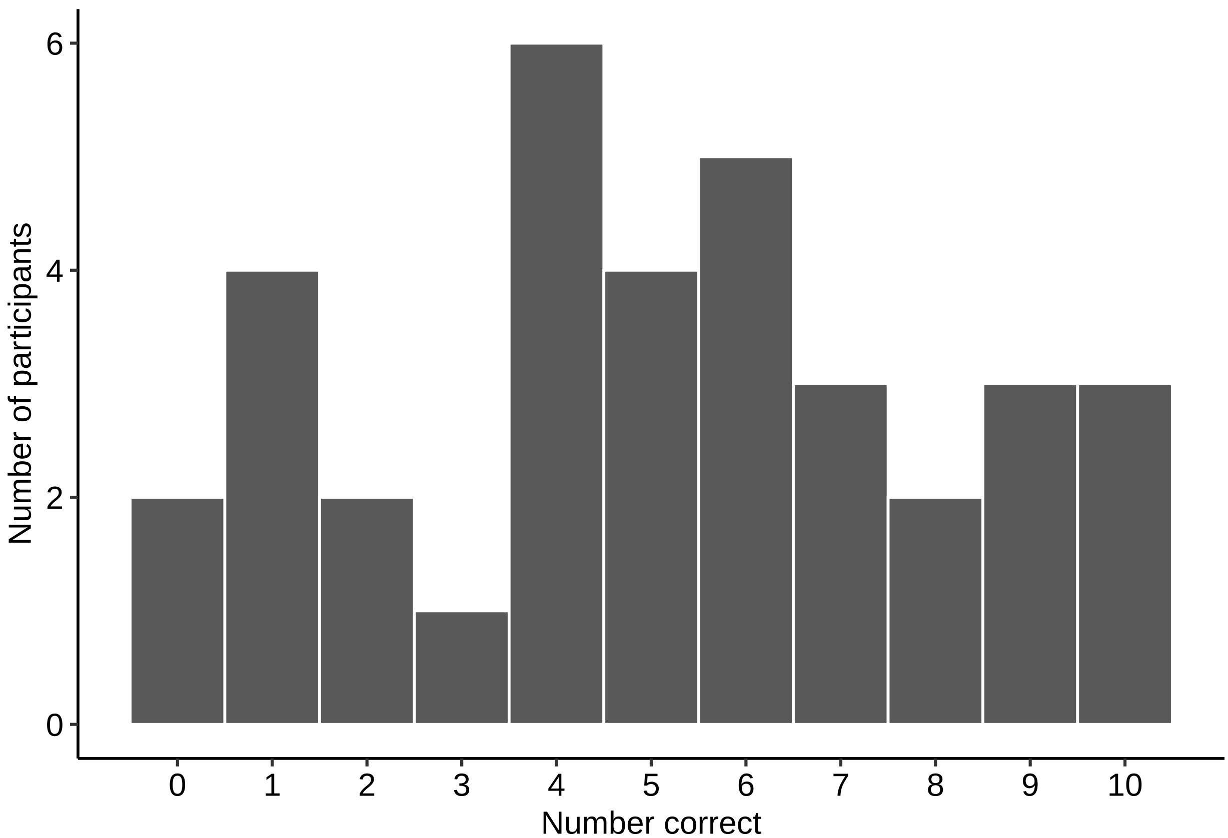


Figure 5: Distribution of delayed memory scores

```
  geom_smooth(method = "lm") +
  scale_x_continuous("Immediate number correct", breaks = c(0:10)) +
  scale_y_continuous("Delayed number correct", breaks = c(0:10)) +
  labs(title = paste0("r = ", printnum(cor(data$memory, data$delayed_memory, use = "pairwise")))) +
  theme_pubr()
```

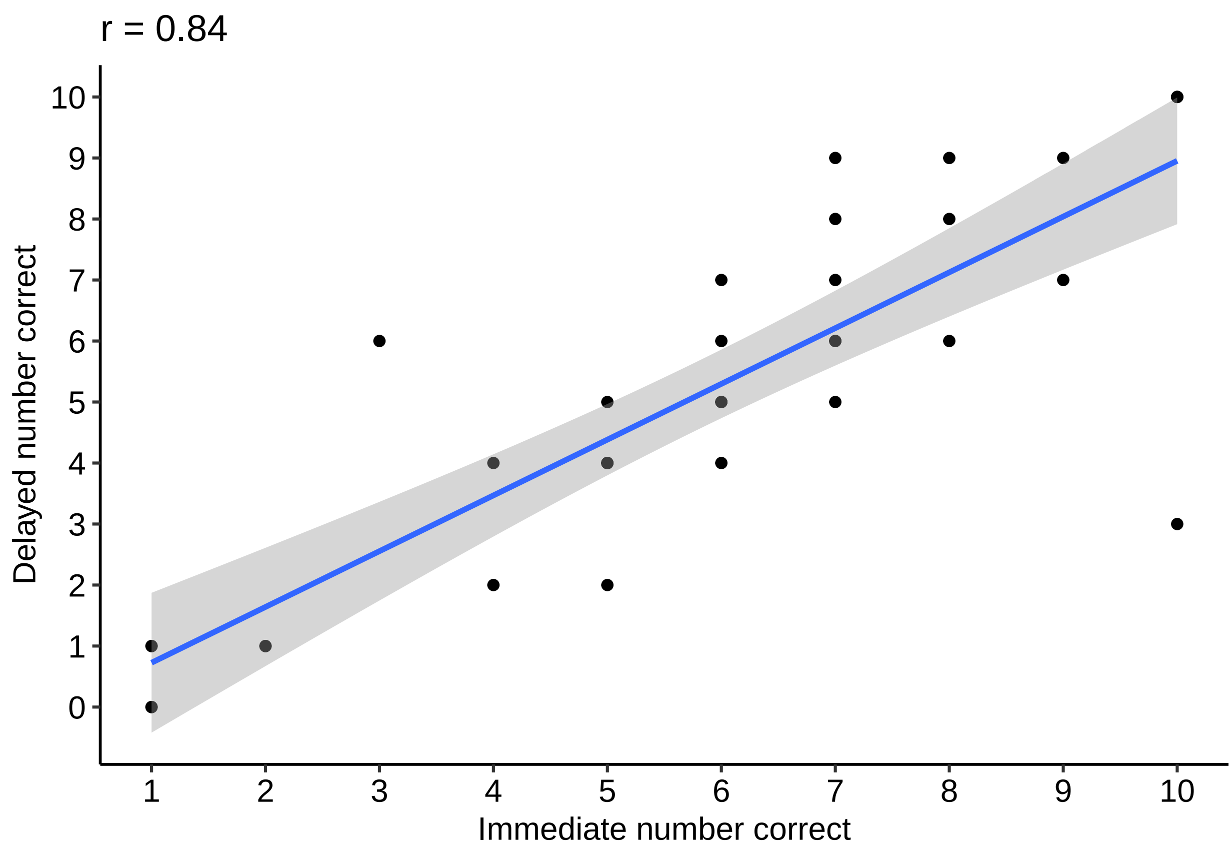


Figure 6: Relationship between immediate and delayed recall

Change labels of device variable

These labels are too long!

```
data = data %>%
  mutate(devicetype = factor(
    devicetype,
    levels = c("Desktop or laptop computer", "Mobile",
               "Tablet (for example, iPad, Galaxy Tablet, Amazon Fire, etc.)"), #what is this?!
    labels = c("Computer", "Mobile", "Tablet")
  ))
```

Long-form dataset

Item responses

We need one dataset that contains the responses to and timing of the personality items in long form. This will be used for nearly all the statistical models, which will nest items within person. To create this, we first select the responses to the items of different formats. For this set of analyses, we use data collected in both Block 1 and Block 2 – that is, each participant saw the same format for every item during Block 1, but a random format for each item in Block 2.

These variable names have one of two formats: `[trait]_[abcd]` (for example, `talkative_a`) or `[trait]_[abcd]_2` (for example, `talkative_a_2`). We search for these items using regular expressions.

```
item_responses = str_subset(
  names(data),
  "^([[:alpha:]])+_[abcd](_2)?$"
)
```

Similarly, we'll need to know how long it took participants to respond to these items. These variable names have one of two formats: `t_[trait]_[abcd]_page_submit` (for example, `t_talkative_a_page_submit`) or `t_[trait]_[abcd]_2_page_submit` (for example, `t_talkative_a_2_page_submit`). We search for these items using regular expressions.

```
item_timing = str_subset(
  names(data),
  "^t_([[:alpha:]])+_[abcd]_(2_)?page_submit$"
)
```

We extract just the participant IDs, delayed memory, and these variables.

```
items_df = data %>%
  select(proid, delayed_memory, all_of(item_responses), all_of(item_timing))
```

Next we reshape these data into long form. This requires several steps. We'll need to identify whether each value is a response or timing; we can use the presence of the string `t_` for this. Next, we'll identify the block based on whether the string contains `_2`. Then, we identify the condition based on which letter (`a`, `b`, `c`, or `d`) follows an underscore. Throughout, we'll strip the item string of extraneous information until we're left with only the adjective assessed. Finally, we'll use spread to create separate columns for the response and the timing variables.

```
items_df = items_df %>%
  gather(item, value, all_of(item_responses), all_of(item_timing)) %>%
  filter(!is.na(value)) %>%
  # identify whether timing or response
  mutate(variable = ifelse(str_detect(item, "^t_"), "timing", "response"),
         item = str_remove(item, "^t_"),
         item = str_remove(item, "_page_submit$")) %>%
  #identify block
  mutate(block = ifelse(str_detect(item, "_2"), "2", "1"),
         item = str_remove(item, "_2")) %>%
  separate(item, into = c("item", "format")) %>%
  spread(variable, value)
```

We also remove responses to the adjectives “human” and “asleep”, as these are not personality items per-se and included for the purpose of attention checks.

```
items_df = items_df %>%
  filter(item != "human") %>%
  filter(item != "asleep")
```

We give labels to the formats, to clarify interpretations and aid table and figure construction.

```
items_df$format = as.factor(items_df$format)
items_df$format = relevel(items_df$format, ref = "a")
items_df$format = factor(items_df$format,
                          levels = c("a","b","c","d"),
                          labels = c("Adjective\nOnly", "Am\nAdjective", "Tend to be\nAdjective",
```

Transform seconds

The variable `seconds` appears to have a very severe right skew. We log-transform this variable for later analyses.

```
items_df = items_df %>%
  mutate(seconds_log = log(timing))
```

```
items_df %>%
  gather(variable, value, timing, seconds_log) %>%
  mutate(variable = factor(variable,
                           levels = c("timing", "seconds_log"),
                           labels = c("Seconds (raw)", "Seconds (log)"))) %>%
  ggplot(aes(x = value)) +
  geom_histogram(bins = 100) +
  facet_wrap(~variable, scales = "free") +
  labs(x = NULL, y = "Number of participants") +
  theme_pubr()
```

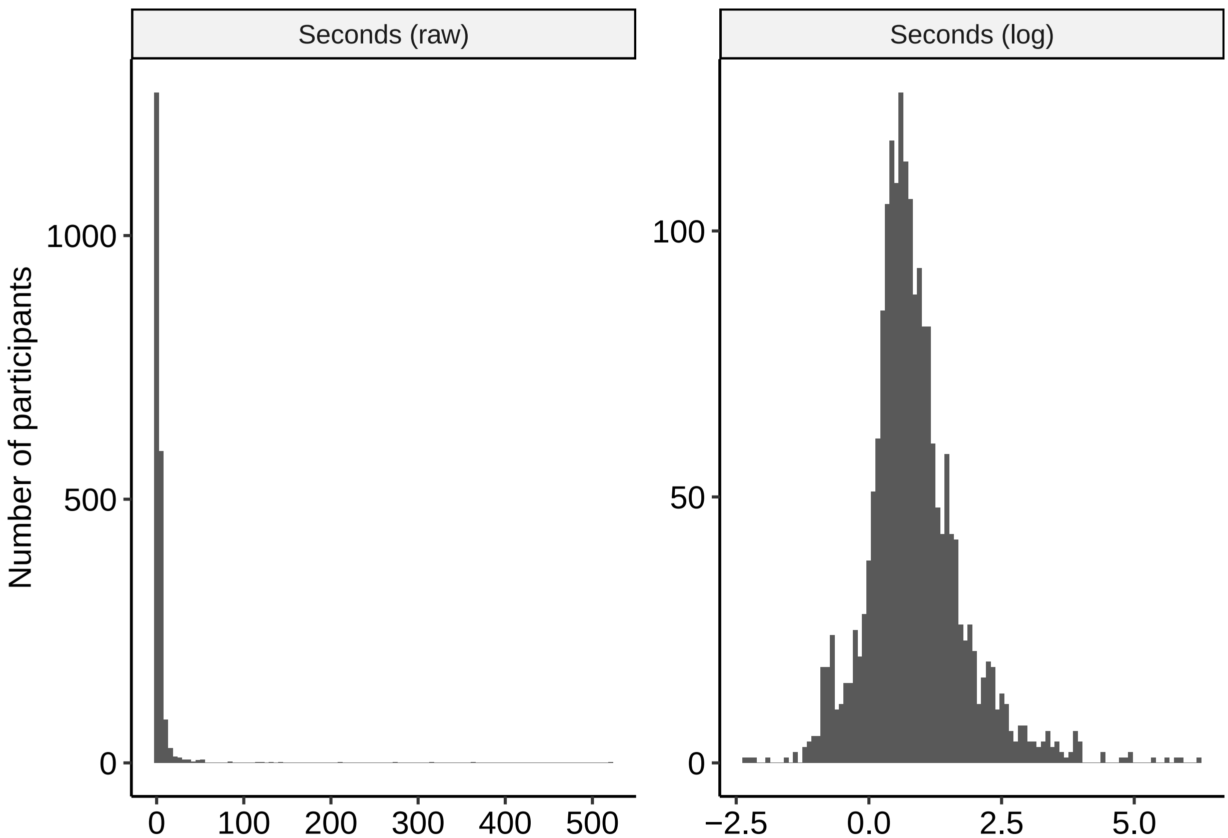


Figure 7: Distribution of seconds, raw and transformed.

Does item format affect response?

The primary aims of this study are to evaluate the effects of item wording in online, self-report personality assessment. Specifically, we intend to consider the extent to which incremental wording changes may influence differences in the distributions of responses, response times, and psychometric properties of the items. These wording changes will include a progression from using (1) trait-descriptive adjectives by themselves, (2) with the linking verb "to be" (Am. . .), (3) with the additional verb "to tend" (Tend to be. . .), and (4) with the pronoun "someone" (Am someone who tends to be. . .).

Using a protocol that administers each adjective twice to the same participant (in different combinations of item format administered randomly across participants), we will use between-person analyses to compare responses using group-level data for the different formats.

These analyses will attempt to account for delayed_memory effects by collecting data on immediate and delayed recall (5 minutes and approximately two weeks) using a delayed_memory paradigm that was developed based on a similar recall task used in the HRS (Runge et al., 2015).

Analysis: Block 1 data only

We used a multilevel model, nesting response within participant to account for dependence. Our primary predictor was format. Here, we use only Block 1 data.

```
item_block1 = filter(items_df, block == "1")

mod.format_b1 = lmer(response~format + (1|proid),
                  data = item_block1)
anova(mod.format_b1)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##        Sum Sq Mean Sq NumDF DenDF F value Pr(>F)
## format 1.7991  0.5997     3    31  0.3946 0.7577
```

```
plot_b1 = plot_model(mod.format_b1, type = "pred")

plot_b1$format +
  labs(x = NULL,
       y = "Average response",
       title = "Average responses by item formatting (Block 1 Data)") +
  theme_pubclean()
```

```
means_by_group = item_block1 %>%
  group_by(format) %>%
  summarise(m = mean(response),
            s = sd(response))

item_block1 %>%
  ggplot(aes(x = response, fill = format)) +
  geom_histogram(bins = 6, color = "white") +
  geom_vline(aes(xintercept = m), data = means_by_group) +
  geom_text(aes(x = 1,
                y = 125,
                label = paste("M =", round(m,2),
```

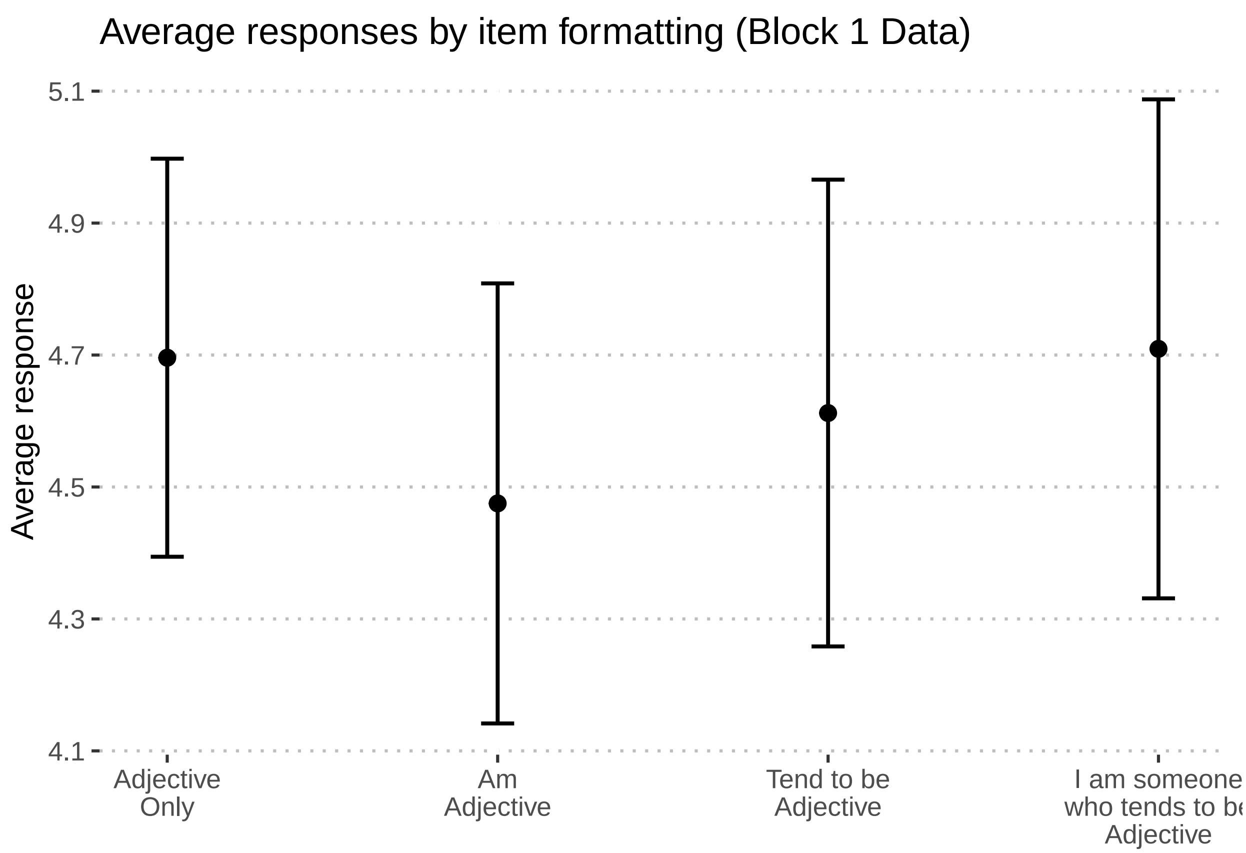


Figure 8: Predicted response on personality items by condition, using only Block 1 data.

```
                                    "\nSD =", round(s,2))),
            data = means_by_group,
            hjust =0,
            vjust = 1) +
  facet_wrap(~format) +
  guides(fill = "none") +
  scale_x_continuous(breaks = 1:6) +
  labs(y = "Number of particpants",
       title = "Distribution of responses by format (Block 1 data)") +
  theme_pubr()
```

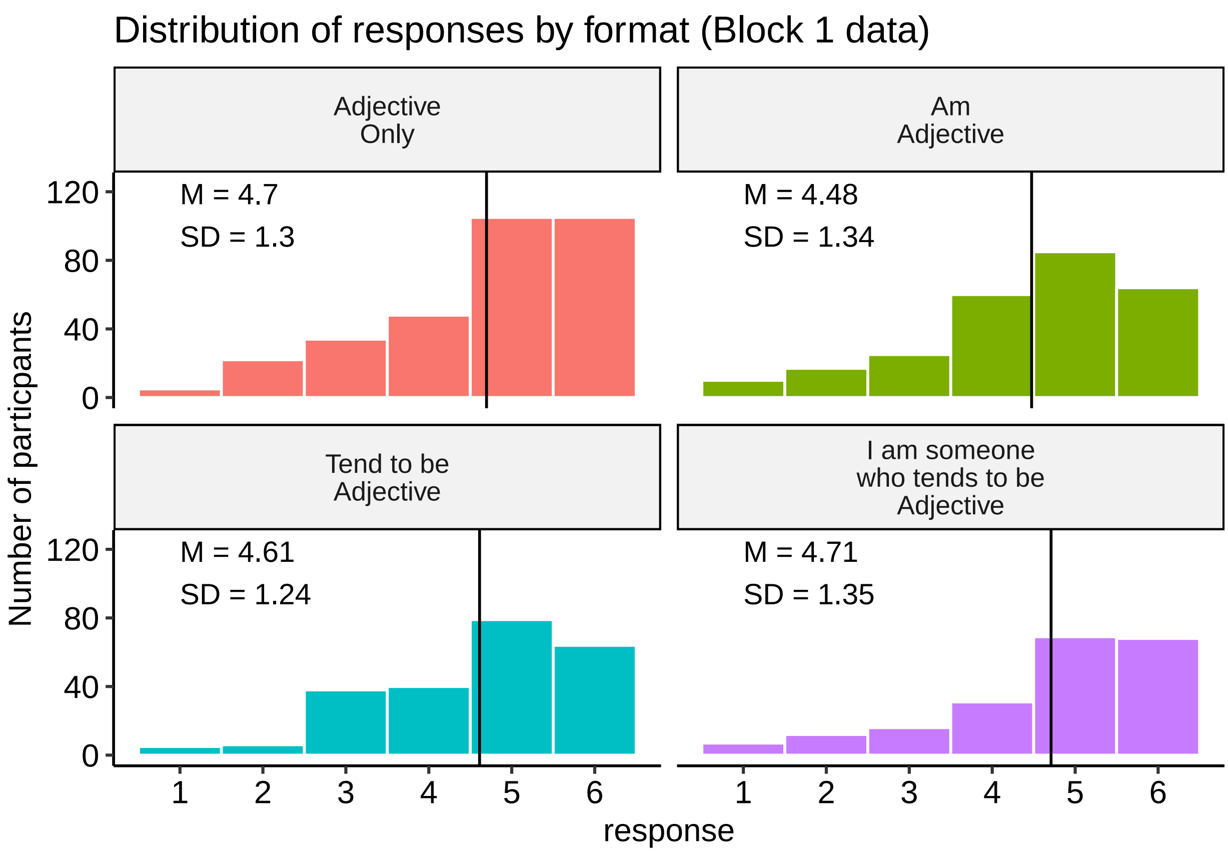


Figure 9: Distribution of responses by category, block 1 data only

One model for each adjective

We can also repeat this analysis separately for each trait.

```
mod_by_item_b1 = item_block1 %>%
  group_by(item) %>%
  nest() %>%
  mutate(mod = map(data, ~lm(response~format, data = .))) %>%
  mutate(aov = map(mod, anova))

summary_by_item_b1 = mod_by_item_b1 %>%
```

Table 2: Format effects on response by item (block 1 data only)

item	sumsq	meansq	statistic	p.value	p.adj
active	2.80	0.93	0.61	0.61	0.61
adventurous	1.34	0.45	0.50	0.68	0.68
calm	3.77	1.26	1.29	0.30	0.30
careless	18.23	6.08	2.77	0.06	0.06
caring	2.47	0.82	1.17	0.34	0.34
cautious	1.55	0.52	0.32	0.81	0.81
creative	2.35	0.78	0.79	0.51	0.51
curious	0.52	0.17	0.15	0.93	0.93
friendly	0.52	0.17	0.34	0.80	0.80
hardworking	0.03	0.01	0.02	1.00	1.00
helpful	0.20	0.07	0.08	0.97	0.97
imaginative	1.40	0.47	0.58	0.63	0.63
impulsive	2.65	0.88	0.45	0.72	0.72
intelligent	1.86	0.62	0.44	0.73	0.73
lively	1.15	0.38	0.36	0.78	0.78
moody	6.21	2.07	1.14	0.35	0.35
nervous	1.54	0.51	0.20	0.89	0.89
organized	2.20	0.73	0.66	0.58	0.58
outgoing	1.58	0.53	0.40	0.76	0.76
reckless	5.14	1.71	0.65	0.59	0.59
responsible	0.87	0.29	0.65	0.59	0.59
softhearted	1.25	0.42	0.30	0.83	0.83
sophisticated	1.08	0.36	0.21	0.89	0.89
sympathetic	3.42	1.14	1.10	0.36	0.36
talkative	18.53	6.18	3.19	0.04	0.04
thorough	0.33	0.11	0.08	0.97	0.97
thrifty	8.09	2.70	2.06	0.13	0.13
warm	0.71	0.24	0.20	0.90	0.90
worrying	8.95	2.98	1.25	0.31	0.31

```
  mutate(tidy = map(aov, broom::tidy)) %>%
  select(item, tidy) %>%
  unnest(cols = c(tidy)) %>%
  filter(term == "format") %>%
  select(-term, -df) %>%
  mutate(p.adj = p.adjust(p.value, method = "holm"))

summary_by_item_b1 %>%
  mutate(across(
    starts_with("p"),
    papaja::printnum
  )) %>%
  kable(digits = 2, booktabs = T, caption = "Format effects on response by item (block 1 data only)") %
  kable_styling()
```

Table 3: Differences in response to Talkative by format (Block 1 data only)

Contrast	Difference in means	SE	df	t	p
Adjective Only - Am Adjective	-1.49	0.63	31	-2.39	0.139
Adjective Only - Tend to be Adjective	-1.40	0.65	31	-2.16	0.192
Adjective Only - I am someone who tends to be Adjective	0.01	0.67	31	0.02	1.000
Am Adjective - Tend to be Adjective	0.10	0.68	31	0.14	1.000
Am Adjective - I am someone who tends to be Adjective	1.51	0.70	31	2.15	0.192
Tend to be Adjective - I am someone who tends to be Adjective	1.41	0.72	31	1.96	0.192

Pairwise t-tests for significant ANOVAs

Here we identify the specific items with significant differences.

```
sig_item_b1 = summary_by_item_b1 %>%
  filter(p.value < .05)

sig_item_b1 = sig_item_b1$item
sig_item_b1
```

```
## [1] "talkative"
```

Then we create models for each adjective. We use the `emmeans` package to perform pairwise comparisons, again with a Holm correction on the *p*-values. We also plot the means and 95% confidence intervals of each mean.

This code will have to be changed after final data collection. It is not self-adapting!

Talkative

```
talkative_model_b1 = item_block1 %>%
  filter(item == "talkative") %>%
  lm(response~format, data = .)

talkative_em_b1 = emmeans(talkative_model_b1, "format")
pairs(talkative_em_b1, adjust = "holm") %>%
  kable(booktabs = T,
        digits = c(0,2,2,2,2,3),
        caption = "Differences in response to Talkative by format (Block 1 data only)",
        col.names = c("Contrast", "Difference in means", "SE", "df", "t", "p")) %>%
  kable_styling()
```

```
plot_model(talkative_model_b1, type = "pred", terms = c("format"))
```

Analysis: Block 1 and Block 2

We used a multilevel model, nesting response within participant to account for dependence. Our primary predictor was format. Here, we use only Block 1 data.

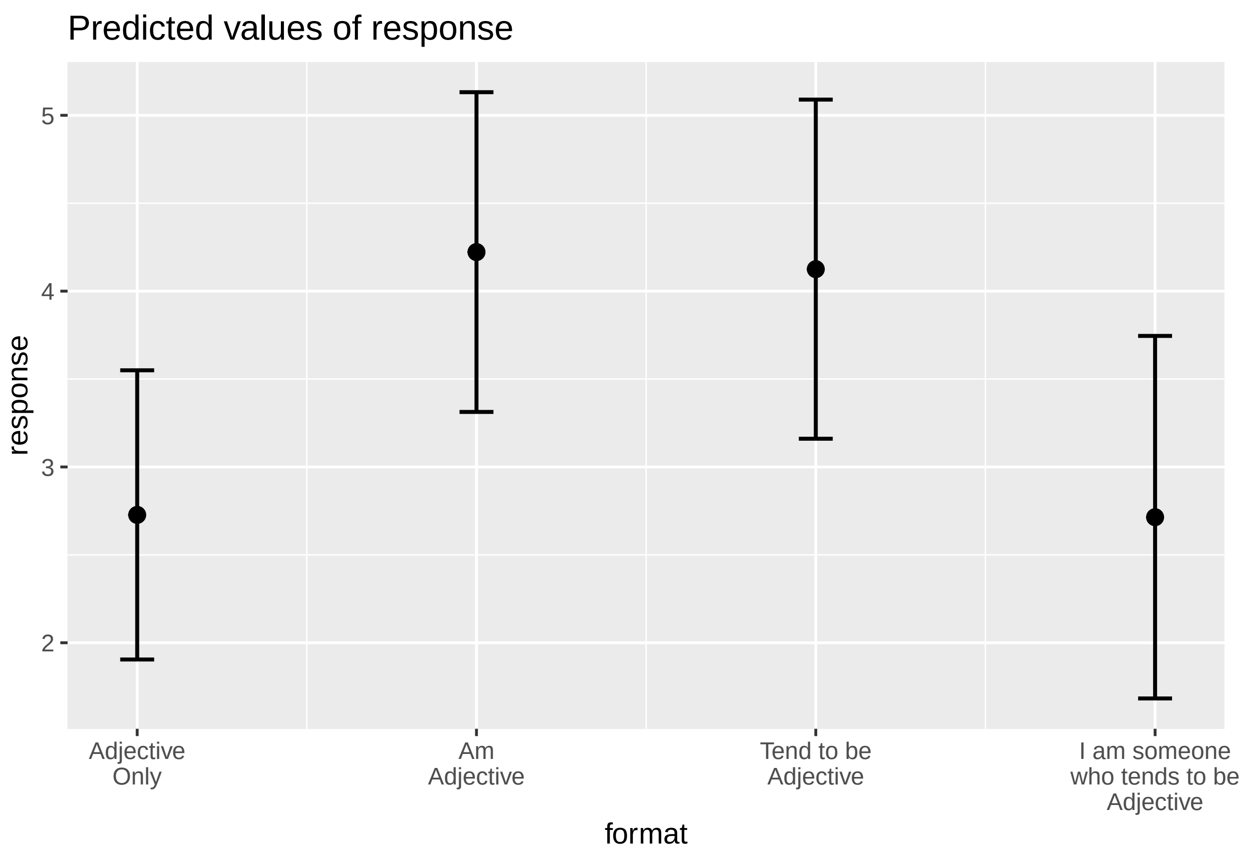


Figure 10: Average response to “talkative” by format (block 1 data only)

```
mod.format_b2 = lmer(response~format + (1|proid),
                  data = items_df)
anova(mod.format_b2)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##        Sum Sq Mean Sq NumDF  DenDF F value Pr(>F)
## format 4.0137  1.3379     3 1959.5  0.8666 0.4577
```

```
plot_b2 = plot_model(mod.format_b2, type = "pred")

plot_b2$format +
  labs(x = NULL,
       y = "Average response",
       title = "Average responses by item formatting (Block 1 and Block 2)") +
  theme_pubclean()
```

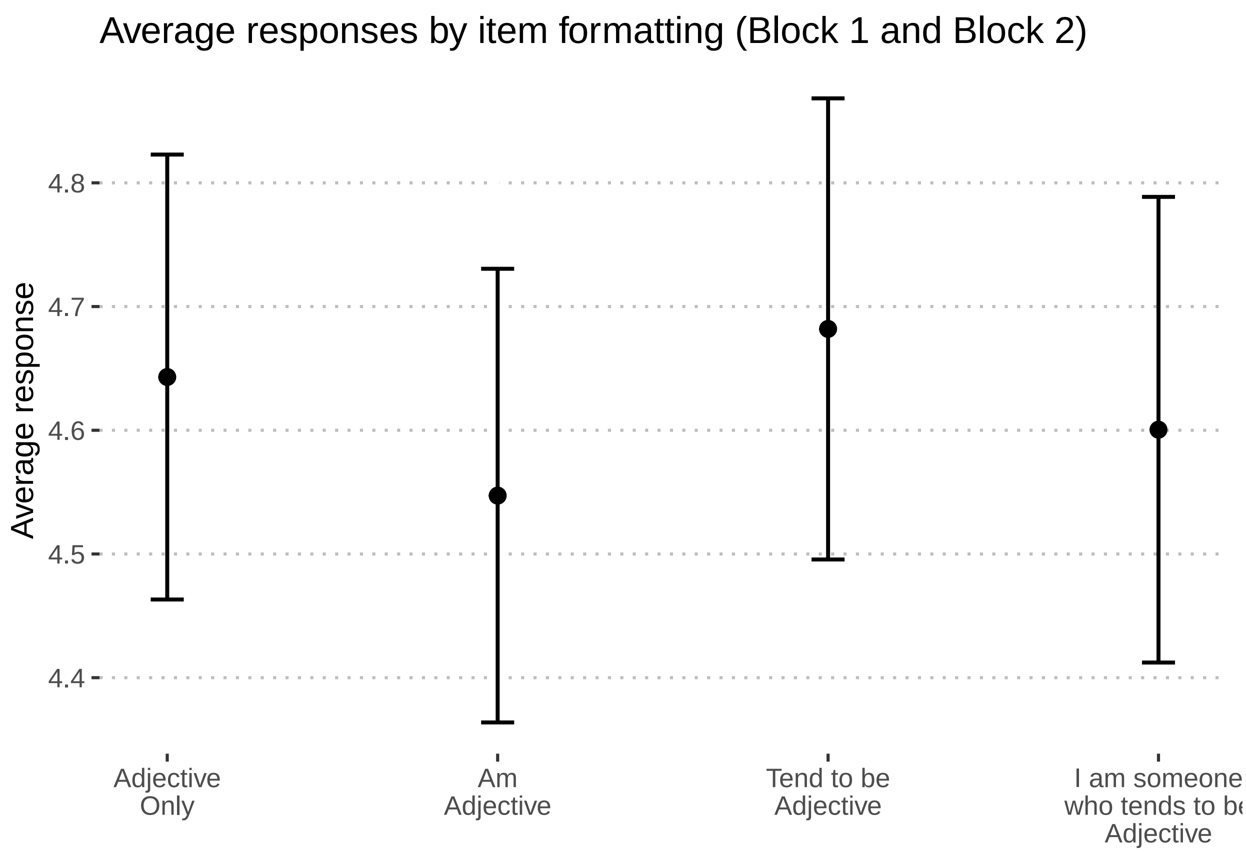


Figure 11: Predicted response on personality items by condition, using only Block 1 data.

```
means_by_group = items_df %>%
  group_by(format) %>%
  summarise(m = mean(response),
            s = sd(response))
```

```
items_df %>%
  ggplot(aes(x = response, fill = format)) +
  geom_histogram(bins = 6, color = "white") +
  geom_vline(aes(xintercept = m), data = means_by_group) +
  geom_text(aes(x = 1,
                y = 200,
                label = paste("M =", round(m,2),
                              "\nSD =", round(s,2))),
            data = means_by_group,
            hjust =0,
            vjust = 1) +
  facet_wrap(~format) +
  guides(fill = "none") +
  scale_x_continuous(breaks = 1:6) +
  labs(y = "Number of particpants",
       title = "Distribution of responses by format (Block 1 and Block 2)") +
  theme_pubr()
```

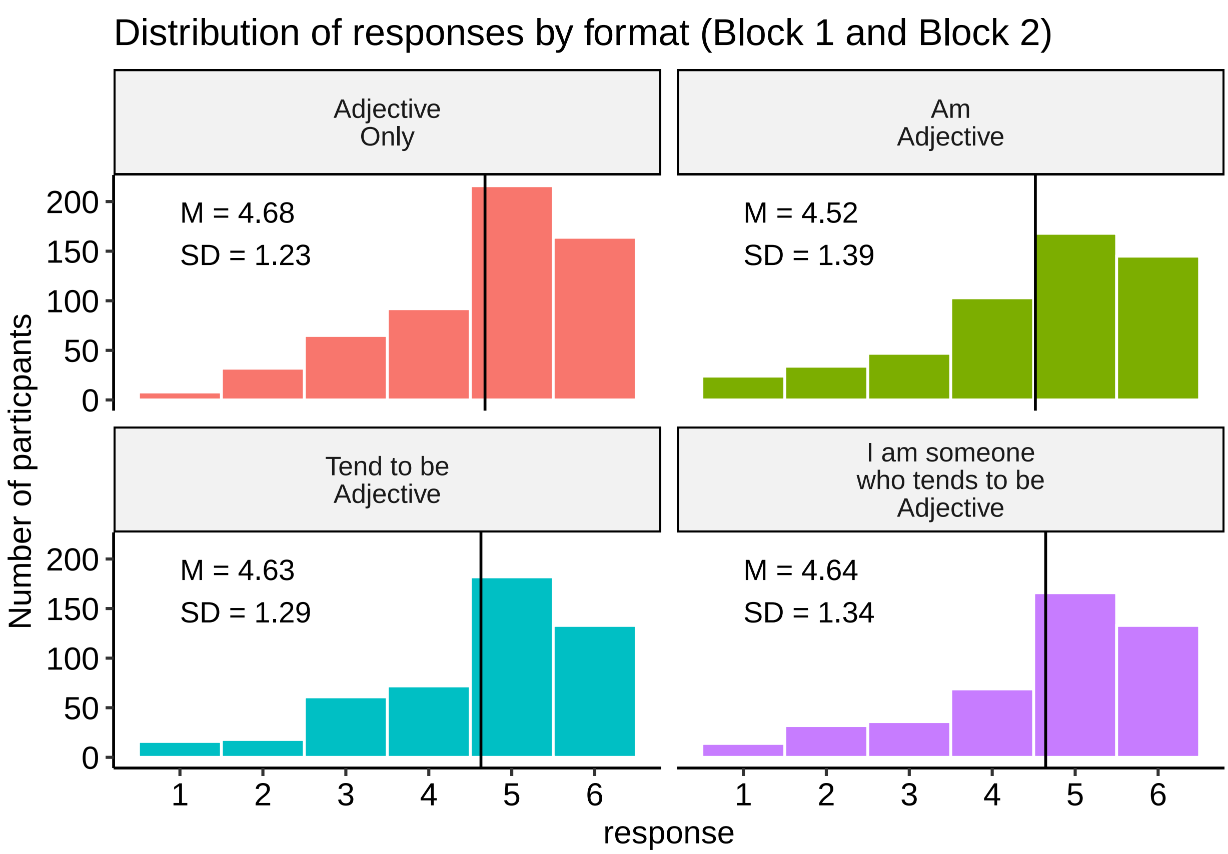


Figure 12: Distribution of responses by category, block 1 and block 2

One model for each adjective

We can also repeat this analysis separately for each trait.

```
mod_by_item_b2 = items_df %>%
  group_by(item) %>%
  nest() %>%
  mutate(mod = map(data, ~lmer(response~format + (1|proid), data = .))) %>%
  mutate(aov = map(mod, anova))

summary_by_item_b2 = mod_by_item_b2 %>%
  mutate(tidy = map(aov, broom::tidy)) %>%
  select(item, tidy) %>%
  unnest(cols = c(tidy)) %>%
  filter(term == "format") %>%
  select(-term) %>%
  mutate(p.adj = p.adjust(p.value, method = "holm"))

summary_by_item_b2 %>%
  mutate(across(
    starts_with("p"),
    papaja::printnum
  )) %>%
  kable(digits = 2, booktabs = T, caption = "Format effects on response by item (block 1 data only)") %
  kable_styling()
```

Pairwise t-tests for significant ANOVAs

Here we identify the specific items with significant differences.

```
sig_item_b2 = summary_by_item_b2 %>%
  filter(p.value < .05)

sig_item_b2 = sig_item_b2$item
sig_item_b2
```

```
## [1] "careless" "thrifty"
```

Then we create models for each adjective. We use the `emmeans` package to perform pairwise comparisons, again with a Holm correction on the *p*-values. We also plot the means and 95% confidence intervals of each mean.

This code will have to be changed after final data collection. It is not self-adapting!

Careless

```
careless_model_b2 = items_df %>%
  filter(item == "careless") %>%
  lmer(response~format + (1|proid),
                               data = .)

careless_em_b2 = emmeans(careless_model_b2, "format")
pairs(careless_em_b2, adjust = "holm") %>%
  kable(booktabs = T,
```

Table 4: Format effects on response by item (block 1 data only)

item	sumsq	meansq	NumDF	DenDF	statistic	p.value	p.adj
active	0.76	0.25	3	40.00	1.01	0.40	0.40
adventurous	1.51	0.50	3	48.65	1.09	0.36	0.36
calm	0.13	0.04	3	45.02	0.13	0.94	0.94
careless	8.29	2.76	3	53.32	2.84	0.05	0.05
caring	1.16	0.39	3	52.66	1.28	0.29	0.29
cautious	2.30	0.77	3	54.97	0.98	0.41	0.41
creative	0.14	0.05	3	49.22	0.13	0.94	0.94
curious	2.35	0.78	3	52.00	1.05	0.38	0.38
friendly	0.94	0.31	3	46.85	1.45	0.24	0.24
hardworking	2.10	0.70	3	52.03	2.06	0.12	0.12
helpful	0.90	0.30	3	60.54	0.82	0.49	0.49
imaginative	1.03	0.34	3	55.07	0.86	0.47	0.47
impulsive	1.38	0.46	3	41.41	0.85	0.47	0.47
intelligent	1.72	0.57	3	51.16	1.15	0.34	0.34
lively	1.08	0.36	3	42.77	0.89	0.46	0.46
moody	1.21	0.40	3	42.77	0.69	0.57	0.57
nervous	1.35	0.45	3	46.22	0.45	0.72	0.72
organized	0.22	0.07	3	37.39	0.41	0.75	0.75
outgoing	0.71	0.24	3	39.34	0.83	0.48	0.48
reckless	1.14	0.38	3	48.60	0.36	0.78	0.78
responsible	1.51	0.50	3	56.50	0.82	0.49	0.49
softhearted	1.95	0.65	3	57.46	0.74	0.53	0.53
sophisticated	0.05	0.02	3	44.04	0.04	0.99	0.99
sympathetic	0.49	0.16	3	51.48	0.51	0.68	0.68
talkative	6.36	2.12	3	43.86	1.95	0.13	0.13
thorough	2.23	0.74	3	40.16	2.72	0.06	0.06
thrifty	6.39	2.13	3	51.69	3.30	0.03	0.03
warm	0.10	0.03	3	52.22	0.08	0.97	0.97
worrying	1.72	0.57	3	41.67	0.63	0.60	0.60

Table 5: Differences in response to Careless by format (Block 1 and Block 2)

Contrast	Difference in means	SE	df	t	p
Adjective Only - Am Adjective	0.82	0.45	58.04	1.83	0.359
Adjective Only - Tend to be Adjective	0.14	0.44	54.32	0.32	0.749
Adjective Only - I am someone who tends to be Adjective	-0.44	0.44	54.32	-1.01	0.633
Am Adjective - Tend to be Adjective	-0.68	0.45	51.43	-1.52	0.542
Am Adjective - I am someone who tends to be Adjective	-1.26	0.45	51.43	-2.82	0.041
Tend to be Adjective - I am someone who tends to be Adjective	-0.58	0.45	49.48	-1.31	0.591

```
      digits = c(0,2,2,2,2,3),
      caption = "Differences in response to Careless by format (Block 1 and Block 2)",
      col.names = c("Contrast", "Difference in means", "SE", "df", "t", "p")) %>%
  kable_styling()
```

```
plot_model(careless_model_b2, type = "pred", terms = c("format"))
```

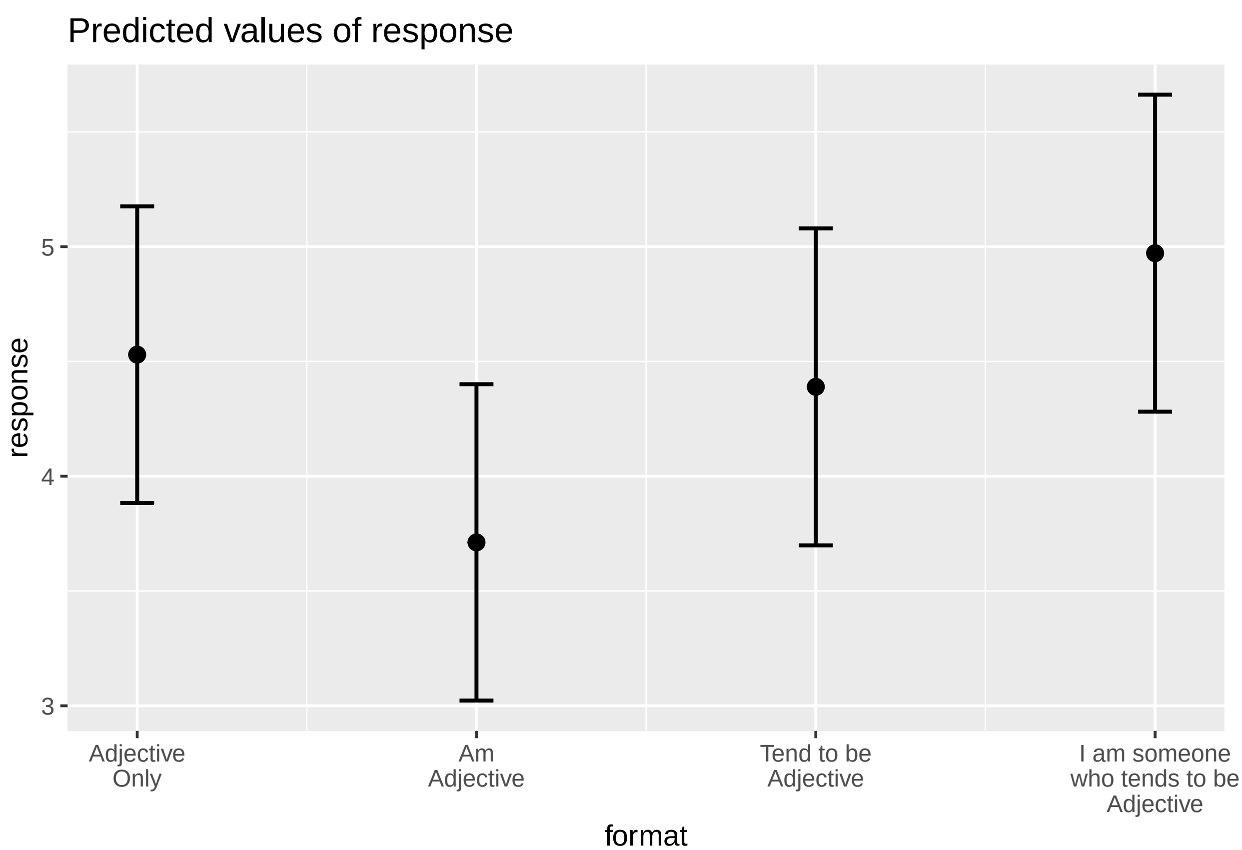


Figure 13: Average response to “careless” by format (Block 1 and Block 2)

Table 6: Differences in response to Thifty by format (Block 1 and Block 2)

Contrast	Difference in means	SE	df	t	p
Adjective Only - Am Adjective	-0.02	0.36	56.66	-0.04	1.000
Adjective Only - Tend to be Adjective	-0.92	0.36	55.47	-2.54	0.083
Adjective Only - I am someone who tends to be Adjective	-0.09	0.35	52.28	-0.26	1.000
Am Adjective - Tend to be Adjective	-0.90	0.37	56.24	-2.45	0.084
Am Adjective - I am someone who tends to be Adjective	-0.08	0.37	54.54	-0.21	1.000
Tend to be Adjective - I am someone who tends to be Adjective	0.83	0.33	43.22	2.49	0.084

Thrifty

```
thrifty_model_b2 = items_df %>%
  filter(item == "thrifty") %>%
  lmer(response~format + (1|proid),
                              data = .)

thrifty_em_b2 = emmeans(thrifty_model_b2, "format")
pairs(thrifty_em_b2, adjust = "holm") %>%
  kable(booktabs = T,
        digits = c(0,2,2,2,2,3),
        caption = "Differences in response to Thifty by format (Block 1 and Block 2)",
        col.names = c("Contrast", "Difference in means", "SE", "df", "t", "p")) %>%
  kable_styling()
```

```
plot_model(thrifty_model_b2, type = "pred", terms = c("format"))
```

Analysis: Account for memory effects

```
mod.format_mem = lmer(response~format*delayed_memory + (1|proid),
                      data = items_df)
anova(mod.format_mem)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##                       Sum Sq Mean Sq NumDF   DenDF F value Pr(>F)
## format                6.7506  2.2502     3 1977.31  1.4579 0.2242
## delayed_memory        2.3953  2.3953     1   33.16  1.5519 0.2216
## format:delayed_memory 4.4469  1.4823     3 1977.58  0.9604 0.4105
```

```
summary(mod.format_mem)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: response ~ format * delayed_memory + (1 | proid)
##    Data: items_df
```

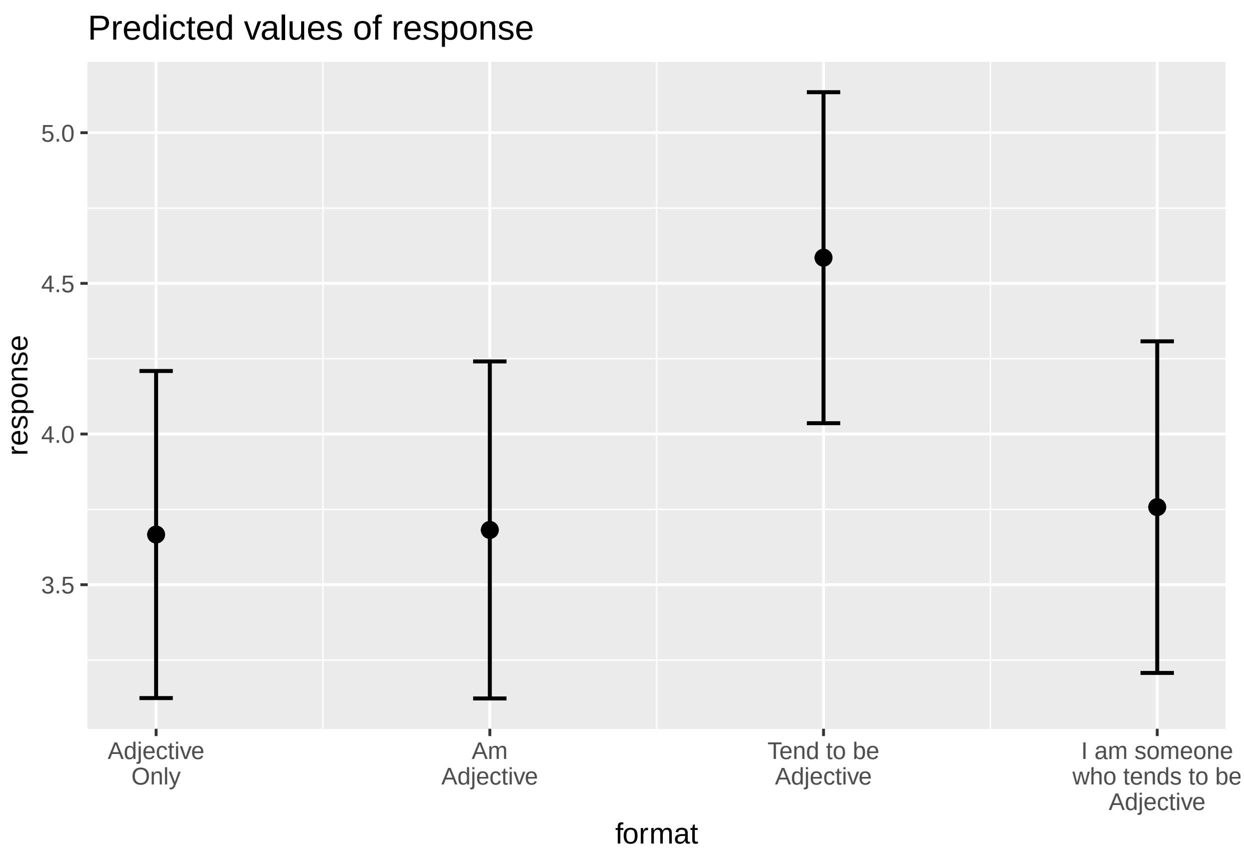


Figure 14: Average response to “thrifty” by format (Block 1 and Block 2)

```
##
## REML criterion at convergence: 6740.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.4138 -0.4310  0.2627  0.6936  1.7952
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  proid    (Intercept) 0.1811   0.4256
##  Residual             1.5435   1.2424
## Number of obs: 2030, groups:  proid, 35
##
## Fixed effects:
##                                                               Estimate
## (Intercept)                                                  4.608e+00
## formatAm\nAdjective                                         -3.353e-01
## formatTend to be\nAdjective                                 -1.070e-01
## formatI am someone\nwho tends to be\nAdjective              -1.945e-01
## delayed_memory                                               6.223e-03
## formatAm\nAdjective:delayed_memory                           4.654e-02
## formatTend to be\nAdjective:delayed_memory                   2.972e-02
## formatI am someone\nwho tends to be\nAdjective:delayed_memory  3.050e-02
##                                                             Std. Error
## (Intercept)                                                  1.851e-01
## formatAm\nAdjective                                          1.654e-01
## formatTend to be\nAdjective                                  1.749e-01
## formatI am someone\nwho tends to be\nAdjective               1.796e-01
## delayed_memory                                               3.164e-02
## formatAm\nAdjective:delayed_memory                           2.765e-02
## formatTend to be\nAdjective:delayed_memory                   3.082e-02
## formatI am someone\nwho tends to be\nAdjective:delayed_memory  3.082e-02
##                                                                     df
## (Intercept)                                                  6.417e+01
## formatAm\nAdjective                                          1.990e+03
## formatTend to be\nAdjective                                  1.946e+03
## formatI am someone\nwho tends to be\nAdjective               1.999e+03
## delayed_memory                                               6.577e+01
## formatAm\nAdjective:delayed_memory                           1.971e+03
## formatTend to be\nAdjective:delayed_memory                   1.927e+03
## formatI am someone\nwho tends to be\nAdjective:delayed_memory  2.006e+03
##                                                             t value Pr(>|t|)
## (Intercept)                                                  24.900   <2e-16
## formatAm\nAdjective                                          -2.028   0.0427
## formatTend to be\nAdjective                                  -0.612   0.5408
## formatI am someone\nwho tends to be\nAdjective               -1.083   0.2787
## delayed_memory                                                0.197   0.8447
## formatAm\nAdjective:delayed_memory                            1.683   0.0925
## formatTend to be\nAdjective:delayed_memory                    0.964   0.3350
## formatI am someone\nwho tends to be\nAdjective:delayed_memory   0.990   0.3224
##
## (Intercept)                                                 ***
## formatAm\nAdjective                                         *
## formatTend to be\nAdjective
```

```
## formatI am someone\nwho tends to be\nAdjective
## delayed_memory
## formatAm\nAdjective:delayed_memory                            .
## formatTend to be\nAdjective:delayed_memory
## formatI am someone\nwho tends to be\nAdjective:delayed_memory
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) frmtAA frTtbA frIaswttbA dlyd_m frAA:_ fTtbA:
## frmtAmAdjct -0.449
## frmtTndtbAd -0.442  0.475
## frmIaswttbA -0.418  0.458  0.471
## delayd_mmry -0.868  0.395  0.389  0.364
## frmtAAdjc:_  0.405 -0.857 -0.428 -0.415     -0.476
## frmtTtbAd:_  0.377 -0.406 -0.865 -0.399     -0.441  0.491
## fIaswttbA:_  0.363 -0.402 -0.410 -0.872     -0.422  0.487  0.462
```

```
plot_model(mod.format_mem,
           type = "pred",
           term = c("format", "delayed_memory[meansd]")) +
  geom_line() +
  labs(x = NULL,
       y = "Average response") +
  scale_color_discrete("Memory", labels = c("-1SD", "Mean", "+1SD"))+
  theme_pubclean()
```

One model for each adjective

```
mod_by_item_mem = items_df %>%
  group_by(item) %>%
  nest() %>%
  mutate(mod = map(data, ~lm(response~format*delayed_memory, data = .))) %>%
  mutate(aov = map(mod, anova))

summary_by_item_mem = mod_by_item_mem %>%
  mutate(tidy = map(aov, broom::tidy)) %>%
  select(item, tidy) %>%
  unnest(cols = c(tidy)) %>%
  filter(term == "format:delayed_memory") %>%
  select(-term, -df) %>%
  mutate(p.adj = p.adjust(p.value, method = "holm"))

summary_by_item_mem %>%
  mutate(across(
    starts_with("p"),
    papaja::printnum
  )) %>%
  kable(digits = 2, booktabs = T) %>%
  kable_styling()
```

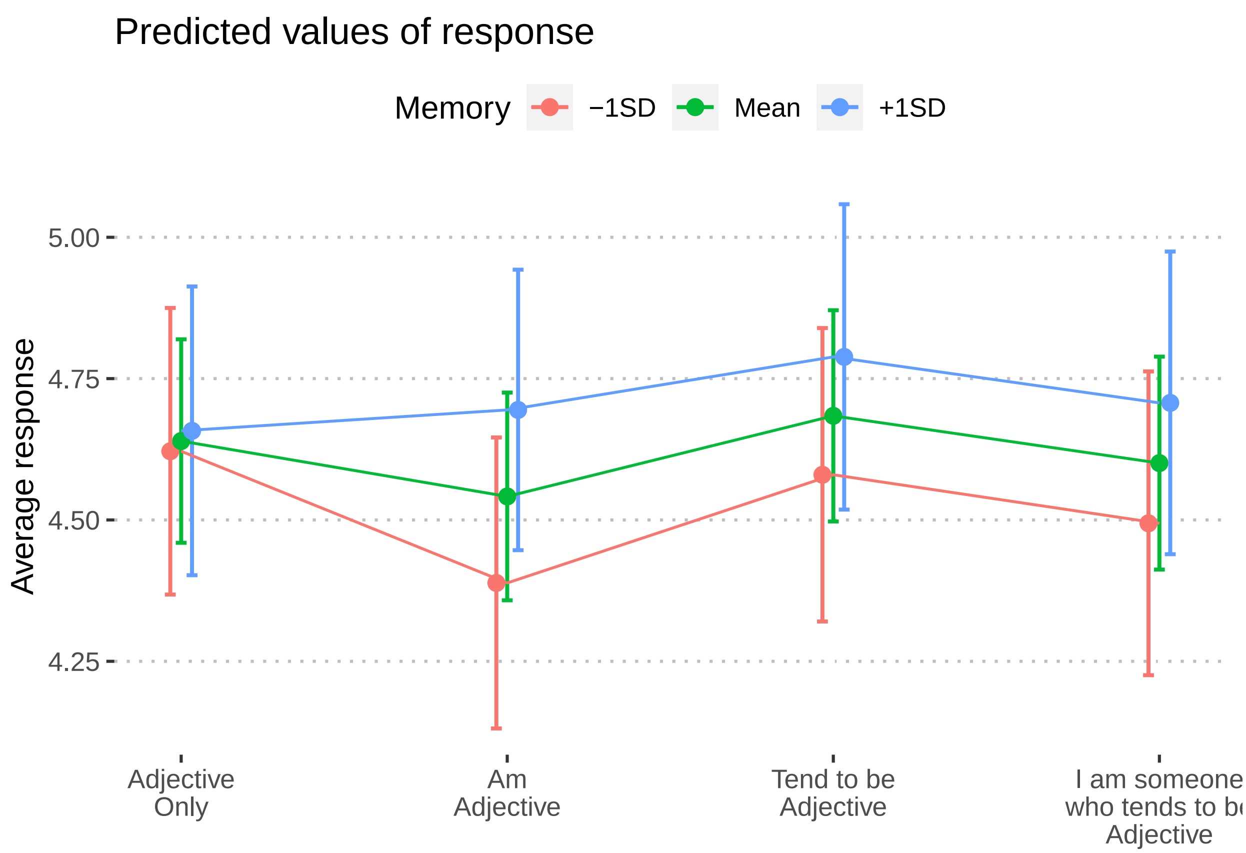


Figure 15: Predicted response on personality items by condition after controlling for delayed_memory.

item	sumsq	meansq	statistic	p.value	p.adj
active	0.29	0.10	0.06	0.98	0.98
adventurous	1.17	0.39	0.39	0.76	0.76
calm	0.15	0.05	0.06	0.98	0.98
careless	3.74	1.25	0.55	0.65	0.65
caring	0.16	0.05	0.08	0.97	0.97
cautious	0.42	0.14	0.12	0.95	0.95
creative	1.12	0.37	0.44	0.73	0.73
curious	1.28	0.43	0.27	0.84	0.84
friendly	1.83	0.61	1.01	0.39	0.39
hardworking	0.74	0.25	0.37	0.77	0.77
helpful	1.31	0.44	0.73	0.54	0.54
imaginative	0.65	0.22	0.35	0.79	0.79
impulsive	7.37	2.46	1.13	0.34	0.34
intelligent	2.94	0.98	0.95	0.42	0.42
lively	1.21	0.40	0.28	0.84	0.84
moody	10.15	3.38	1.76	0.16	0.16
nervous	4.40	1.47	0.57	0.64	0.64
organized	5.06	1.69	1.53	0.22	0.22
outgoing	1.72	0.57	0.40	0.75	0.75
reckless	7.65	2.55	1.01	0.39	0.39
responsible	2.53	0.84	1.11	0.35	0.35
softhearted	0.46	0.15	0.13	0.94	0.94
sophisticated	4.21	1.40	0.89	0.45	0.45
sympathetic	1.11	0.37	0.40	0.75	0.75
talkative	8.53	2.84	1.23	0.31	0.31
thorough	3.87	1.29	1.13	0.34	0.34
thrifty	0.61	0.20	0.13	0.94	0.94
warm	1.65	0.55	0.69	0.56	0.56
worrying	1.77	0.59	0.22	0.88	0.88

Pairwise t-tests for significant ANOVAs

Here we identify the specific items with significant differences.

```
sig_item_mem = summary_by_item_mem %>%
  filter(p.value < .05)

sig_item_mem = sig_item_mem$item
sig_item_mem
```

```
## character(0)
```

How does format affect timining of responses?

Analysis: Block 1 data only

We used a multilevel model, nesting log-seconds within participant to account for dependence. Our primary predictor was format. Here, we use only Block 1 data.

```
item_block1 = filter(items_df, block == "1")

mod.format_b1 = lmer(seconds_log~format + (1|proid),
                     data = item_block1)
anova(mod.format_b1)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##        Sum Sq Mean Sq NumDF DenDF F value  Pr(>F)
## format 6.9057  2.3019     3    31  3.8542 0.01875 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
plot_b1 = plot_model(mod.format_b1, type = "pred")

plot_b1$format +
  labs(x = NULL,
       y = "Average time (log seconds)",
       title = "Average time by item formatting (Block 1 Data)") +
  theme_pubclean()
```

```
plot_b1$format$data %>%
  mutate(predicted = exp(predicted),
         conf.low = exp(conf.low),
         conf.high = exp(conf.high)) %>%
  mutate(x = factor(x,
                    labels = c("Adjective\nOnly",
                               "Am\nAdjective",
                               "Tend to be\nAdjective",
                               "I am someone\nwho tends to be\nAdjective"))) %>%
  ggplot(aes(x = x, y = predicted)) +
  geom_point() +
  geom_errorbar(aes(ymin = conf.low, ymax = conf.high)) +
  labs(x = NULL, y = "seconds", title = "Average time by item formatting (Block 1 Data)") +
  theme_pubclean()
```

```
means_by_group = item_block1 %>%
  group_by(format) %>%
  summarise(m = mean(seconds_log),
            s = sd(seconds_log))

item_block1 %>%
  ggplot(aes(x = seconds_log, fill = format)) +
  geom_histogram(bins = 50, color = "white") +
  geom_vline(aes(xintercept = m), data = means_by_group) +
```

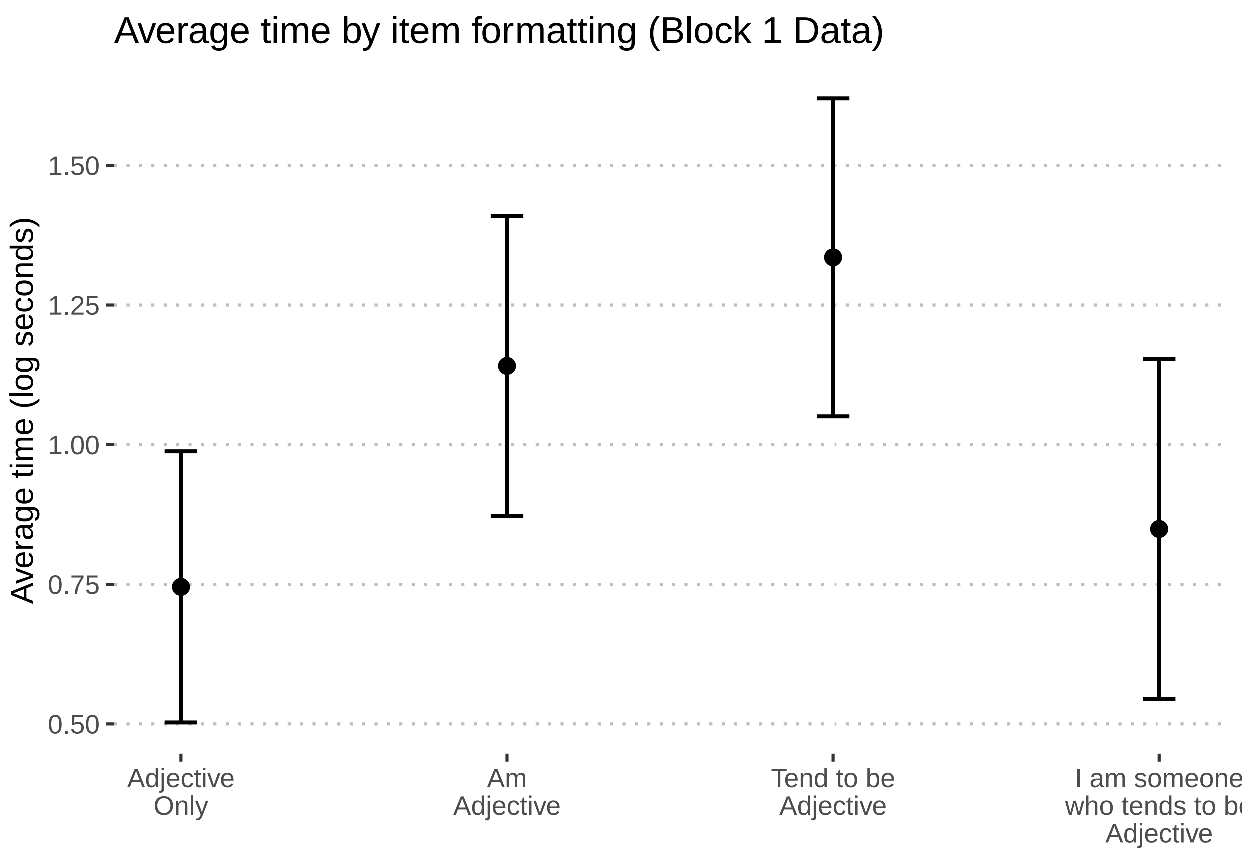


Figure 16: Predicted seconds (log) on personality items by condition, using only Block 1 data.

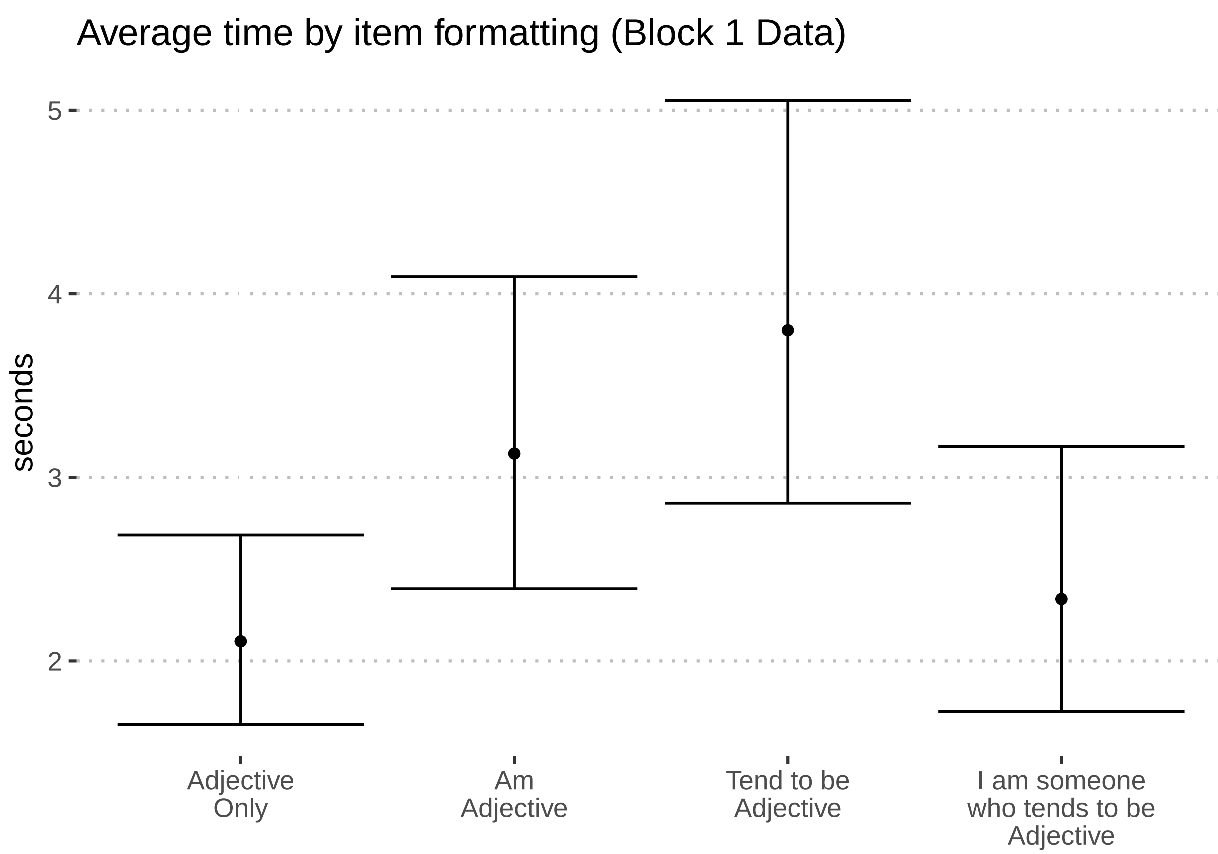


Figure 17: Predicted seconds on personality items by condition, using only Block 1 data.

```
  geom_text(aes(x = 1,
                y = 40,
                label = paste("M =", round(m,2),
                              "\nSD =", round(s,2))),
            data = means_by_group,
            hjust =0,
            vjust = 1) +
  facet_wrap(~format) +
  guides(fill = "none") +
  labs(x = "Log-seconds",
    y = "Number of particpants",
      title = "Distribution of log-seconds by format (Block 1 data)") +
  theme_pubr()
```

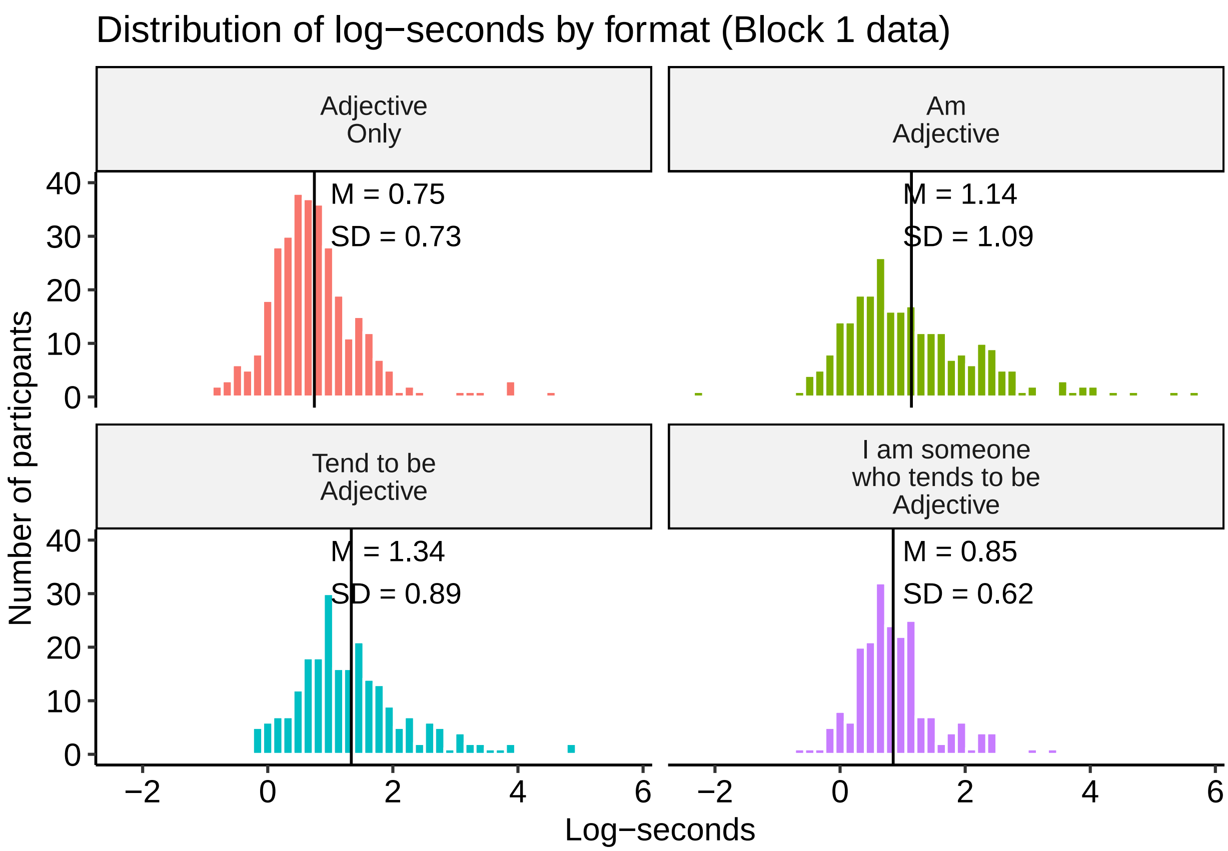


Figure 18: Distribution of time by category, block 1 data only

One model for each adjective

We can also repeat this analysis separately for each trait.

```
mod_by_item_b1 = item_block1 %>%
  group_by(item) %>%
  nest() %>%
  mutate(mod = map(data, ~lm(seconds_log~format, data = .))) %>%
```

```
  mutate(aov = map(mod, anova))

summary_by_item_b1 = mod_by_item_b1 %>%
  mutate(tidy = map(aov, broom::tidy)) %>%
  select(item, tidy) %>%
  unnest(cols = c(tidy)) %>%
  filter(term == "format") %>%
  select(-term, -df) %>%
  mutate(p.adj = p.adjust(p.value, method = "holm"))

summary_by_item_b1 %>%
  mutate(across(
    starts_with("p"),
    papaja::printnum
  )) %>%
  kable(digits = 2, booktabs = T, caption = "Format effects on log-seconds by item (block 1 data only)")
  kable_styling()
```

Pairwise t-tests for significant ANOVAs

Here we identify the specific items with significant differences.

```
sig_item_b1 = summary_by_item_b1 %>%
  filter(p.value < .05)

sig_item_b1 = sig_item_b1$item
sig_item_b1
```

```
## [1] "adventurous" "calm"        "caring"      "helpful"     "softhearted"
## [6] "sympathetic"
```

Then we create models for each adjective. We use the `emmeans` package to perform pairwise comparisons, again with a Holm correction on the *p*-values. We also plot the means and 95% confidence intervals of each mean.

This code will have to be changed after final data collection. It is not self-adapting!

Helpful

```
helpful_model_b1 = item_block1 %>%
  filter(item == "helpful") %>%
  lm(seconds_log~format, data = .)

helpful_em_b1 = emmeans(helpful_model_b1, "format")
pairs(helpful_em_b1, adjust = "holm") %>%
  kable(booktabs = T,
        digits = c(0,2,2,2,2,3),
        caption = "Differences in log-seconds to Helpful by format (Block 1 data only)",
        col.names = c("Contrast", "Difference in means", "SE", "df", "t", "p")) %>%
  kable_styling()
```

Table 7: Format effects on log-seconds by item (block 1 data only)

item	sumsq	meansq	statistic	p.value	p.adj
active	2.02	0.67	0.71	0.56	0.56
adventurous	3.39	1.13	3.21	0.04	0.04
calm	8.79	2.93	3.39	0.03	0.03
careless	0.57	0.19	0.29	0.83	0.83
caring	5.72	1.91	4.45	0.01	0.01
cautious	0.12	0.04	0.07	0.97	0.97
creative	8.30	2.77	2.80	0.06	0.06
curious	2.70	0.90	2.02	0.13	0.13
friendly	1.17	0.39	0.66	0.58	0.58
hardworking	6.54	2.18	2.55	0.07	0.07
helpful	2.17	0.72	3.44	0.03	0.03
imaginative	2.50	0.83	1.31	0.29	0.29
impulsive	4.16	1.39	1.05	0.39	0.39
intelligent	4.15	1.38	1.15	0.34	0.34
lively	3.64	1.21	1.42	0.25	0.25
moody	0.27	0.09	0.25	0.86	0.86
nervous	6.11	2.04	2.54	0.07	0.07
organized	3.42	1.14	1.72	0.18	0.18
outgoing	1.97	0.66	1.02	0.40	0.40
reckless	2.88	0.96	2.09	0.12	0.12
responsible	4.78	1.59	2.79	0.06	0.06
softhearted	8.16	2.72	5.51	0.00	0.00
sophisticated	2.34	0.78	0.71	0.56	0.56
sympathetic	12.92	4.31	6.31	0.00	0.00
talkative	0.20	0.07	0.21	0.89	0.89
thorough	6.75	2.25	2.18	0.11	0.11
thrifty	3.15	1.05	0.97	0.42	0.42
warm	3.70	1.23	2.63	0.07	0.07
worrying	0.85	0.28	0.67	0.58	0.58

Table 8: Differences in log-seconds to Helpful by format (Block 1 data only)

Contrast	Difference in means	SE	df	t	p
Adjective Only - Am Adjective	-0.40	0.21	31	-1.96	0.236
Adjective Only - Tend to be Adjective	-0.46	0.21	31	-2.18	0.184
Adjective Only - I am someone who tends to be Adjective	-0.66	0.22	31	-3.00	0.032
Am Adjective - Tend to be Adjective	-0.06	0.22	31	-0.27	0.812
Am Adjective - I am someone who tends to be Adjective	-0.26	0.23	31	-1.13	0.803
Tend to be Adjective - I am someone who tends to be Adjective	-0.20	0.24	31	-0.84	0.812

```r
plot_model(helpful_model_b1, type = "pred", terms = c("format"))
```

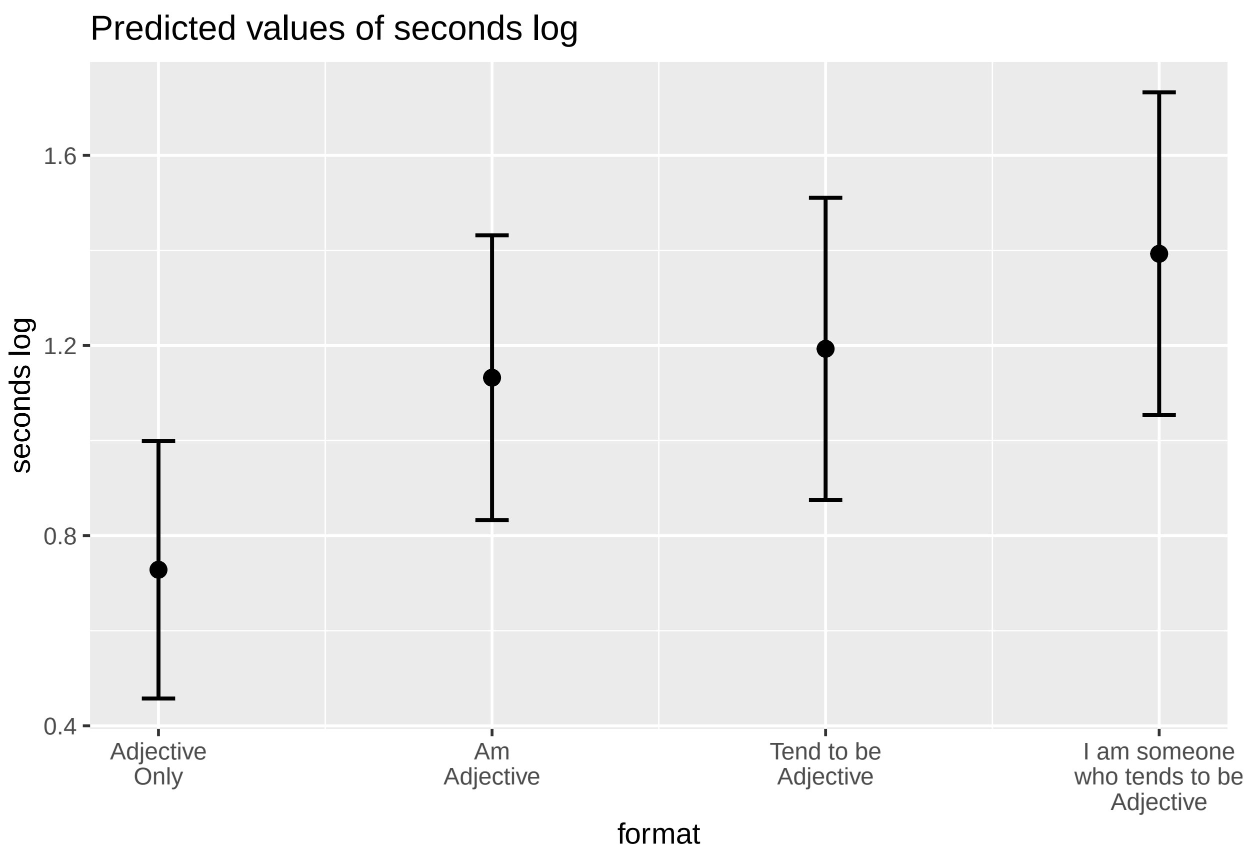


Figure 19: Average log-seconds to "helpful" by format (block 1 data only)

Caring

```r
caring_model_b1 = item_block1 %>%
  filter(item == "caring") %>%
  lm(seconds_log~format, data = .)

caring_em_b1 = emmeans(caring_model_b1, "format")
pairs(caring_em_b1, adjust = "holm") %>%
  kable(booktabs = T,
        digits = c(0,2,2,2,2,3),
        caption = "Differences in log-seconds to Caring by format (Block 1 data only)",
        col.names = c("Contrast", "Difference in means", "SE", "df", "t", "p")) %>%
  kable_styling()
```

```r
plot_model(caring_model_b1, type = "pred", terms = c("format"))
```

Table 9: Differences in log-seconds to Caring by format (Block 1 data only)

Contrast	Difference in means	SE	df	t	p
Adjective Only - Am Adjective	-1.03	0.29	31	-3.51	0.008
Adjective Only - Tend to be Adjective	-0.66	0.30	31	-2.17	0.189
Adjective Only - I am someone who tends to be Adjective	-0.32	0.32	31	-1.02	0.750
Am Adjective - Tend to be Adjective	0.37	0.32	31	1.17	0.750
Am Adjective - I am someone who tends to be Adjective	0.71	0.33	31	2.15	0.189
Tend to be Adjective - I am someone who tends to be Adjective	0.34	0.34	31	0.99	0.750

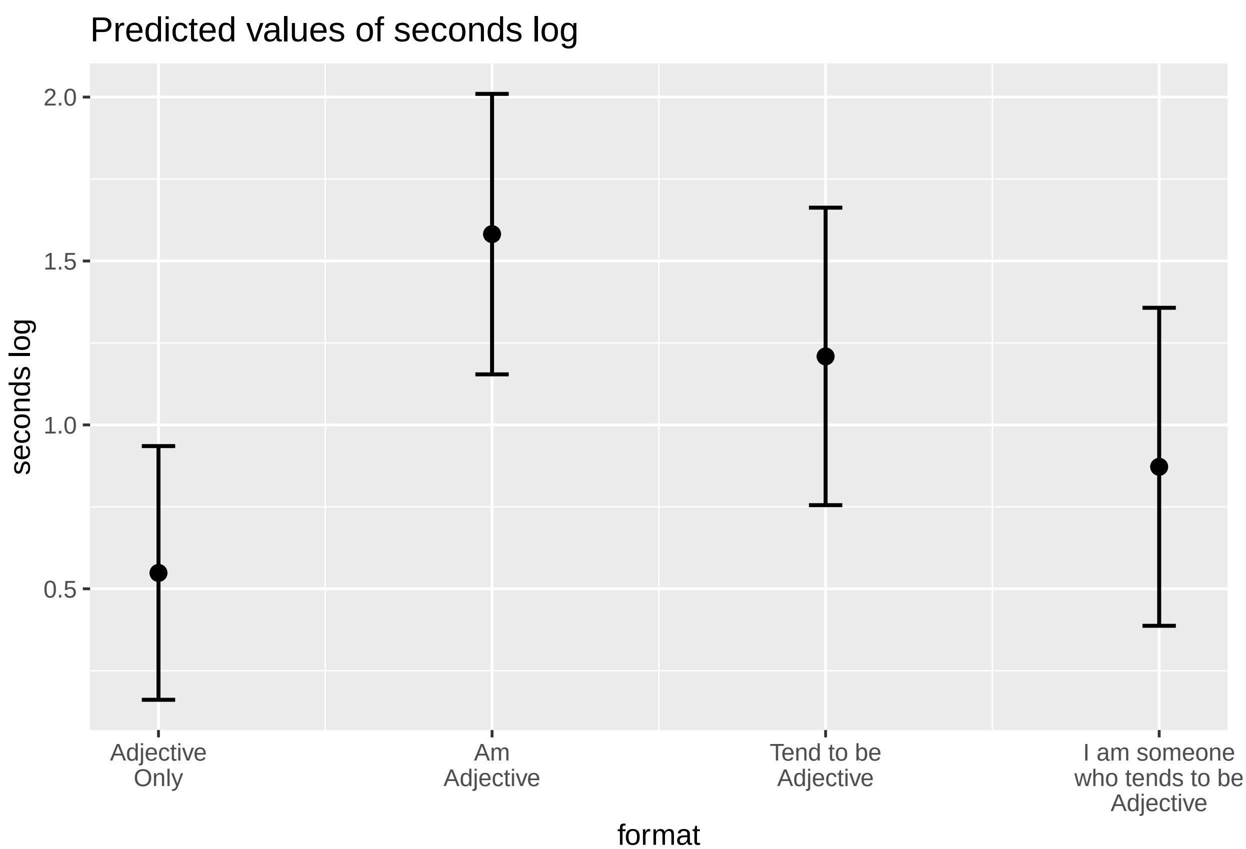


Figure 20: Average log-seconds to “caring” by format (block 1 data only)

Table 10: Differences in log-seconds to Soft-hearted by format (Block 1 data only)

Contrast	Difference in means	SE	df	t	p
Adjective Only - Am Adjective	-1.01	0.32	31	-3.21	0.019
Adjective Only - Tend to be Adjective	-0.96	0.33	31	-2.94	0.031
Adjective Only - I am someone who tends to be Adjective	-0.06	0.34	31	-0.18	1.000
Am Adjective - Tend to be Adjective	0.05	0.34	31	0.15	1.000
Am Adjective - I am someone who tends to be Adjective	0.95	0.35	31	2.69	0.046
Tend to be Adjective - I am someone who tends to be Adjective	0.90	0.36	31	2.47	0.057

Soft-hearted

```
softhearted_model_b1 = item_block1 %>%
  filter(item == "softhearted") %>%
  lm(seconds_log~format, data = .)

softhearted_em_b1 = emmeans(softhearted_model_b1, "format")
pairs(softhearted_em_b1, adjust = "holm") %>%
  kable(booktabs = T,
        digits = c(0,2,2,2,2,3),
        caption = "Differences in log-seconds to Soft-hearted by format (Block 1 data only)",
        col.names = c("Contrast", "Difference in means", "SE", "df", "t", "p")) %>%
  kable_styling()
```

```
plot_model(softhearted_model_b1, type = "pred", terms = c("format"))
```

Calm

```
calm_model_b1 = item_block1 %>%
  filter(item == "calm") %>%
  lm(seconds_log~format, data = .)

calm_em_b1 = emmeans(calm_model_b1, "format")
pairs(calm_em_b1, adjust = "holm") %>%
  kable(booktabs = T,
        digits = c(0,2,2,2,2,3),
        caption = "Differences in log-seconds to Calm by format (Block 1 data only)",
        col.names = c("Contrast", "Difference in means", "SE", "df", "t", "p")) %>%
  kable_styling()
```

```
plot_model(calm_model_b1, type = "pred", terms = c("format"))
```

Sympathetic

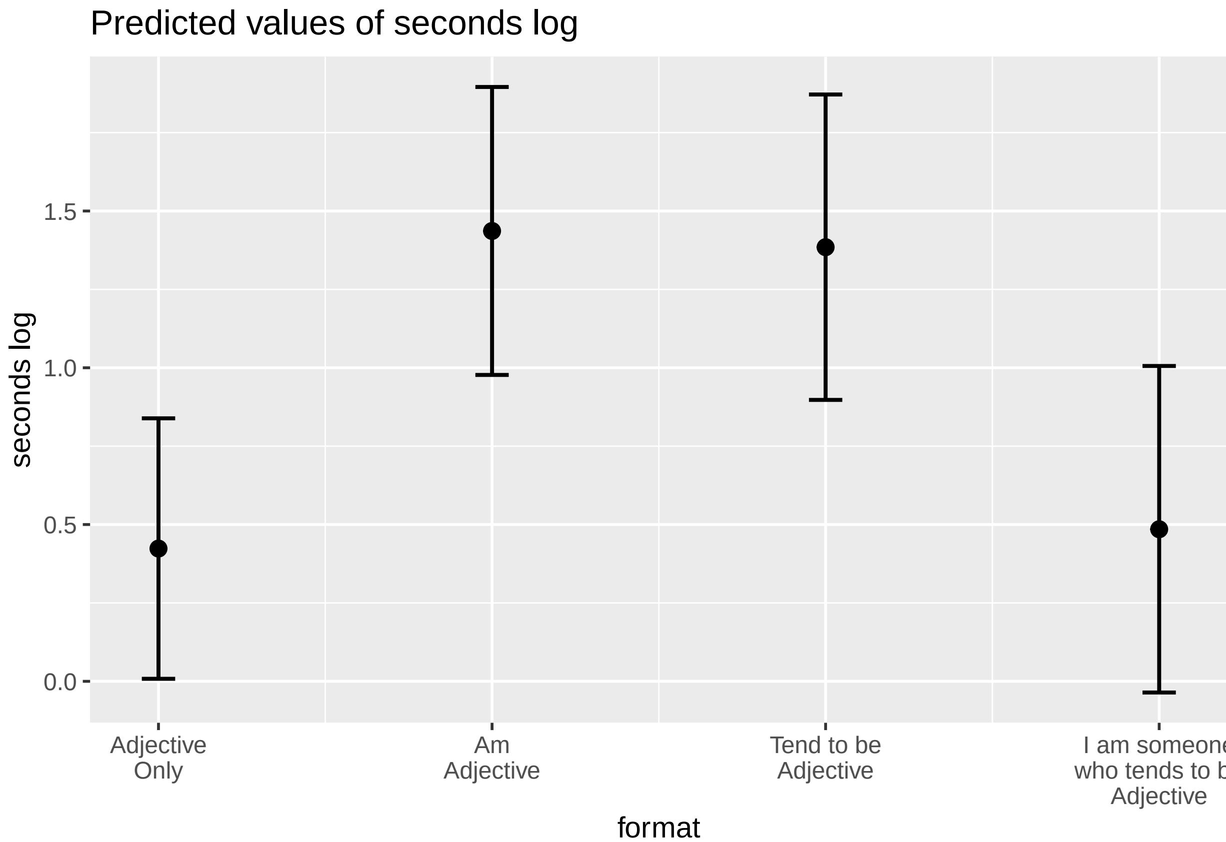


Figure 21: Average log-seconds to “softhearted” by format (block 1 data only)

Table 11: Differences in log-seconds to Calm by format (Block 1 data only)

Contrast	Difference in means	SE	df	t	p
Adjective Only - Am Adjective	-1.07	0.42	31	-2.55	0.095
Adjective Only - Tend to be Adjective	-0.91	0.43	31	-2.10	0.175
Adjective Only - I am someone who tends to be Adjective	0.01	0.45	31	0.02	1.000
Am Adjective - Tend to be Adjective	0.16	0.45	31	0.35	1.000
Am Adjective - I am someone who tends to be Adjective	1.08	0.47	31	2.30	0.142
Tend to be Adjective - I am someone who tends to be Adjective	0.92	0.48	31	1.91	0.197

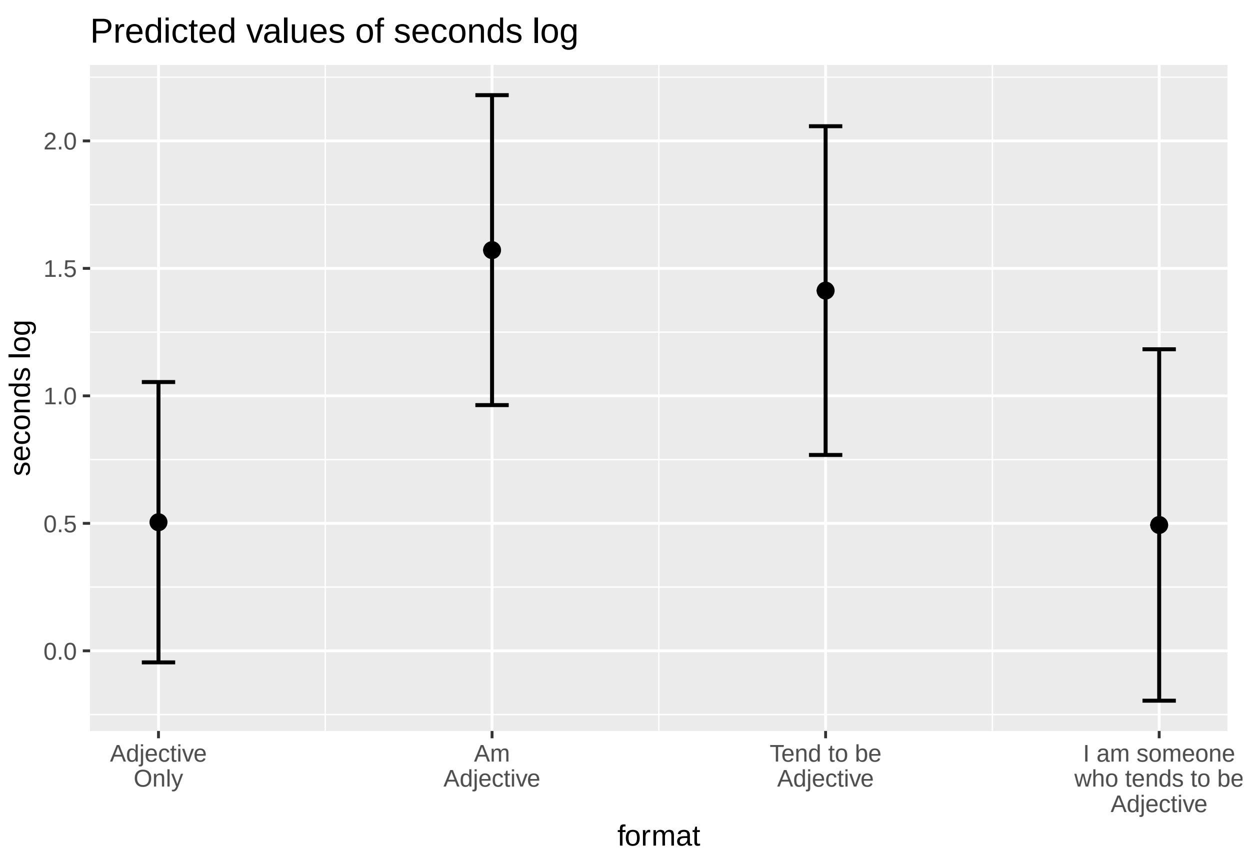


Figure 22: Average log-seconds to “calm” by format (block 1 data only)

Table 12: Differences in log-seconds to Sympathetic by format (Block 1 data only)

Contrast	Difference in means	SE	df	t	p
Adjective Only - Am Adjective	-0.52	0.37	31	-1.39	0.350
Adjective Only - Tend to be Adjective	-1.45	0.38	31	-3.77	0.004
Adjective Only - I am someone who tends to be Adjective	0.18	0.40	31	0.44	0.663
Am Adjective - Tend to be Adjective	-0.93	0.40	31	-2.32	0.109
Am Adjective - I am someone who tends to be Adjective	0.69	0.42	31	1.66	0.320
Tend to be Adjective - I am someone who tends to be Adjective	1.62	0.43	31	3.79	0.004

```
sympathetic_model_b1 = item_block1 %>%
  filter(item == "sympathetic") %>%
  lm(seconds_log~format, data = .)

sympathetic_em_b1 = emmeans(sympathetic_model_b1, "format")
pairs(sympathetic_em_b1, adjust = "holm") %>%
  kable(booktabs = T,
        digits = c(0,2,2,2,2,3),
        caption = "Differences in log-seconds to Sympathetic by format (Block 1 data only)",
        col.names = c("Contrast", "Difference in means", "SE", "df", "t", "p")) %>%
  kable_styling()
```

```
plot_model(sympathetic_model_b1, type = "pred", terms = c("format"))
```

Adventurous

```
adventurous_model_b1 = item_block1 %>%
  filter(item == "adventurous") %>%
  lm(seconds_log~format, data = .)

adventurous_em_b1 = emmeans(adventurous_model_b1, "format")
pairs(adventurous_em_b1, adjust = "holm") %>%
  kable(booktabs = T,
        digits = c(0,2,2,2,2,3),
        caption = "Differences in log-seconds to Adventurous by format (Block 1 data only)",
        col.names = c("Contrast", "Difference in means", "SE", "df", "t", "p")) %>%
  kable_styling()
```

```
plot_model(adventurous_model_b1, type = "pred", terms = c("format"))
```

Analysis: Block 1 and Block 2

We used a multilevel model, nesting log-seconds within participant to account for dependence. Our primary predictor was format. Here, we use only Block 1 data.

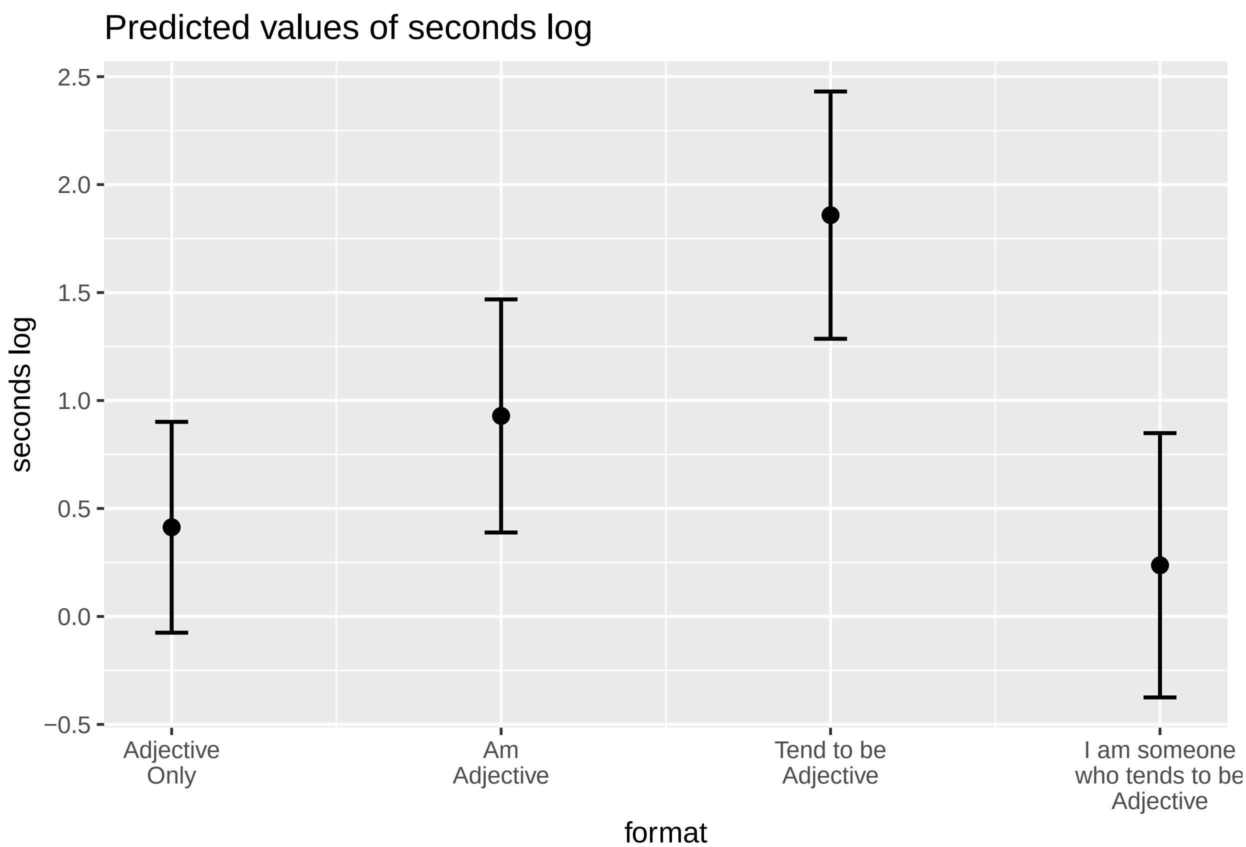


Figure 23: Average log-seconds to “sympathetic” by format (block 1 data only)

Table 13: Differences in log-seconds to Adventurous by format (Block 1 data only)

Contrast	Difference in means	SE	df	t	p
Adjective Only - Am Adjective	-0.12	0.27	31	-0.46	1.000
Adjective Only - Tend to be Adjective	-0.82	0.28	31	-2.96	0.035
Adjective Only - I am someone who tends to be Adjective	-0.27	0.29	31	-0.93	1.000
Am Adjective - Tend to be Adjective	-0.69	0.29	31	-2.41	0.111
Am Adjective - I am someone who tends to be Adjective	-0.14	0.30	31	-0.48	1.000
Tend to be Adjective - I am someone who tends to be Adjective	0.55	0.31	31	1.79	0.332

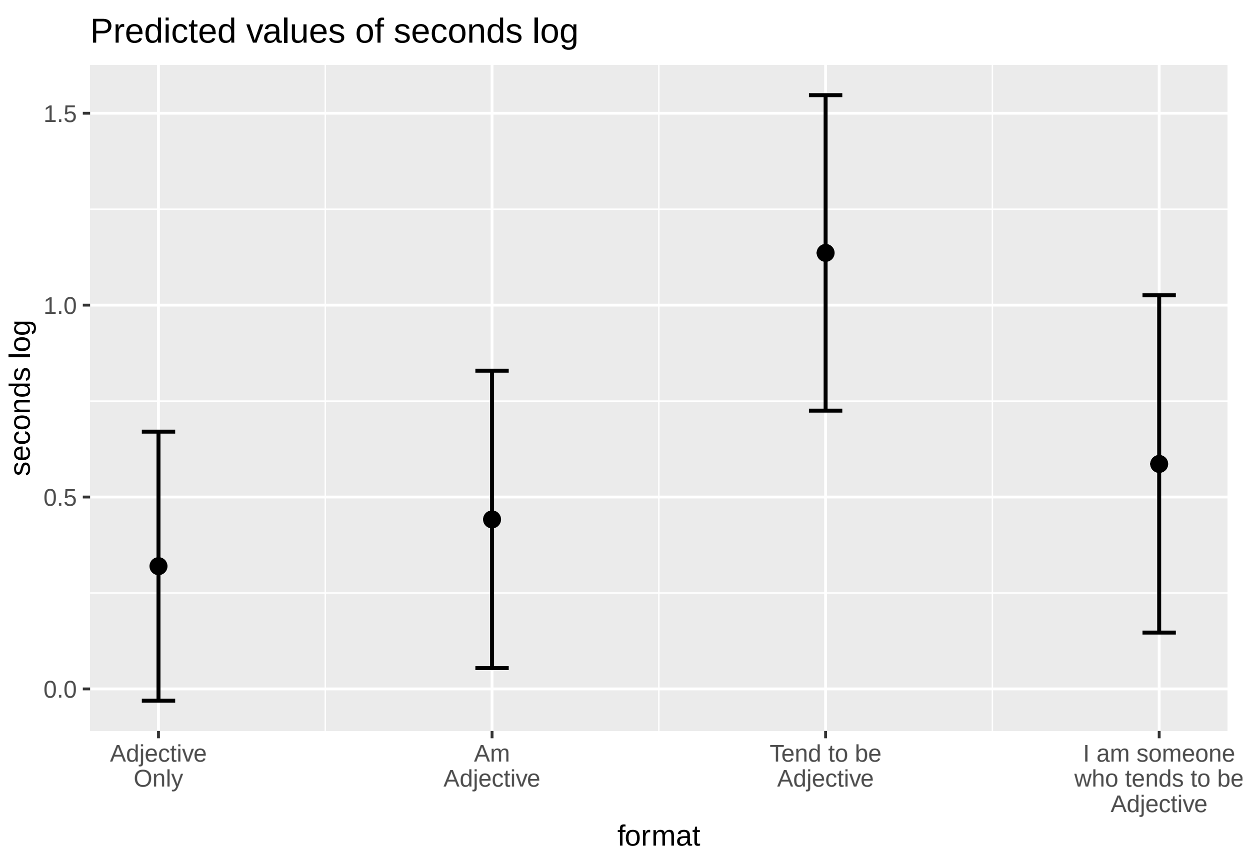


Figure 24: Average log-seconds to “adventurous” by format (block 1 data only)

```
mod.format_b2 = lmer(seconds_log~format + (1|proid),
                    data = items_df)
anova(mod.format_b2)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##        Sum Sq Mean Sq NumDF  DenDF F value    Pr(>F)
## format  23.85  7.9499     3 2017.4   11.82 1.127e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
plot_b2 = plot_model(mod.format_b2, type = "pred")

plot_b2$format +
  labs(x = NULL,
       y = "Average log-seconds",
       title = "Average responses by item formatting (Block 1 and Block 2)") +
  theme_pubclean()
```

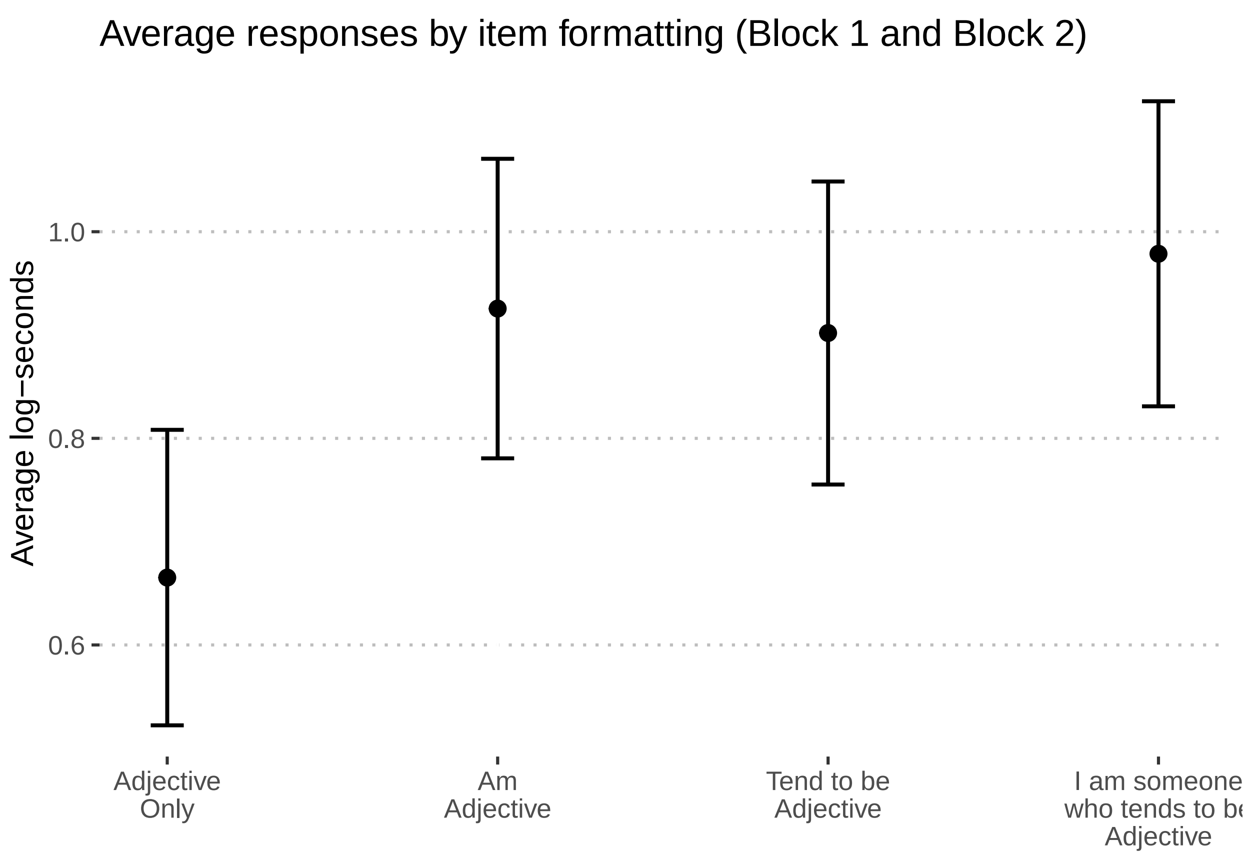


Figure 25: Predicted log-seconds on personality items by condition, using only Block 1 data.

```
plot_b2$format$data %>%
  mutate(predicted = exp(predicted),
         conf.low = exp(conf.low),
```

```
        conf.high = exp(conf.high)) %>%
  mutate(x = factor(x,
                    labels = c("Adjective\nOnly",
                               "Am\nAdjective",
                               "Tend to be\nAdjective",
                               "I am someone\nwho tends to be\nAdjective"))) %>%
  ggplot(aes(x = x, y = predicted)) +
  geom_point() +
  geom_errorbar(aes(ymin = conf.low, ymax = conf.high)) +
  labs(x = NULL, y = "seconds", title = "Average time by item formatting (Block 1 and Block 2)") +
  theme_pubclean()
```

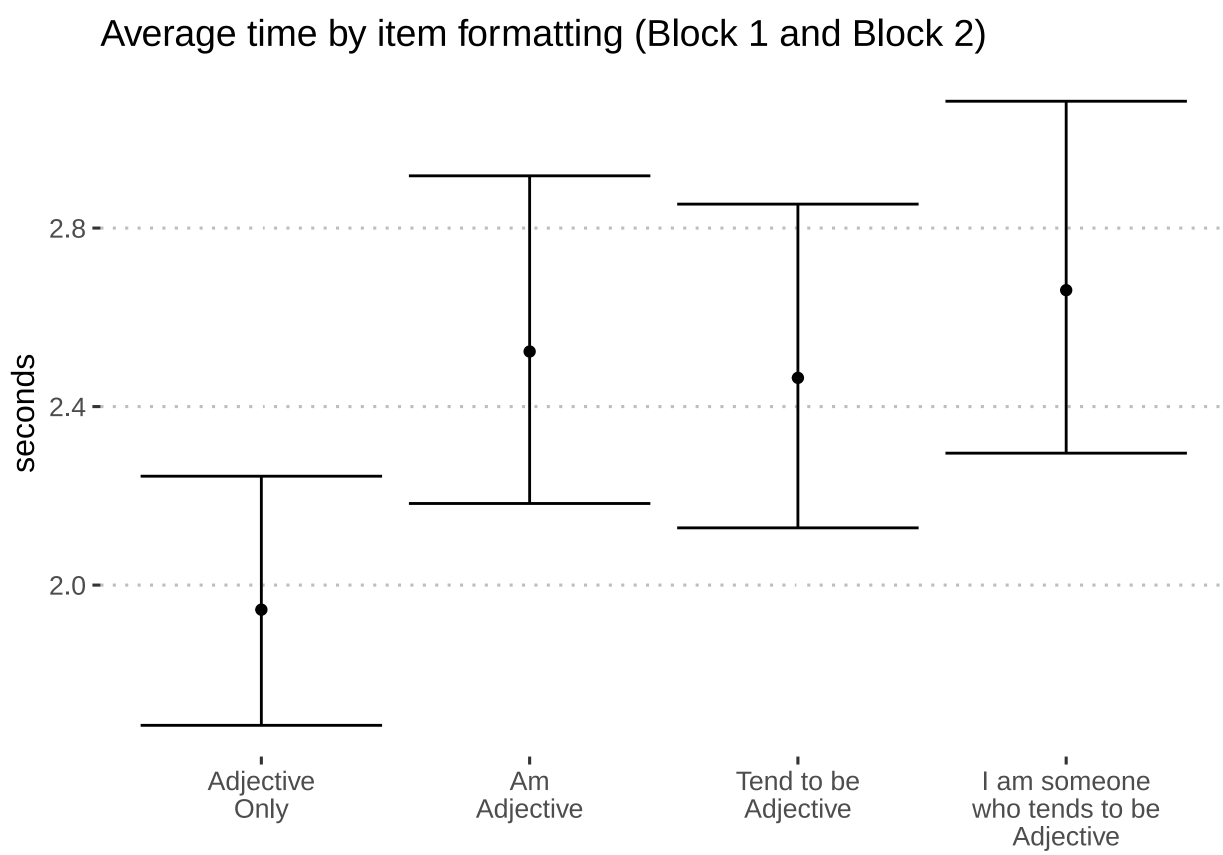


Figure 26: Predicted seconds on personality items by condition, using only Block 1 data.

```
means_by_group = items_df %>%
  group_by(format) %>%
  summarise(m = mean(seconds_log),
            s = sd(seconds_log))

items_df %>%
  ggplot(aes(x = seconds_log, fill = format)) +
  geom_histogram(bins = 100, color = "white") +
  geom_vline(aes(xintercept = m), data = means_by_group) +
  geom_text(aes(x = 1,
                y = 50,
```

```
                label = paste("M =", round(m,2),
                              "\nSD =", round(s,2))),
            data = means_by_group,
            hjust =0,
            vjust = 1) +
  facet_wrap(~format) +
  guides(fill = "none") +
  labs(y = "Number of particpants",
       title = "Distribution of responses by format (Block 1 and Block 2)") +
  theme_pubr()
```

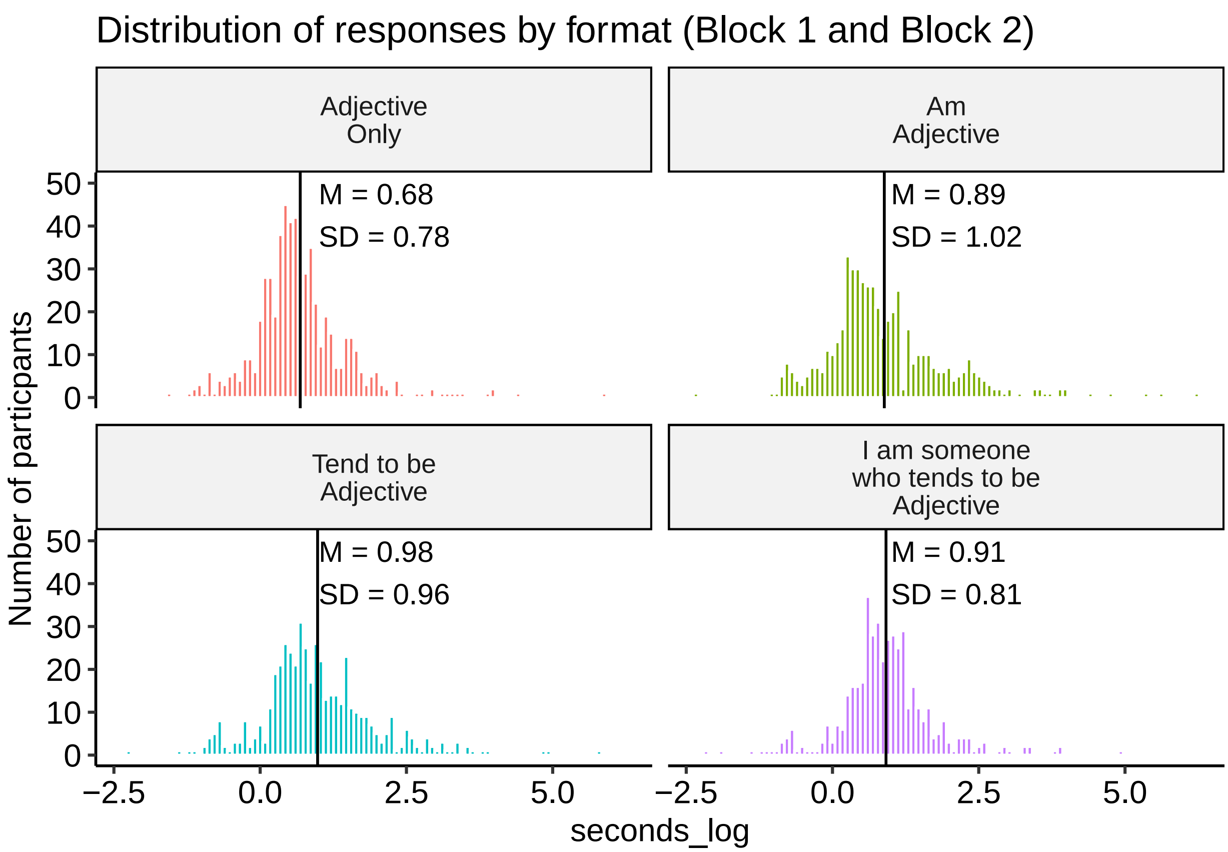


Figure 27: Distribution of log-seconds by category, block 1 and block 2

One model for each adjective

We can also repeat this analysis separately for each trait.

```
mod_by_item_b2 = items_df %>%
  group_by(item) %>%
  nest() %>%
  mutate(mod = map(data, ~lmer(seconds_log~format + (1|proid), data = .))) %>%
  mutate(aov = map(mod, anova))

summary_by_item_b2 = mod_by_item_b2 %>%
```

Table 14: Format effects on log-seconds by item (block 1 data only)

item	sumsq	meansq	NumDF	DenDF	statistic	p.value	p.adj
active	1.82	0.61	3	66.00	0.81	0.49	0.49
adventurous	4.20	1.40	3	59.34	2.99	0.04	0.04
calm	5.06	1.69	3	66.00	2.02	0.12	0.12
careless	2.03	0.68	3	66.00	1.10	0.36	0.36
caring	5.00	1.67	3	66.00	3.01	0.04	0.04
cautious	2.13	0.71	3	51.97	1.14	0.34	0.34
creative	1.97	0.66	3	66.00	0.70	0.55	0.55
curious	2.31	0.77	3	62.52	1.40	0.25	0.25
friendly	0.41	0.14	3	65.49	0.18	0.91	0.91
hardworking	5.53	1.84	3	66.00	2.74	0.05	0.05
helpful	2.04	0.68	3	65.91	2.92	0.04	0.04
imaginative	4.68	1.56	3	66.00	1.98	0.13	0.13
impulsive	2.81	0.94	3	62.76	0.96	0.42	0.42
intelligent	3.20	1.07	3	66.00	1.02	0.39	0.39
lively	2.25	0.75	3	66.00	0.70	0.56	0.56
moody	1.40	0.47	3	65.34	0.96	0.42	0.42
nervous	6.47	2.16	3	66.00	2.42	0.07	0.07
organized	2.16	0.72	3	66.00	1.02	0.39	0.39
outgoing	2.85	0.95	3	60.19	1.46	0.23	0.23
reckless	1.60	0.53	3	63.75	0.53	0.66	0.66
responsible	2.67	0.89	3	66.00	1.57	0.20	0.20
softhearted	2.34	0.78	3	66.00	1.08	0.36	0.36
sophisticated	1.95	0.65	3	66.00	0.67	0.57	0.57
sympathetic	4.19	1.40	3	66.00	1.75	0.17	0.17
talkative	0.22	0.07	3	45.63	0.34	0.79	0.79
thorough	1.54	0.51	3	60.65	0.66	0.58	0.58
thrifty	2.16	0.72	3	55.08	1.48	0.23	0.23
warm	0.81	0.27	3	66.00	0.42	0.74	0.74
worrying	0.76	0.25	3	66.00	0.45	0.72	0.72

```
  mutate(tidy = map(aov, broom::tidy)) %>%
  select(item, tidy) %>%
  unnest(cols = c(tidy)) %>%
  filter(term == "format") %>%
  select(-term) %>%
  mutate(p.adj = p.adjust(p.value, method = "holm"))

summary_by_item_b2 %>%
  mutate(across(
    starts_with("p"),
    papaja::printnum
  )) %>%
  kable(digits = 2, booktabs = T, caption = "Format effects on log-seconds by item (block 1 data only)")
  kable_styling()
```

Table 15: Differences in log-seconds to Helpful by format (Block 1 and Block 2)

Contrast	Difference in means	SE	df	t	p
Adjective Only - Am Adjective	-0.28	0.17	65.88	-1.67	0.414
Adjective Only - Tend to be Adjective	-0.32	0.18	65.98	-1.76	0.414
Adjective Only - I am someone who tends to be Adjective	-0.50	0.18	65.98	-2.80	0.040
Am Adjective - Tend to be Adjective	-0.04	0.18	65.77	-0.19	0.848
Am Adjective - I am someone who tends to be Adjective	-0.21	0.18	65.08	-1.20	0.702
Tend to be Adjective - I am someone who tends to be Adjective	-0.18	0.19	65.85	-0.94	0.702

Pairwise t-tests for significant ANOVAs

Here we identify the specific items with significant differences.

```
sig_item_b2 = summary_by_item_b2 %>%
  filter(p.value < .05)

sig_item_b2 = sig_item_b2$item
sig_item_b2
```

```
## [1] "adventurous" "caring"      "helpful"
```

Then we create models for each adjective. We use the `emmeans` package to perform pairwise comparisons, again with a Holm correction on the *p*-values. We also plot the means and 95% confidence intervals of each mean.

This code will have to be changed after final data collection. It is not self-adapting!

Helfpul

```
helpful_model_b2 = items_df %>%
  filter(item == "helpful") %>%
  lmer(seconds_log~format + (1|proid),
                               data = .)

helpful_em_b2 = emmeans(helpful_model_b2, "format")
pairs(helpful_em_b2, adjust = "holm") %>%
  kable(booktabs = T,
        digits = c(0,2,2,2,2,3),
        caption = "Differences in log-seconds to Helpful by format (Block 1 and Block 2)",
        col.names = c("Contrast", "Difference in means", "SE", "df", "t", "p")) %>%
  kable_styling()
```

```
plot_model(helpful_model_b2, type = "pred", terms = c("format"))
```

Caring

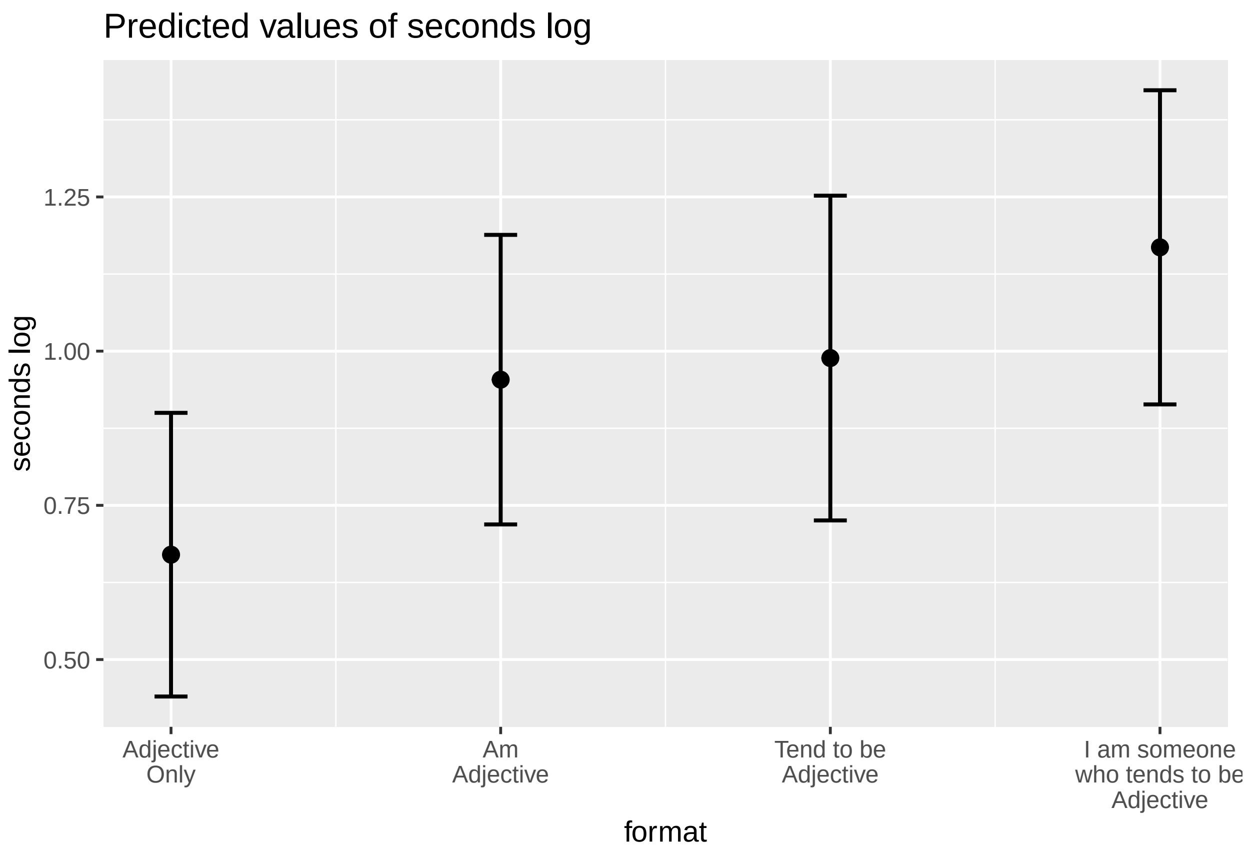


Figure 28: Average log-seconds to “helpful” by format (Block 1 and Block 2)

Table 16: Differences in log-seconds to Caring by format (Block 1 and Block 2)

Contrast	Difference in means	SE	df	t	p
Adjective Only - Am Adjective	-0.65	0.25	65.21	-2.56	0.076
Adjective Only - Tend to be Adjective	-0.60	0.26	63.58	-2.34	0.113
Adjective Only - I am someone who tends to be Adjective	-0.29	0.25	65.99	-1.15	0.757
Am Adjective - Tend to be Adjective	0.05	0.27	65.96	0.18	0.860
Am Adjective - I am someone who tends to be Adjective	0.36	0.26	65.29	1.35	0.723
Tend to be Adjective - I am someone who tends to be Adjective	0.31	0.27	64.65	1.16	0.757

```
caring_model_b2 = items_df %>%
  filter(item == "caring") %>%
  lmer(seconds_log~format + (1|proid),
                              data = .)

caring_em_b2 = emmeans(caring_model_b2, "format")
pairs(caring_em_b2, adjust = "holm") %>%
  kable(booktabs = T,
        digits = c(0,2,2,2,2,3),
        caption = "Differences in log-seconds to Caring by format (Block 1 and Block 2)",
        col.names = c("Contrast", "Difference in means", "SE", "df", "t", "p")) %>%
  kable_styling()
```

```
plot_model(caring_model_b2, type = "pred", terms = c("format"))
```

Adventurous

```
adventurous_model_b2 = items_df %>%
  filter(item == "adventurous") %>%
  lmer(seconds_log~format + (1|proid),
                              data = .)

adventurous_em_b2 = emmeans(adventurous_model_b2, "format")
pairs(adventurous_em_b2, adjust = "holm") %>%
  kable(booktabs = T,
        digits = c(0,2,2,2,2,3),
        caption = "Differences in log-seconds to Adventurous by format (Block 1 and Block 2)",
        col.names = c("Contrast", "Difference in means", "SE", "df", "t", "p")) %>%
  kable_styling()
```

```
plot_model(adventurous_model_b2, type = "pred", terms = c("format"))
```

Analysis: Account for memory effects

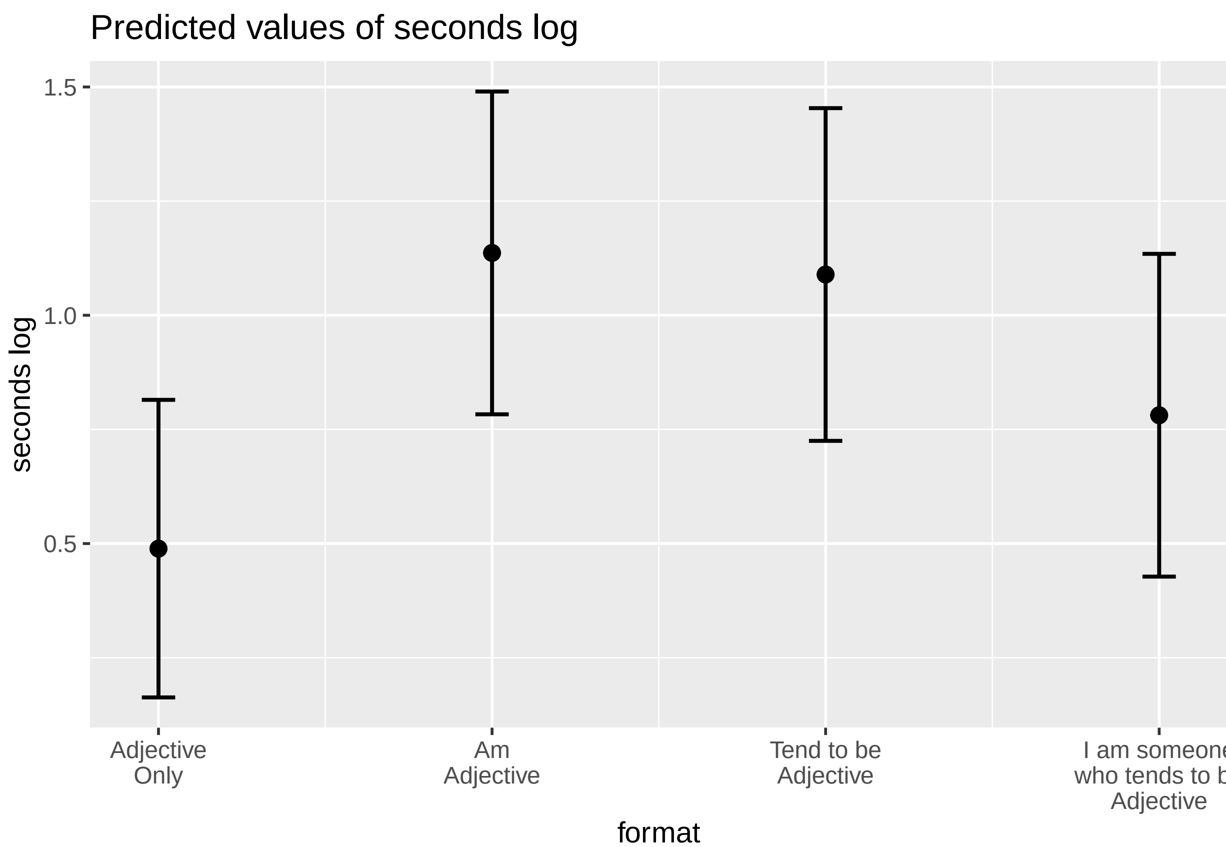


Figure 29: Average log-seconds to “caring” by format (Block 1 and Block 2)

Table 17: Differences in log-seconds to Adventurous by format (Block 1 and Block 2)

Contrast	Difference in means	SE	df	t	p
Adjective Only - Am Adjective	-0.31	0.26	59.39	-1.20	0.724
Adjective Only - Tend to be Adjective	-0.68	0.25	55.91	-2.69	0.057
Adjective Only - I am someone who tends to be Adjective	-0.57	0.27	65.51	-2.13	0.185
Am Adjective - Tend to be Adjective	-0.36	0.27	55.75	-1.35	0.724
Am Adjective - I am someone who tends to be Adjective	-0.26	0.28	62.04	-0.93	0.724
Tend to be Adjective - I am someone who tends to be Adjective	0.10	0.28	63.89	0.37	0.724

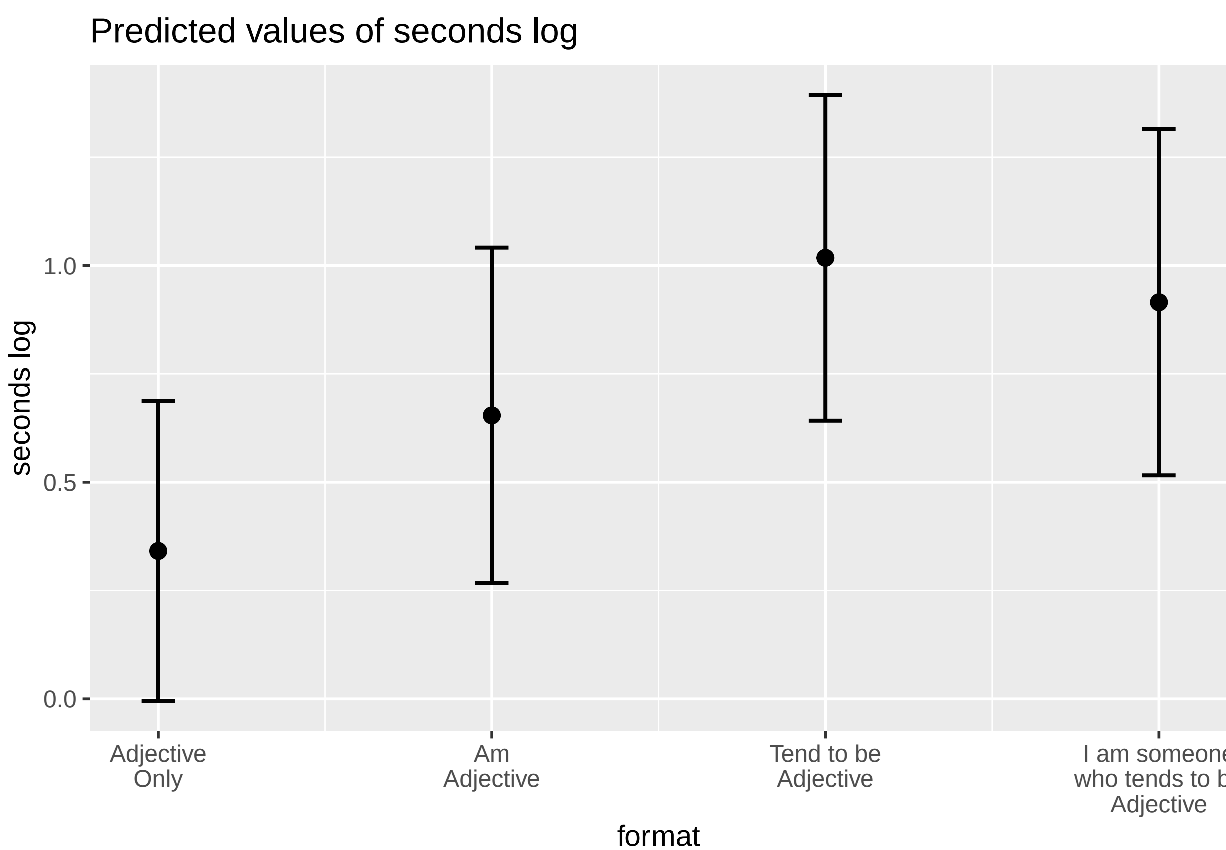


Figure 30: Average log-seconds to "adventurous" by format (Block 1 and Block 2)

```r
mod.format_mem = lmer(seconds_log~format*delayed_memory + (1|proid),
                  data = items_df)
anova(mod.format_mem)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##                       Sum Sq Mean Sq NumDF   DenDF F value    Pr(>F)
## format               12.2428  4.0809     3 2017.71  6.1286 0.0003801 ***
## delayed_memory        0.0686  0.0686     1   33.07  0.1031 0.7502193
## format:delayed_memory 15.3343  5.1114     3 2018.21  7.6762 4.229e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
summary(mod.format_mem)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: seconds_log ~ format * delayed_memory + (1 | proid)
##    Data: items_df
##
## REML criterion at convergence: 5057.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.9381 -0.5402 -0.1418  0.3674  6.8773
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  proid    (Intercept) 0.1397   0.3738
##  Residual             0.6659   0.8160
## Number of obs: 2030, groups:  proid, 35
##
## Fixed effects:
##                                                               Estimate
## (Intercept)                                                    0.72370
## formatAm\nAdjective                                           -0.10681
## formatTend to be\nAdjective                                    0.13727
## formatI am someone\nwho tends to be\nAdjective                 0.38377
## delayed_memory                                                -0.01248
## formatAm\nAdjective:delayed_memory                             0.07052
## formatTend to be\nAdjective:delayed_memory                     0.02071
## formatI am someone\nwho tends to be\nAdjective:delayed_memory -0.01240
##                                                             Std. Error
## (Intercept)                                                    0.14824
## formatAm\nAdjective                                            0.10934
## formatTend to be\nAdjective                                    0.11583
## formatI am someone\nwho tends to be\nAdjective                 0.11866
## delayed_memory                                                 0.02529
## formatAm\nAdjective:delayed_memory                             0.01830
## formatTend to be\nAdjective:delayed_memory                     0.02043
## formatI am someone\nwho tends to be\nAdjective:delayed_memory  0.02036
##                                                                     df
## (Intercept)                                                   51.00813
```

```
## formatAm\nAdjective                                            2020.78149
## formatTend to be\nAdjective                                    2012.77870
## formatI am someone\nwho tends to be\nAdjective                 2021.63962
## delayed_memory                                                   51.97799
## formatAm\nAdjective:delayed_memory                             2017.90928
## formatTend to be\nAdjective:delayed_memory                     2008.14522
## formatI am someone\nwho tends to be\nAdjective:delayed_memory  2021.99447
##                                                                t value Pr(>|t|)
## (Intercept)                                                      4.882 1.08e-05
## formatAm\nAdjective                                             -0.977  0.32877
## formatTend to be\nAdjective                                      1.185  0.23612
## formatI am someone\nwho tends to be\nAdjective                   3.234  0.00124
## delayed_memory                                                  -0.494  0.62371
## formatAm\nAdjective:delayed_memory                               3.854  0.00012
## formatTend to be\nAdjective:delayed_memory                       1.014  0.31077
## formatI am someone\nwho tends to be\nAdjective:delayed_memory   -0.609  0.54247
##
## (Intercept)                                                    ***
## formatAm\nAdjective
## formatTend to be\nAdjective
## formatI am someone\nwho tends to be\nAdjective                 **
## delayed_memory
## formatAm\nAdjective:delayed_memory                             ***
## formatTend to be\nAdjective:delayed_memory
## formatI am someone\nwho tends to be\nAdjective:delayed_memory
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) frmtAA frTtbA frIaswttbA dlyd_m frAA:_ fTtbA:
## frmtAmAdjct -0.370
## frmtTndtbAd -0.366  0.474
## frmIaswttbA -0.346  0.457  0.473
## delayd_mmry -0.868  0.327  0.322  0.302
## frmtAAdjc:_  0.334 -0.857 -0.427 -0.414     -0.394
## frmtTtbAd:_  0.312 -0.405 -0.865 -0.400     -0.365  0.491
## fIaswttbA:_  0.300 -0.402 -0.412 -0.871     -0.350  0.488  0.464
```

```
plot_mem = plot_model(mod.format_mem,
           type = "pred",
           term = c("format", "delayed_memory[meansd]")) +
  geom_line() +
  labs(x = NULL,
       y = "Average log-seconds") +
  scale_color_discrete("Memory", labels = c("-1SD", "Mean", "+1SD"))+
  theme_pubclean()

plot_mem
```

```
plot_mem$data %>% as_tibble %>%
  mutate(predicted = exp(predicted),
         conf.low = exp(conf.low),
         conf.high = exp(conf.high),
```

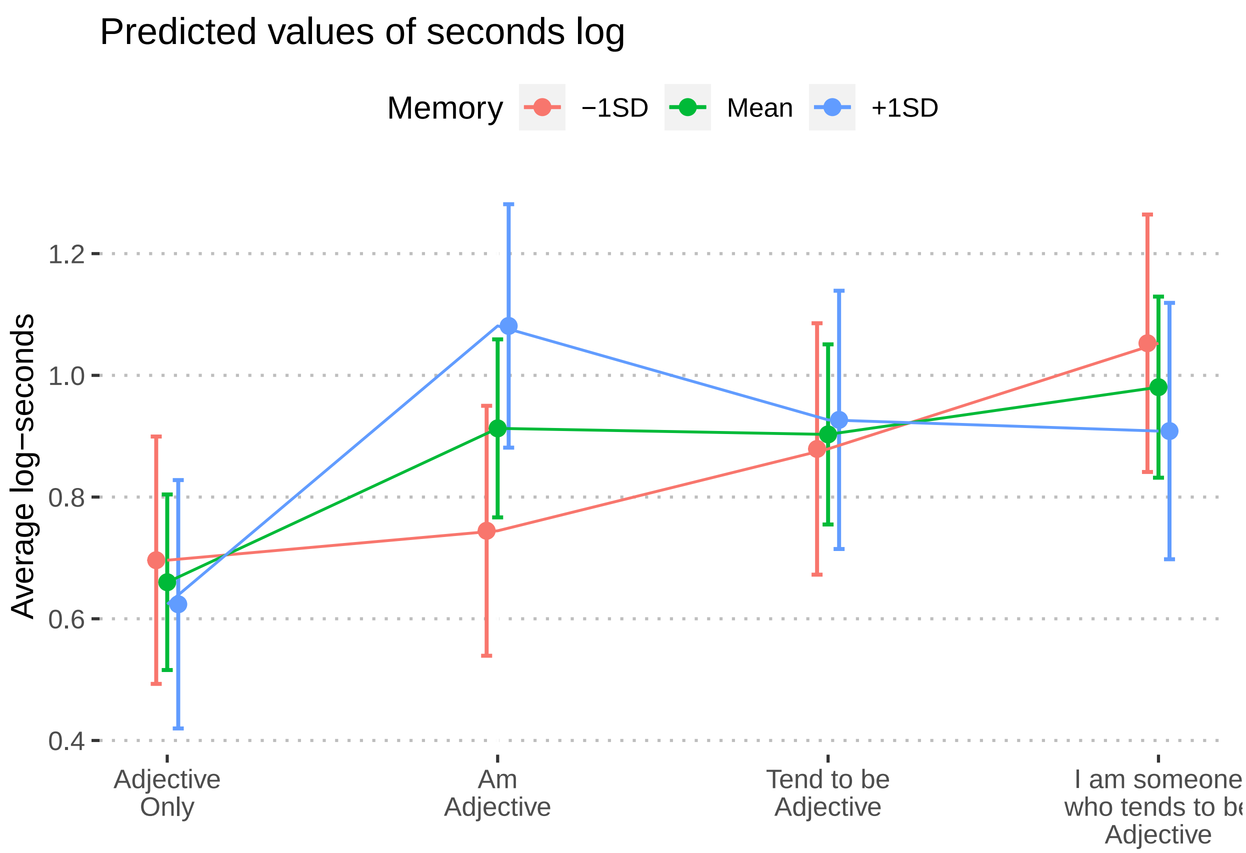


Figure 31: Predicted log-seconds on personality items by condition after controlling for delayed_memory.

```
        group_col = factor(group_col, labels = c("Memory\n-1SD", "Memory\nMean", "Memory\n+1SD"))) %>%
  mutate(x = factor(x,
                    labels = c("Adjective\nOnly",
                               "Am\nAdjective",
                               "Tend to be\nAdjective",
                               "I am someone\nwho tends to be\nAdjective"))) %>%
  ggplot(aes(x = x, y = predicted, color = group_col)) +
  geom_point() +
  geom_errorbar(aes(ymin = conf.low, ymax = conf.high)) +
  labs(x = NULL, y = "seconds", title = "Average time by item formatting (Block 1 and Block 2)") +
  facet_wrap(~group_col) +
  guides(color = "none") +
  theme_pubclean()
```

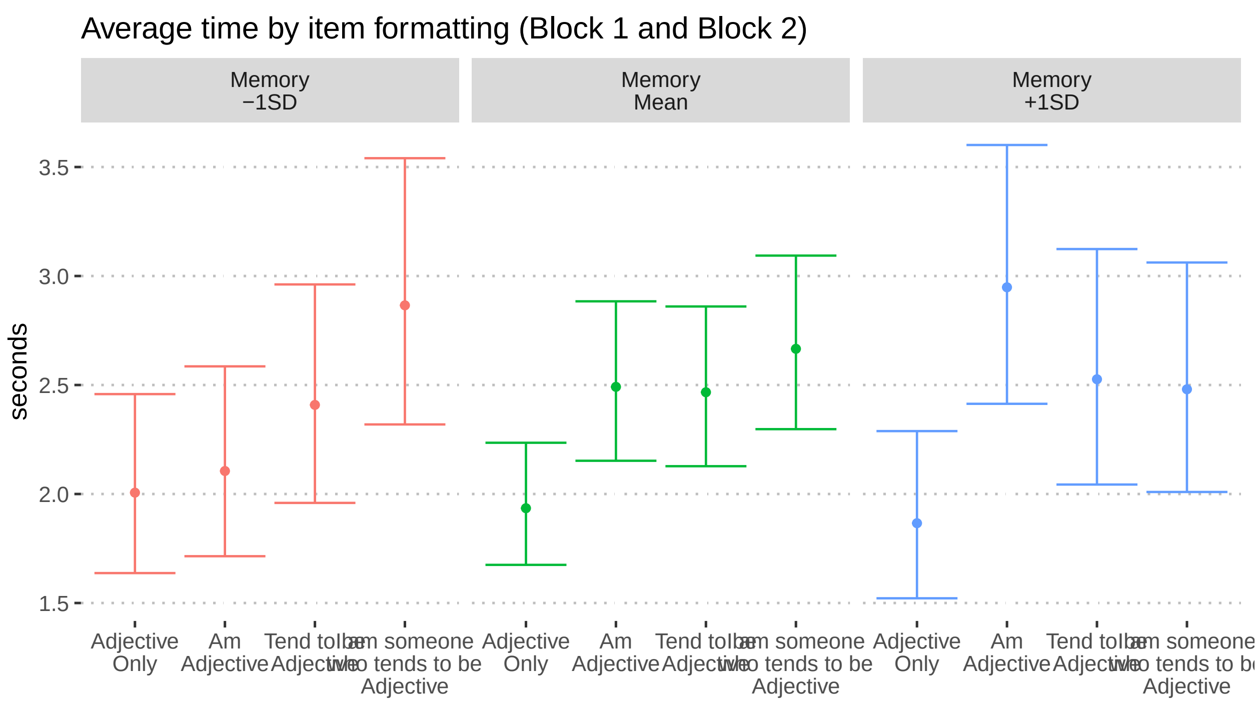


Figure 32: Predicted seconds on personality items by condition after controlling for delayed_memory.

One model for each adjective

```
mod_by_item_mem = items_df %>%
  group_by(item) %>%
  nest() %>%
  mutate(mod = map(data, ~lmer(seconds_log~format*delayed_memory + (1|proid), data = .))) %>%
  mutate(aov = map(mod, anova))

summary_by_item_mem = mod_by_item_mem %>%
  mutate(tidy = map(aov, broom::tidy)) %>%
  select(item, tidy) %>%
```

item	sumsq	meansq	NumDF	DenDF	statistic	p.value	p.adj
active	0.80	0.27	3	62.00	0.35	0.79	0.79
adventurous	1.47	0.49	3	54.52	1.06	0.38	0.38
calm	2.43	0.81	3	62.00	0.96	0.42	0.42
careless	2.81	0.94	3	61.60	1.68	0.18	0.18
caring	1.25	0.42	3	62.00	0.73	0.54	0.54
cautious	1.31	0.44	3	47.18	0.70	0.56	0.56
creative	6.90	2.30	3	62.00	2.61	0.06	0.06
curious	2.25	0.75	3	53.87	1.48	0.23	0.23
friendly	2.29	0.76	3	56.38	1.09	0.36	0.36
hardworking	1.54	0.51	3	62.00	0.74	0.53	0.53
helpful	1.05	0.35	3	56.98	1.43	0.24	0.24
imaginative	6.19	2.06	3	62.00	2.79	0.05	0.05
impulsive	7.30	2.43	3	61.99	2.74	0.05	0.05
intelligent	3.11	1.04	3	62.00	0.97	0.41	0.41
lively	1.36	0.45	3	62.00	0.40	0.75	0.75
moody	0.43	0.14	3	54.42	0.29	0.83	0.83
nervous	2.86	0.95	3	62.00	1.06	0.37	0.37
organized	1.62	0.54	3	62.00	0.75	0.52	0.52
outgoing	6.13	2.04	3	58.90	3.48	0.02	0.02
reckless	6.95	2.32	3	60.74	2.61	0.06	0.06
responsible	7.33	2.44	3	62.00	5.05	0.00	0.00
softhearted	3.03	1.01	3	62.00	1.43	0.24	0.24
sophisticated	8.05	2.68	3	62.00	3.09	0.03	0.03
sympathetic	1.31	0.44	3	62.00	0.53	0.66	0.66
talkative	2.19	0.73	3	39.67	4.15	0.01	0.01
thorough	3.48	1.16	3	55.00	1.48	0.23	0.23
thrifty	1.23	0.41	3	49.59	0.84	0.48	0.48
warm	5.23	1.74	3	56.82	3.30	0.03	0.03
worrying	1.49	0.50	3	62.00	0.87	0.46	0.46

```
  unnest(cols = c(tidy)) %>%
  filter(term == "format:delayed_memory") %>%
  select(-term) %>%
  mutate(p.adj = p.adjust(p.value, method = "holm"))

summary_by_item_mem %>%
  mutate(across(
    starts_with("p"),
    papaja::printnum
  )) %>%
  kable(digits = 2, booktabs = T) %>%
  kable_styling()
```

Pairwise t-tests for significant ANOVAs

Here we identify the specific items with significant differences.

```
sig_item_mem = summary_by_item_mem %>%
  filter(p.value < .05)

sig_item_mem = sig_item_mem$item
sig_item_mem
```

```
## [1] "imaginative"   "outgoing"      "responsible"   "sophisticated"
## [5] "talkative"     "warm"
```

Outgoing

```
outgoing_model_mem = items_df %>%
  filter(item == "outgoing") %>%
  lmer(seconds_log~format*delayed_memory + (1|proid),
                               data = .)
```

```
plot_model(outgoing_model_mem, type = "pred", terms = c("format", "delayed_memory[meansd]"))
```

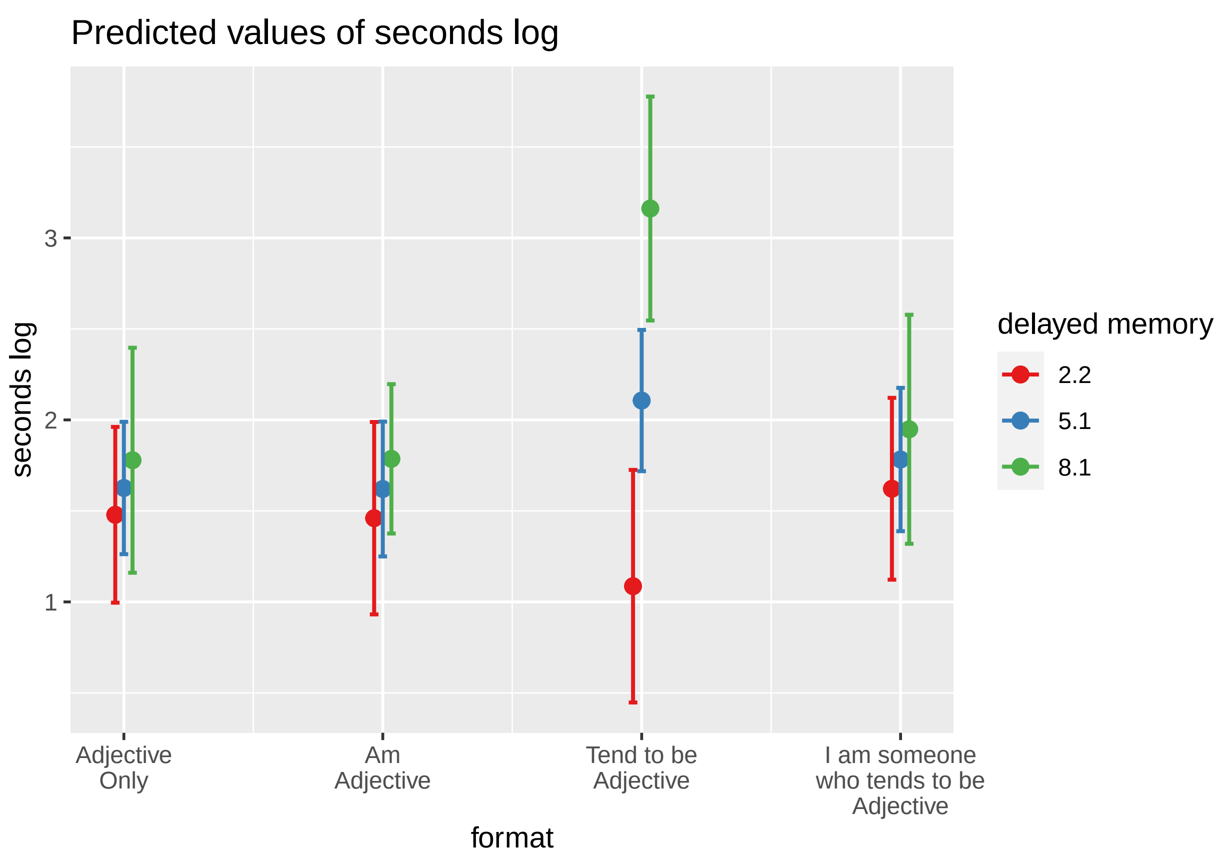


Figure 33: Average log-seconds to "outgoing" by format (Block 1 and Block 2)

Warm

```
warm_model_mem = items_df %>%
  filter(item == "warm") %>%
  lmer(seconds_log~format*delayed_memory + (1|proid),
                                   data = .)
```

```
plot_model(warm_model_mem, type = "pred", terms = c("format", "delayed_memory[meansd]"))
```

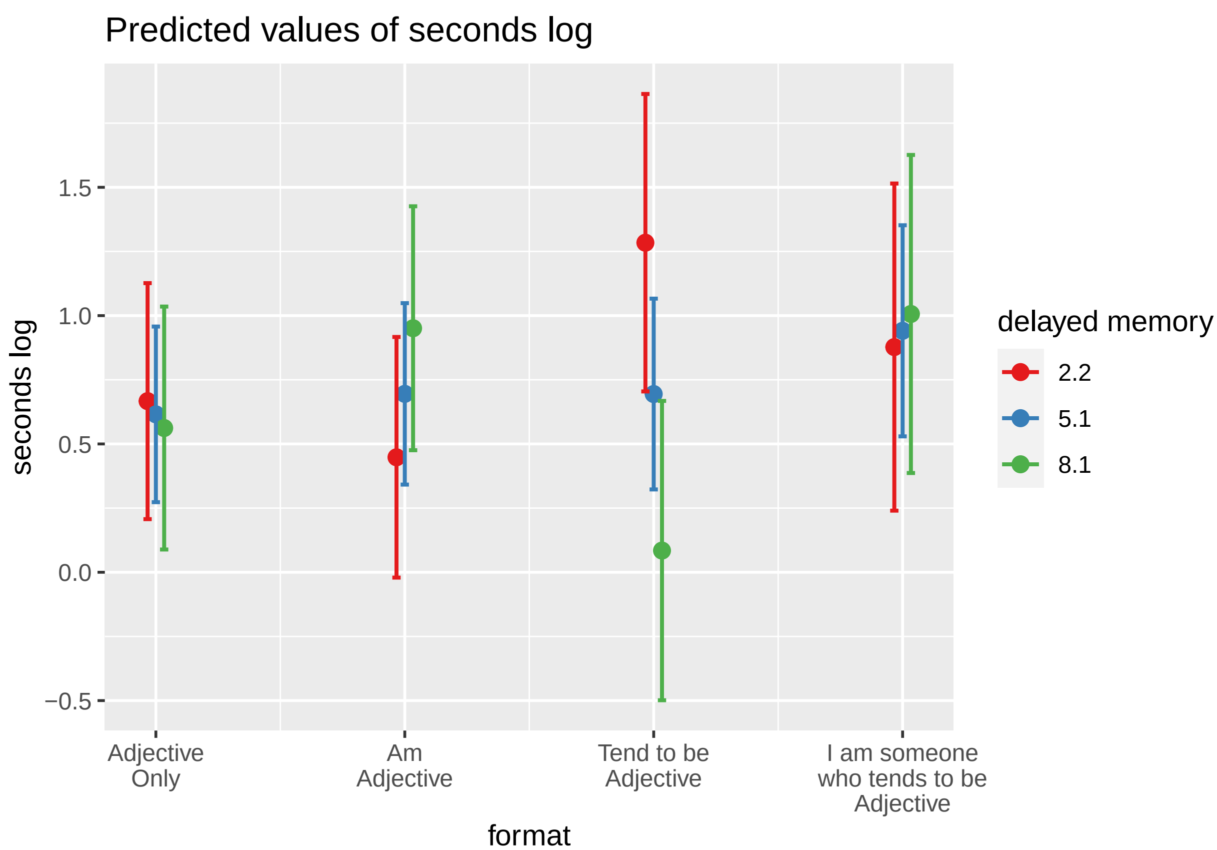


Figure 34: Average log-seconds to "warm" by format (Block 1 and Block 2)

Responsible

```
responsible_model_mem = items_df %>%
  filter(item == "responsible") %>%
  lmer(seconds_log~format*delayed_memory + (1|proid),
                                   data = .)
```

```
plot_model(responsible_model_mem, type = "pred", terms = c("format", "delayed_memory[meansd]"))
```

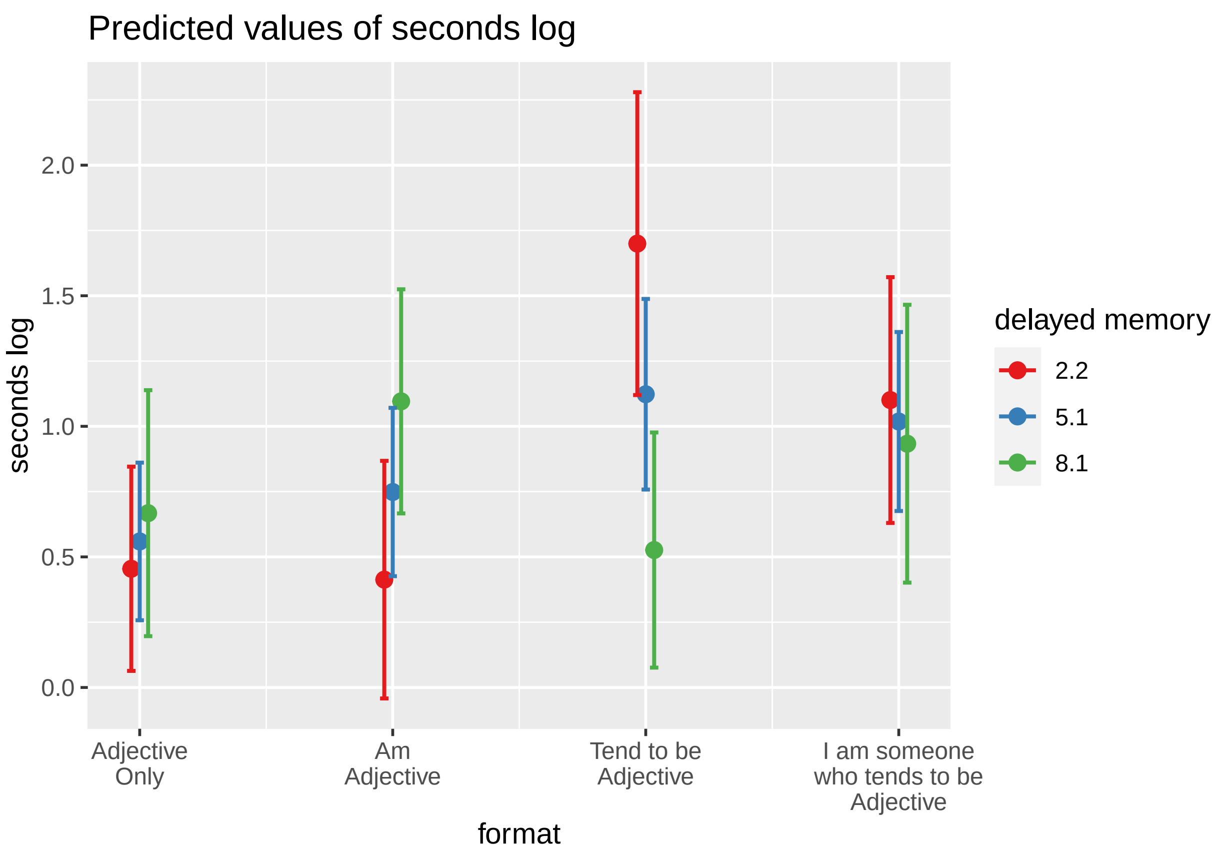


Figure 35: Average log-seconds to "responsible" by format (Block 1 and Block 2)

Imaginative

```
imaginative_model_mem = items_df %>%
  filter(item == "imaginative") %>%
  lmer(seconds_log~format*delayed_memory + (1|proid),
                                  data = .)
```

```
plot_model(imaginative_model_mem, type = "pred", terms = c("format", "delayed_memory[meansd]"))
```

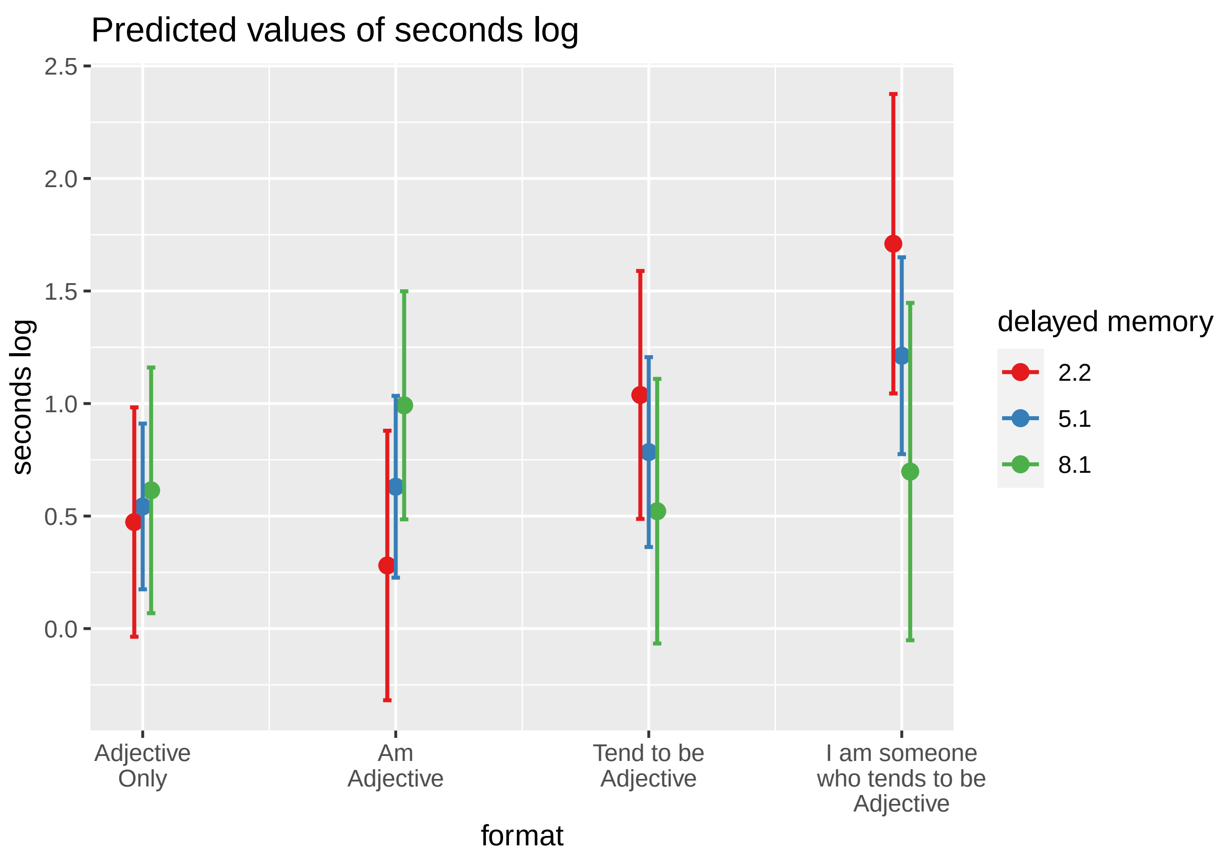


Figure 36: Average log-seconds to "imaginative" by format (Block 1 and Block 2)

Talkative

```
talkative_model_mem = items_df %>%
  filter(item == "talkative") %>%
  lmer(seconds_log~format*delayed_memory + (1|proid),
                                  data = .)
```

```
plot_model(talkative_model_mem, type = "pred", terms = c("format", "delayed_memory[meansd]"))
```

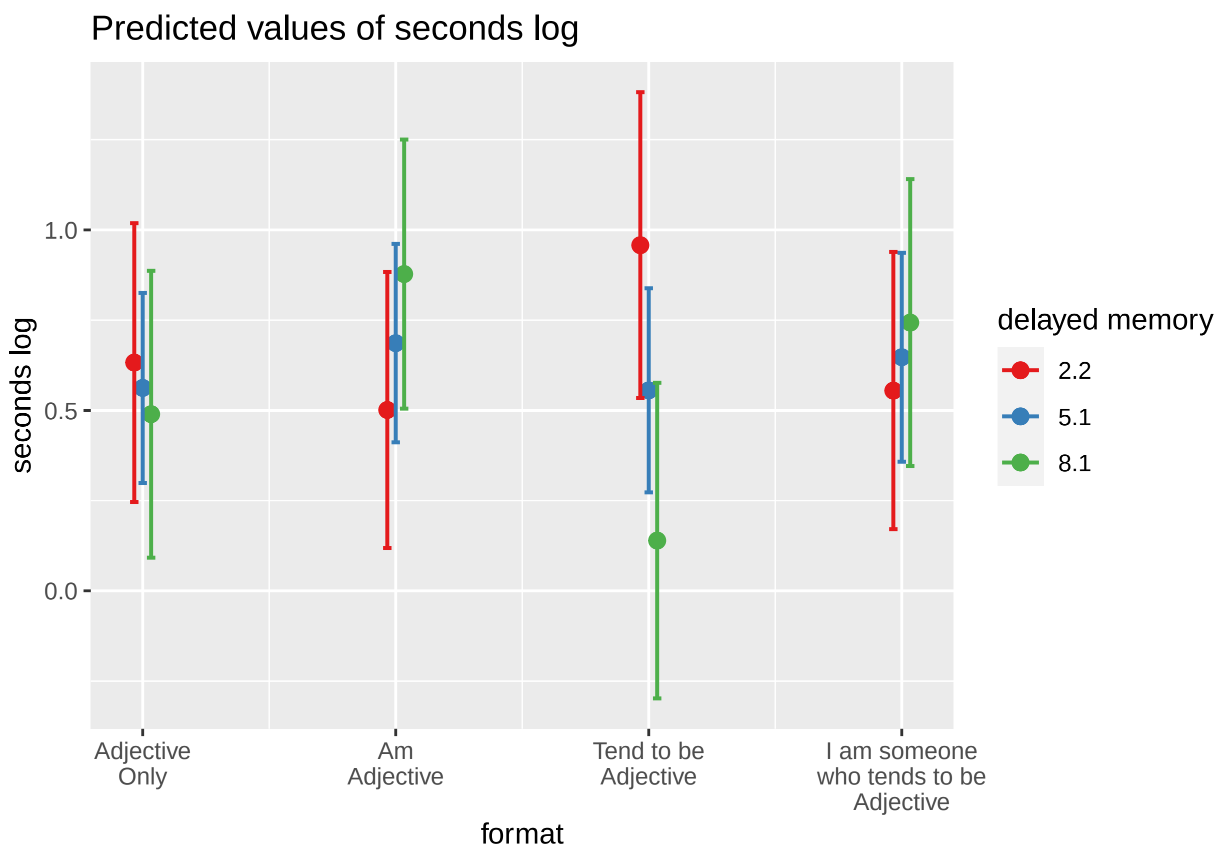


Figure 37: Average log-seconds to “talkative” by format (Block 1 and Block 2)

Sophisticated

```
sophisticated_model_mem = items_df %>%
  filter(item == "sophisticated") %>%
  lmer(seconds_log~format*delayed_memory + (1|proid),
                                data = .)
```

```
plot_model(sophisticated_model_mem, type = "pred", terms = c("format", "delayed_memory[meansd]"))
```

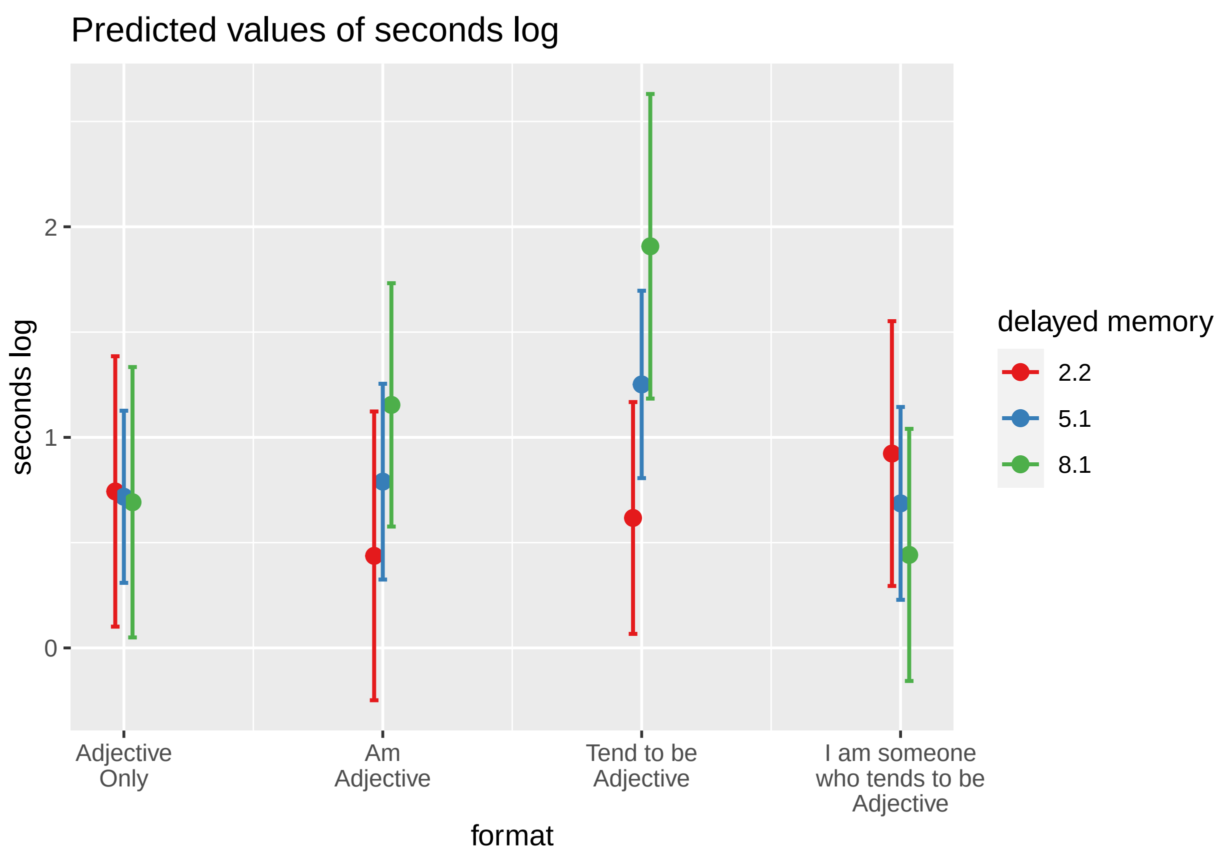


Figure 38: Average log-seconds to "sophisticated" by format (Block 1 and Block 2)

How does device type affect means and timing of responses?

Workspace

```
library(here) # for working with files
library(tidyverse) # for cleaning
library(janitor) # for variable names
library(lme4) # for mulitlevel modeling
library(lmerTest) # for df and p-values
library(emmeans) # for pairwise comparisons
library(sjPlot) # for figures
library(ggpubr) # for prettier plots
library(kableExtra) # for nicer tables
```

Data prep: item responses

First we select the responses to the items of different formats. For this set of analyses, we use data collected in both Block 1 and Block 2 – that is, each participant saw the same format for every item during Block 1, but a random format for each item in Block 2.

These variable names have one of two formats: `[trait]_[abcd]` (for example, `talkative_a`) or `[trait]_[abcd]_2` (for example, `talkative_a_2`). We search for these items using regular expressions.

```
items_seen_b1b2 = str_subset(
  names(data),
  "^([[:alpha:]])+_[abcd](_2)?$"
)

item_responses = data %>%
  select(proid, all_of(items_seen_b1b2), devicetype)
```

Next we reshape these data into long form.

```
item_responses = item_responses %>%
  gather(item, response, -proid, -devicetype) %>%
  mutate(
    block = case_when(              # which block is the item from?
      str_detect(item, "_2") ~ "2",
      TRUE ~ "1"),
    item = str_remove(item, "_2")) %>%  # remove block id from item string
  separate(item, into = c("item", "format")) %>%
  filter(!is.na(response))
```

We give labels to the formats, to clarify interpretations and aid table and figure construction.

```
item_responses$format = as.factor(item_responses$format)
item_responses$format = relevel(item_responses$format, ref = "a")
item_responses$format = factor(item_responses$format,
                               levels = c("a","b","c","d"),
                               labels = c("Adjective\nOnly", "Am\nAdjective", "Tend to be\nAdjective",
```

Data prep: item timing

We also prepare a dataframe to examine the amount of time needed to respond to items. These variable names have one of two formats: `t_[trait]_[abcd]_page_submit` (for example, `t_talkative_a_page_submit`) or `t_[trait]_[abcd]_2_page_submit` (for example, `t_talkative_a_2_page_submit`). We search for these items using regular expressions.

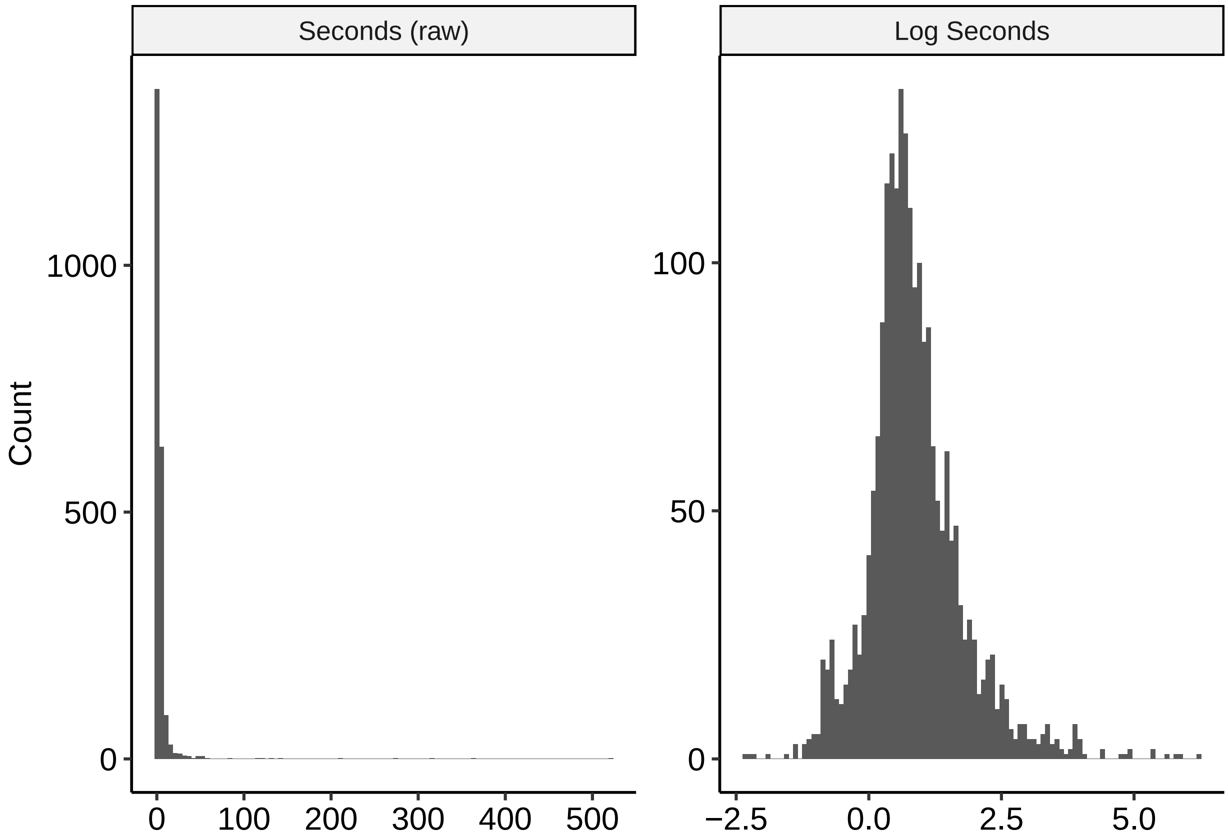
```
item_timing = str_subset(
  names(data),
  "^t_([[:alpha:]])+_[abcd]_(2_)?page_submit$"
)

item_timing = data %>%
  select(proid, all_of(item_timing), devicetype)
```

Next we reshape these data into long form.

```
item_timing = item_timing %>%
  gather(item, seconds, -proid, -devicetype) %>%
  mutate(item = str_remove(item, "^t_"),
         item = str_remove(item, "_2"),
         item = str_remove(item, "_page_submit$")) %>%
  separate(item, into = c("item", "format")) %>%
  filter(!is.na(seconds))
```

Seconds appears to be strongly skewed, so we log-transform this variable.

```
item_timing = item_timing %>%
  mutate(seconds_log = log(seconds))

item_timing %>%
  select(contains("seconds")) %>%
  gather(variable, value) %>%
  mutate(variable = factor(variable,
                           levels = c("seconds", "seconds_log"),
                           labels = c("Seconds (raw)", "Log Seconds"))) %>%
  ggplot(aes(x = value)) +
  geom_histogram(bins = 100) +
  facet_wrap(~variable, scales = "free") +
  labs(x = NULL, y = "Count") +
  theme_pubr()
```



We give labels to the formats, to clarify interpretations and aid table and figure construction.

```
item_timing$format = as.factor(item_timing$format)
item_timing$format = relevel(item_timing$format, ref = "a")
item_timing$format = factor(item_timing$format,
                              levels = c("a","b","c","d"),
                              labels = c("Adjective\nOnly", "Am\nAdjective", "Tend to be\nAdjective",
```

Responses

Response by device

We used a multilevel model, nesting response within participant to account for dependence. Our primary predictor was device format

```
mod.response = lmer(response~devicetype + (1|proid),
                    data = item_responses)
anova(mod.response)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##            Sum Sq Mean Sq NumDF DenDF F value Pr(>F)
## devicetype  4.008   2.004     2    32  1.1793 0.3205
```

```
plot1 = plot_model(mod.response, type = "pred")
```

```
plot1$devicetype +
  labs(x = NULL,
       y = "Average response",
       title = "Average responses by device") +
  theme_pubclean()
```

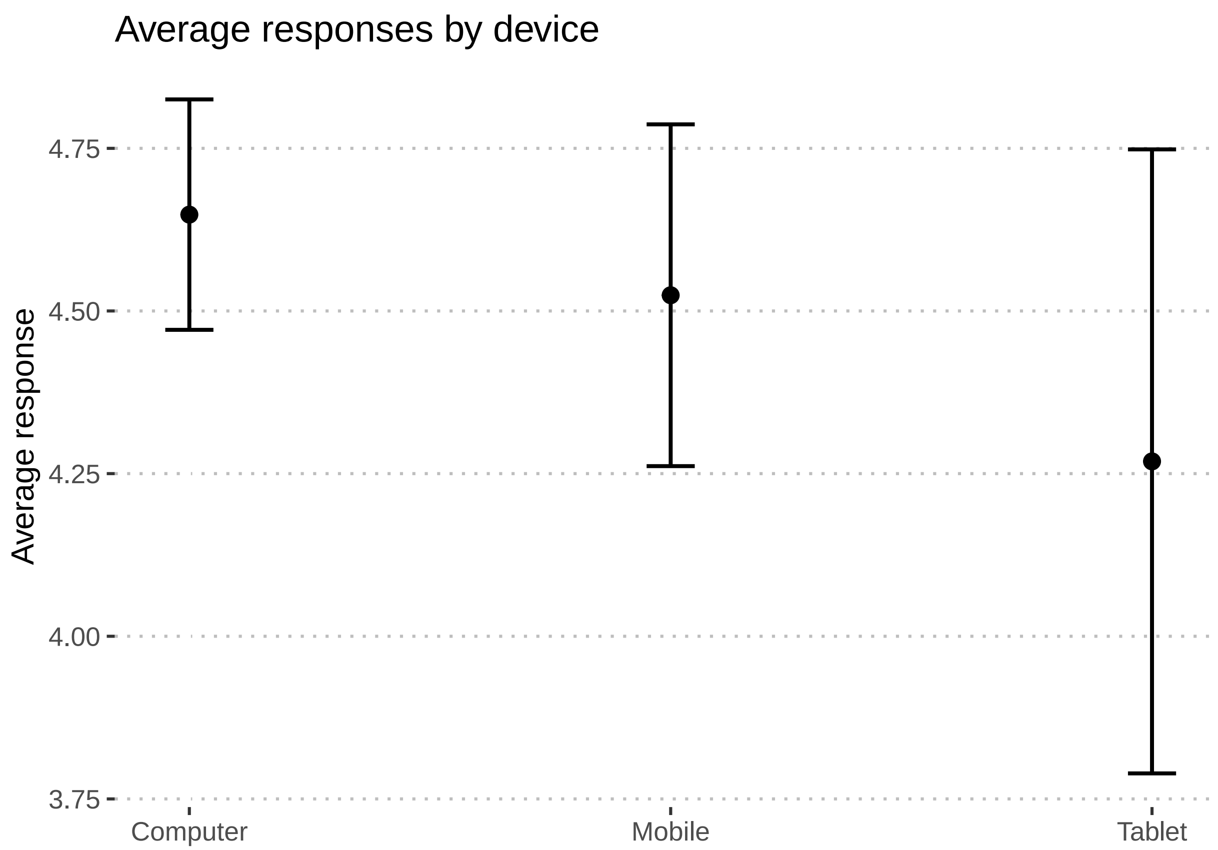


Figure 39: Predicted response on personality items by condition.

```
means_by_group = item_responses %>%
  group_by(devicetype) %>%
  summarise(m = mean(response),
            s = sd(response))

item_responses %>%
  ggplot(aes(x = response)) +
  geom_histogram(aes(fill = block),
                 position = "dodge",
                 bins = 6, color = "white") +
  geom_vline(aes(xintercept = m),
             data = means_by_group) +
  facet_wrap(~devicetype, scales = "free_y") +
  #guides(fill = "none") +
  scale_x_continuous(breaks = 1:6) +
  labs(y = "Number of particpants",
       title = "Distribution of responses by format") +
```

```
  theme_pubr()
```

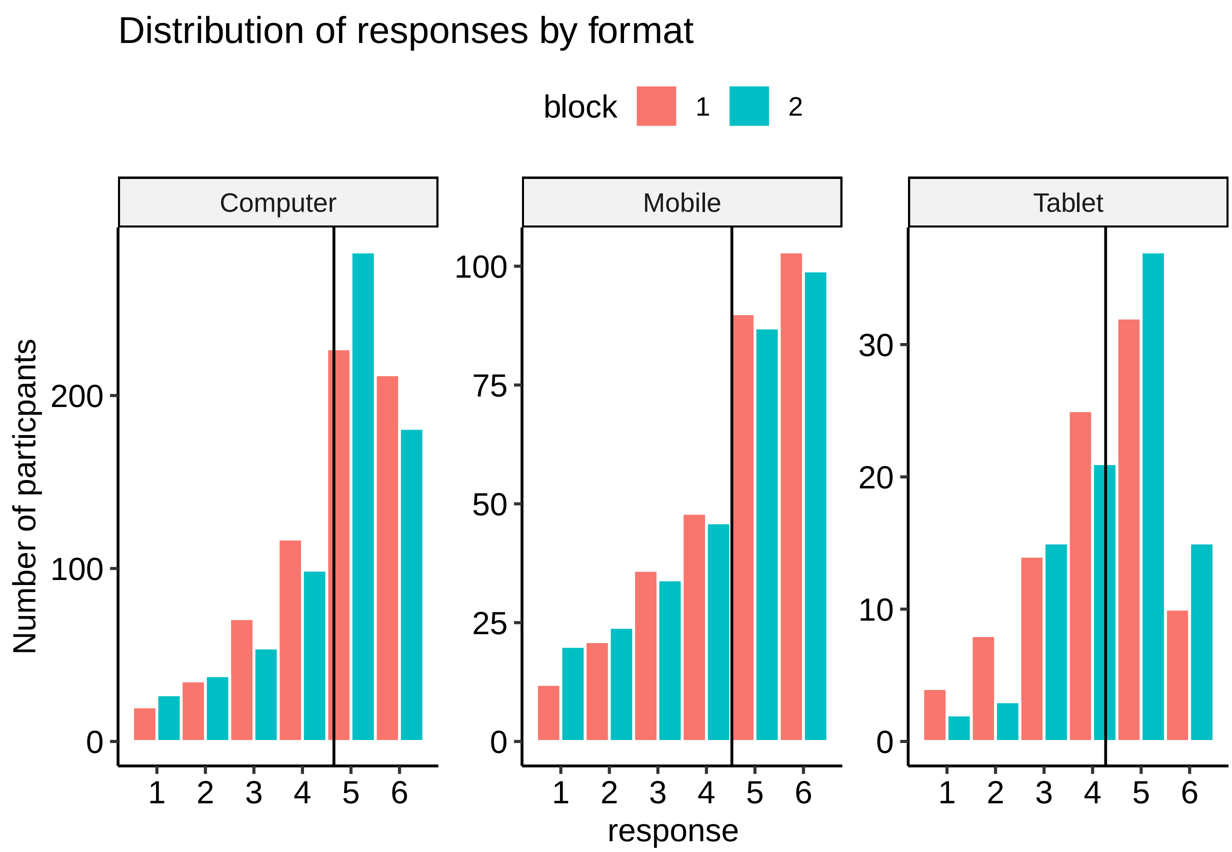


Figure 40: Distribution of responses by category

Device by format

```
mod.response2 = lmer(response~devicetype*format + (1|proid),
                     data = item_responses)
anova(mod.response2)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##                    Sum Sq Mean Sq NumDF   DenDF F value  Pr(>F)
## devicetype         2.4435  1.2217     2   33.06  0.7204 0.49405
## format             5.0203  1.6734     3 2030.62  0.9867 0.39803
## devicetype:format 18.9498  3.1583     6 2033.40  1.8622 0.08374 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
plot2 = plot_model(mod.response2, type = "pred", terms = c("format", "devicetype"))

plot2 +
  labs(x = NULL,
```

```
    y = "Average response",
    title = "Average responses by device") +
  theme_pubclean()
```

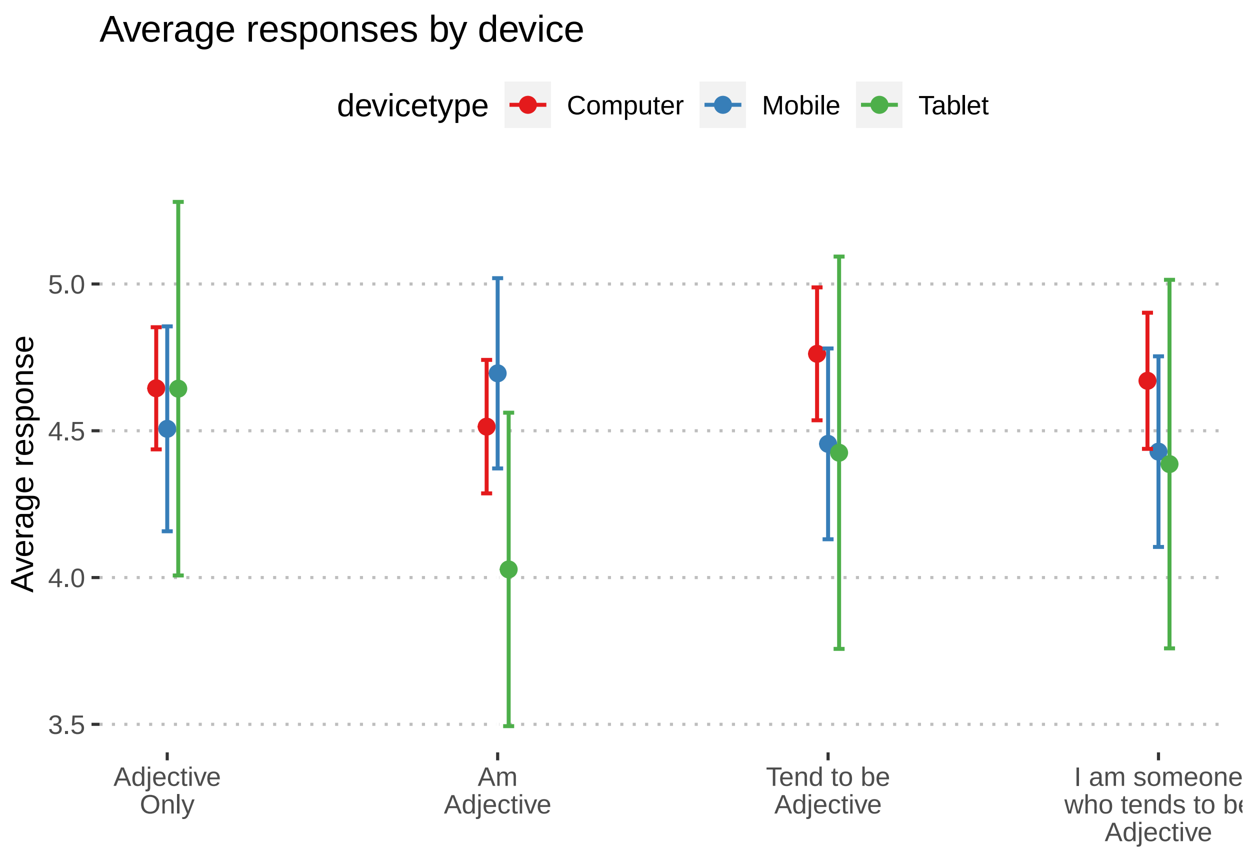


Figure 41: Predicted response on personality items by condition.

Timing

Timing by device

We used a multilevel model, nesting seconds within participant to account for dependence. Our primary predictor was format.

```
mod.timing = lmer(seconds_log~devicetype + (1|proid),
                  data = item_timing)
anova(mod.timing)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##            Sum Sq Mean Sq NumDF DenDF F value Pr(>F)
## devicetype 2.8619  1.4309     2    32  2.0649 0.1434
```

```
plot1 = plot_model(mod.timing, type = "pred")

plot1$devicetype +
  labs(x = NULL,
       y = "Seconds (log)",
       title = "Average seconds time by device type") +
  theme_pubclean()
```

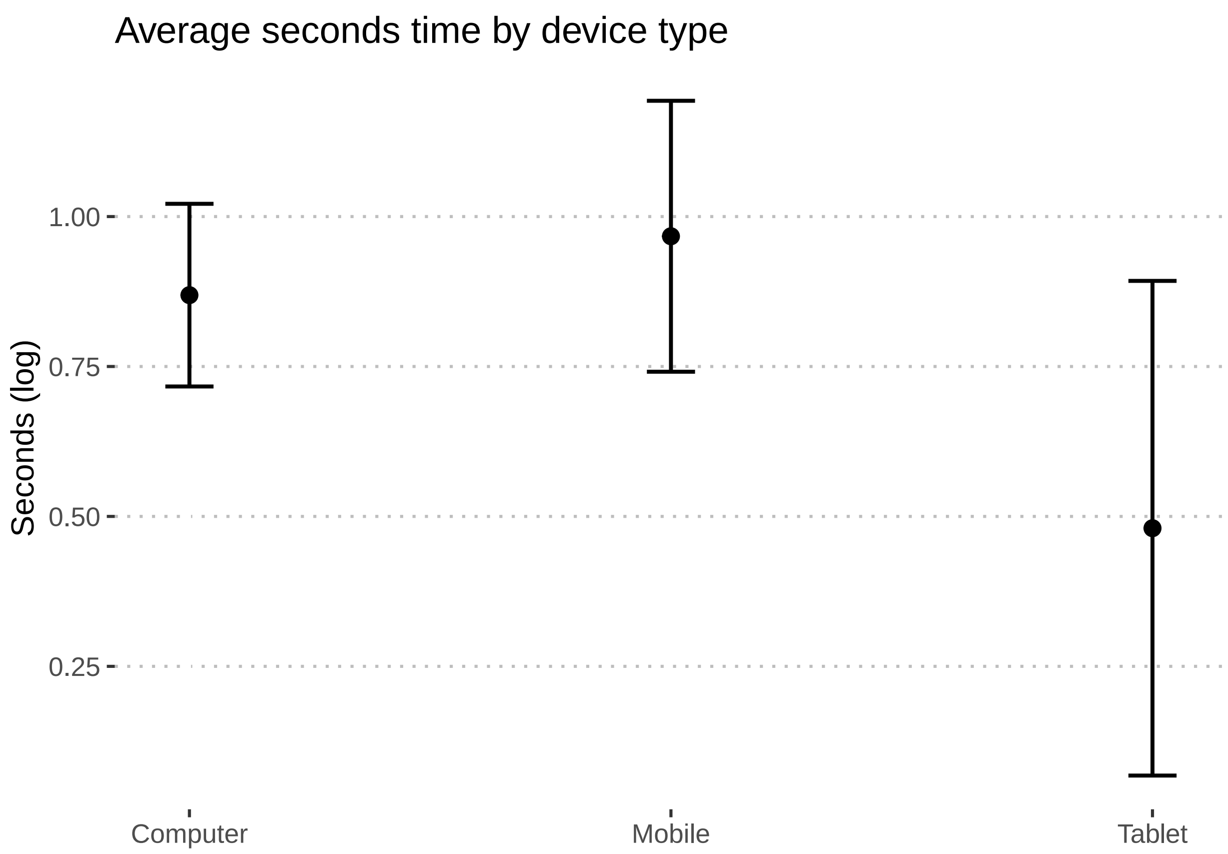


Figure 42: Predicted seconds on personality items by condition.

```
means_by_group = item_timing %>%
  group_by(devicetype) %>%
  summarise(m = mean(seconds),
            s = sd(seconds))

item_timing %>%
  ggplot(aes(x = seconds, fill = devicetype)) +
  geom_histogram(bins = 100) +
  geom_vline(aes(xintercept = m), data = means_by_group) +
  facet_wrap(~devicetype, scales = "free_y") +
  guides(fill = "none") +
  scale_x_log10() +
  labs(y = "Number of particpants",
       title = "Distribution of seconds by format",
       x = "Seconds (logrithmic scale)") +
```

```
  theme_pubr()
```

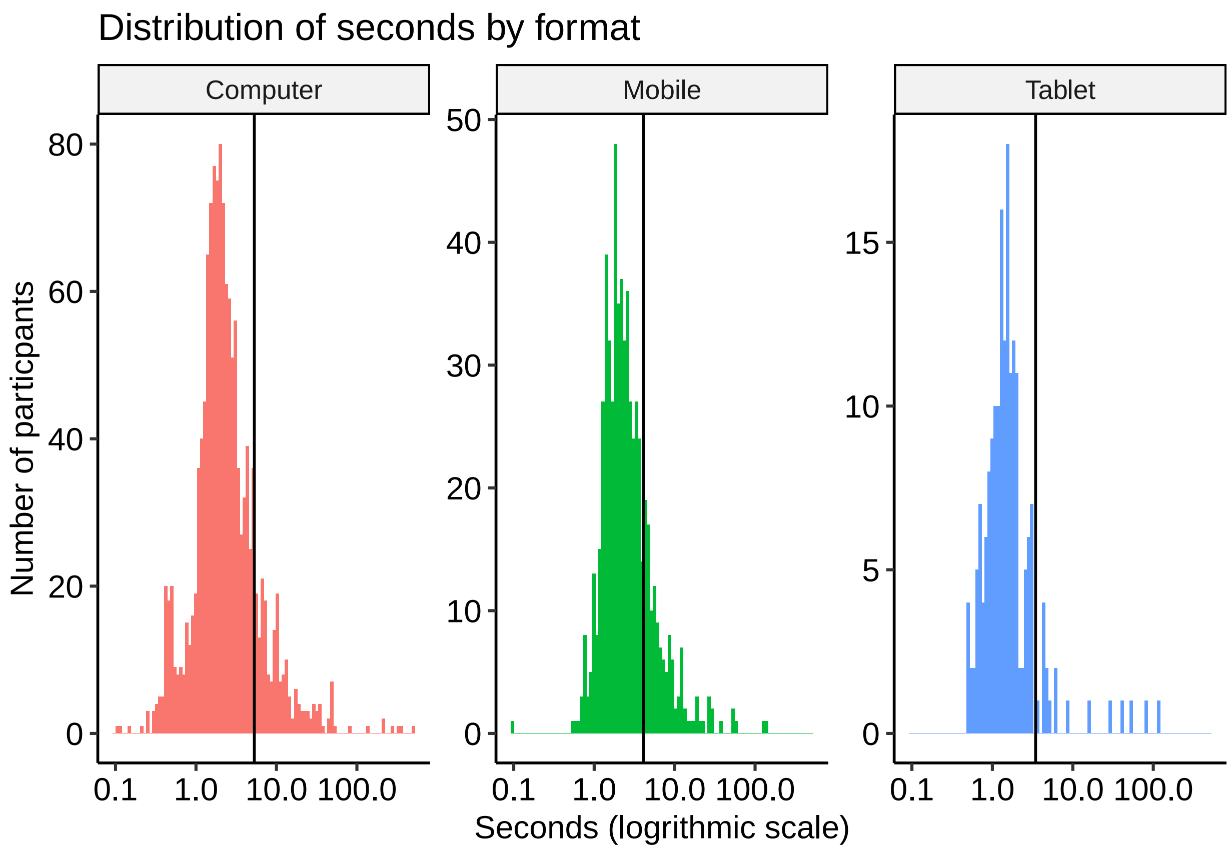


Figure 43: Distribution of secondss by category

Device by format

```
mod.timing2 = lmer(seconds_log~devicetype*format + (1|proid),
                   data = item_timing)
anova(mod.timing2)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##                    Sum Sq Mean Sq NumDF   DenDF F value    Pr(>F)
## devicetype         3.4339  1.7170     2   32.31  2.5424 0.0943136 .
## format            12.5878  4.1959     3 2138.42  6.2132 0.0003366 ***
## devicetype:format 14.5245  2.4208     6 2142.58  3.5846 0.0015414 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
plot1 = plot_model(mod.timing2, type = "pred", terms = c("format", "devicetype"))

plot1 +
  labs(x = NULL,
```

```
       y = "Seconds (log)",
       title = "Average seconds time by device type") +
  theme_pubclean()
```

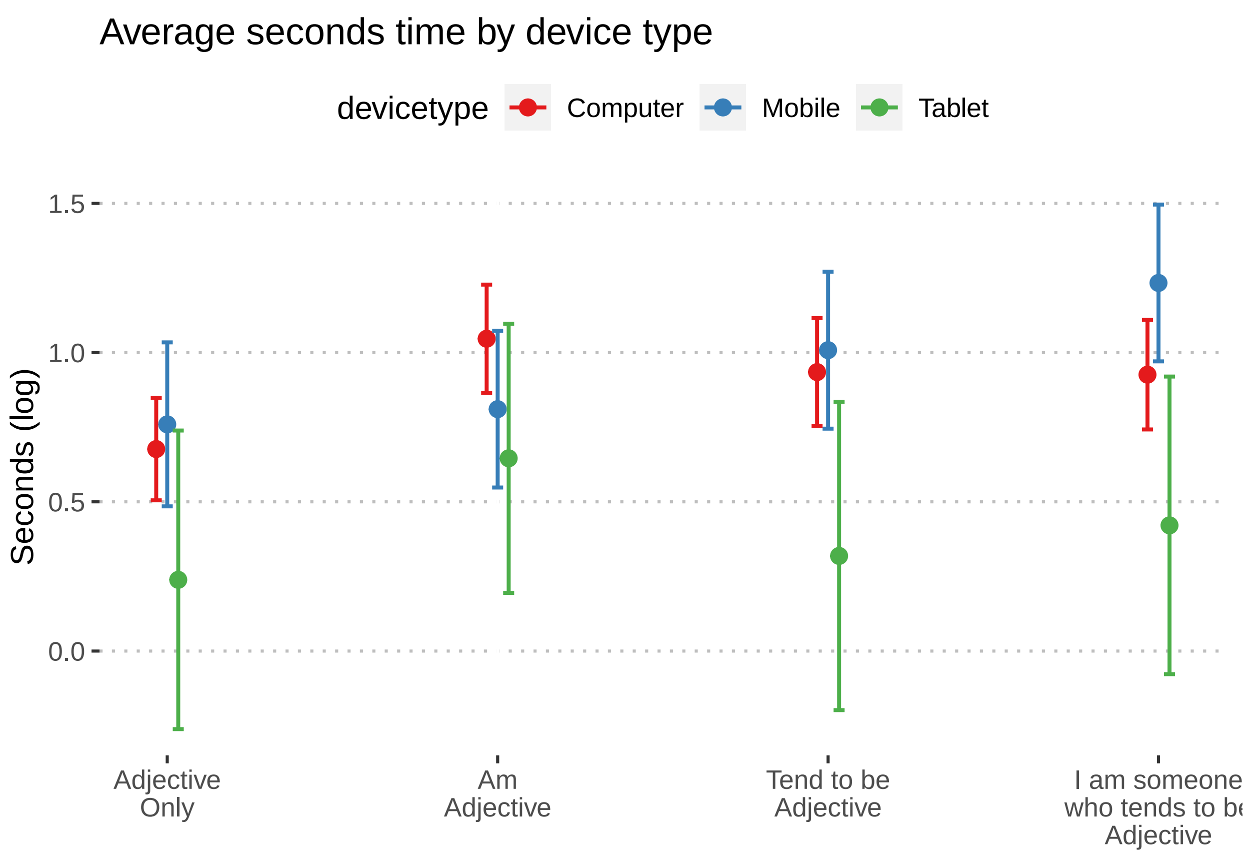


Figure 44: Predicted seconds on personality items by condition.

Power analysis

We conduct power analyses for the main research question – does formatting affect response to personality items – using a simulation method. That is, we generate datasets of varying sample sizes (from as few as 50 participants per condition to as many as 100), then simulate responses based on the models fit to the pilot data.

Model 1

To simplify our code, we write a function that simulates responses to model 1 based on a given sample size, N, and number of repititions.

```
# function to simulate mod.format_b1

sim_format_b1 = function(n, sims){
  p_vals = numeric(length = sims)

  sim_a = expand_grid(
    proid = as.character(1:n),
    item = c(1:33),
    format = "Adjective\nOnly"
  )

  sim_b = expand_grid(
    proid = as.character((n+1):(2*n)),
    item = c(1:33),
    format = "Am\nAdjective"
  )

  sim_c = expand_grid(
    proid = as.character(((2*n)+1):(3*n)),
    item = c(1:33),
    format = "Tend to be\nAdjective"
  )
  sim_d = expand_grid(
    proid = as.character(((3*n)+1):(4*n)),
    item = c(1:33),
    format = "I am someone\nwho tends to be\nAdjective"
  )

  sim_data = rbind(sim_a, sim_b) %>% rbind(sim_c) %>% rbind(sim_d)
  for (i in 1:sims){
    sim_data$response = simulate(mod.format_b1, newdata = sim_data, allow.new.levels = T)[,1]
    sim_mod = lmer(response~format + (1|proid), data = sim_data)
    p_vals[i] = anova(sim_mod)["format", 6]}
  return(p_vals)
}
```

Next we identify the sample sizes for simulation (from 50 to 500 by 25) and create a data frame to hold the results. Power represents the proportion of simulations for which p is less than .05.

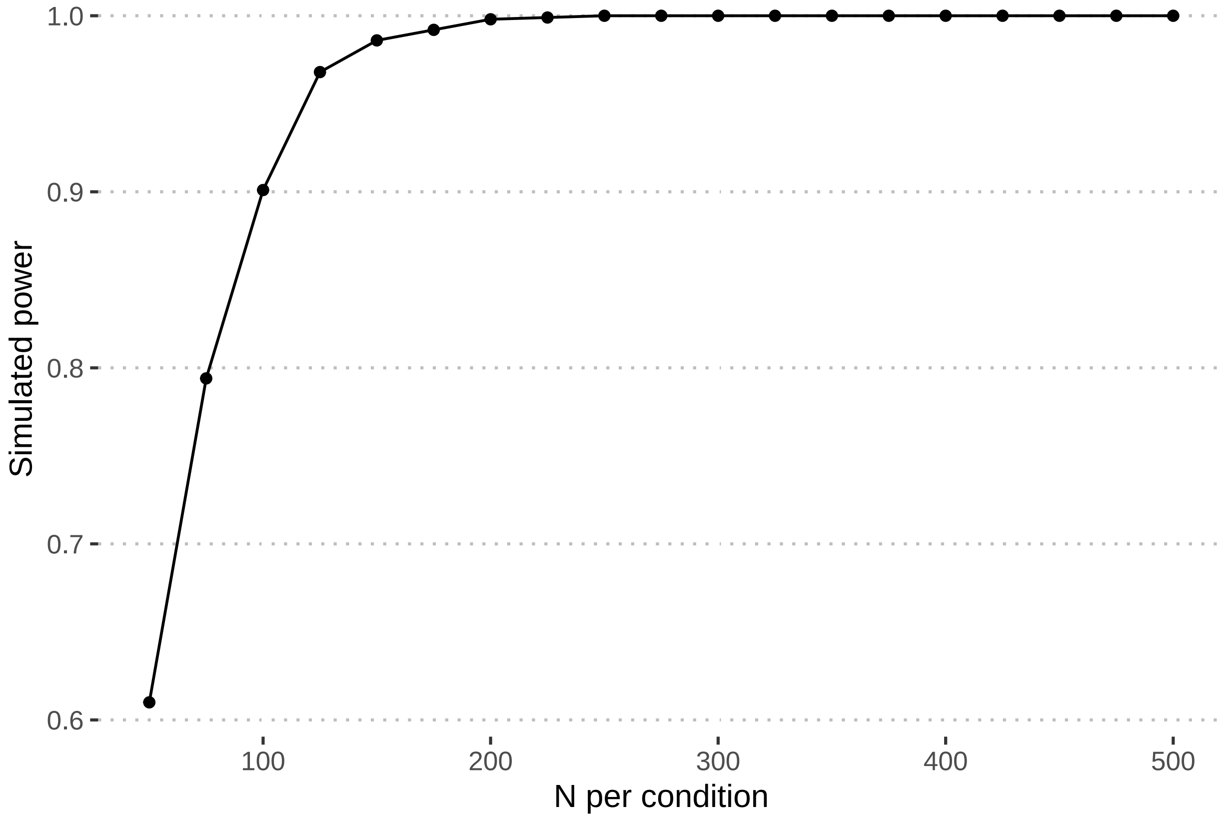
```
# simulate at various sample sizes
# n = number per condition

sample_sizes = seq(50, 500, 25)

n_sims = 1000

power_df = data.frame(
  N = sample_sizes,
  power = 0
)
```

Here we (inefficiently) loop through all sample sizes and calculate power.

```
set.seed(20210729)
for(i in sample_sizes){
  pvalues = sim_format_b1(i, n_sims)
  sig = ifelse(pvalues < .05, 1, 0)
  power_df$power[power_df$N == i] <- sum(sig)/n_sims
}
```

Finally, we plot these effects to determine needed sample size.

```
power_df %>%
  ggplot(aes(x = N, y = power)) +
  geom_line() +
  geom_point() +
  labs(
    x = "N per condition",
    y = "Simulated power"
  ) +
  theme_pubclean()
```




```
#identify minimum sample size

power_df_min = power_df %>%
  filter(power > .95)

N_min = min(power_df_min$N)
```

The simulation suggests that power would be over the threshold of .95 with a sample size of 125 participants per condition.

Model 2

Here we repeat the process for our second model, which uses both blocks of data.

```
# function to simulate mod.format_b2

sim_format_b2 = function(n, sims){
  p_vals = numeric(length = sims)

  sim_a_b2 = expand_grid(
    proid = as.character(1:n),
    item = c(1:33),
    format = "Adjective\nOnly",
    block = "1"
  )
```

```
  sim_b_b2 = expand_grid(
    proid = as.character((n+1):(2*n)),
    item = c(1:33),
    format = "Am\nAdjective",
    block = "1"
  )

  sim_c_b2 = expand_grid(
    proid = as.character(((2*n)+1):(3*n)),
    item = c(1:33),
    format = "Tend to be\nAdjective",
    block = "1"
  )
  sim_d_b2 = expand_grid(
    proid = as.character(((3*n)+1):(4*n)),
    item = c(1:33),
    format = "I am someone\nwho tends to be\nAdjective",
    block = "1"
  )
  sim_b2 = expand_grid(
    proid = as.character(1:(4*n)),
    item = c(1:33),
    block = "2"
  )
  sim_b2$format = sample(
    x = c("Adjective\nOnly",
          "Am\nAdjective",
          "Tend to be\nAdjective",
          "I am someone\nwho tends to be\nAdjective"),
    size = 33*n*4,
    replace = TRUE
  )

  sim_data = full_join(sim_a_b2, sim_b_b2) %>%
    full_join(sim_c_b2) %>%
    full_join(sim_d_b2) %>%
    full_join(sim_b2)

  for (i in 1:sims){
    sim_data$response = simulate(mod.format_b2,
                                 newdata = sim_data,
                                 allow.new.levels = T)[,1]
    sim_mod = lmer(response~format + (1|proid),
                   data = sim_data)
    p_vals[i] = anova(sim_mod)["format", 6]}
  return(p_vals)
}
```

We use the same sample sizes and simulation length for these analyses, so we start by creating a new data frame.

```
power_df_2 = data.frame(
  N = sample_sizes,
```

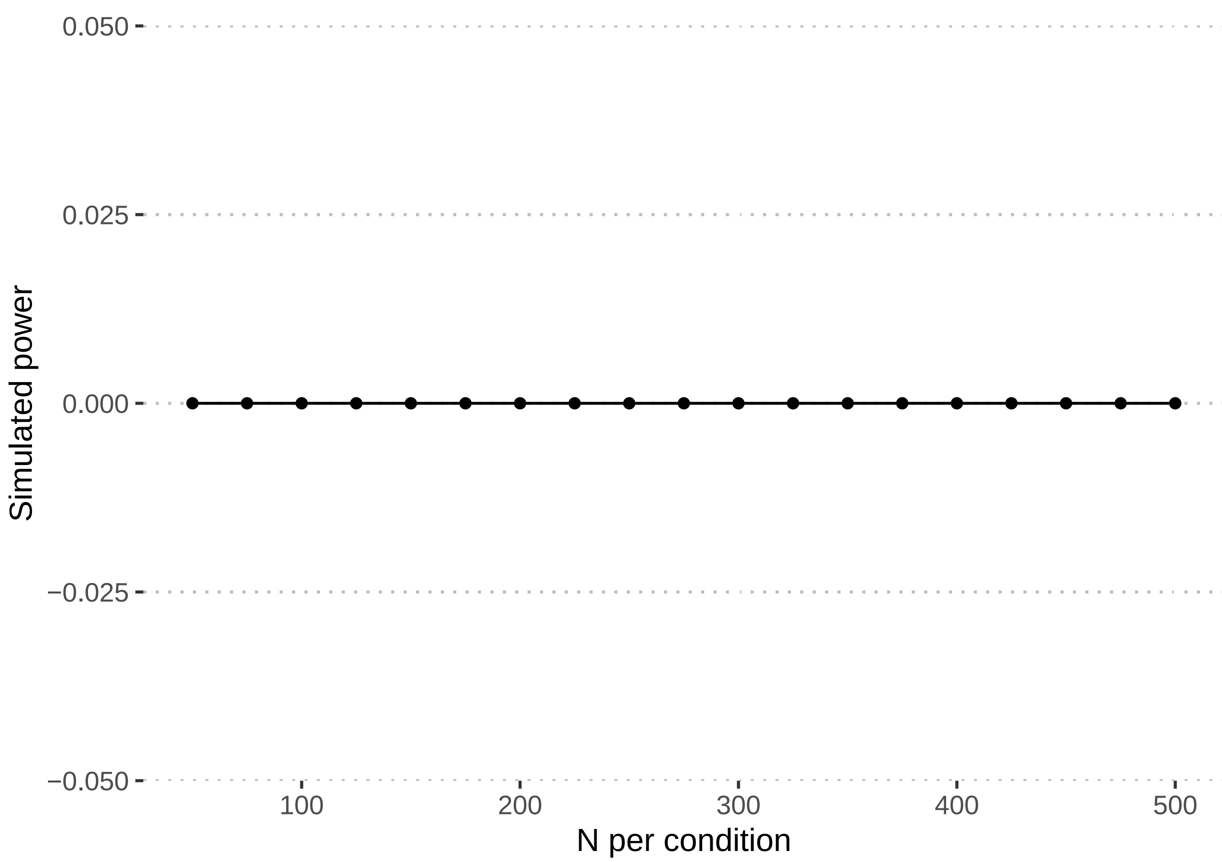
```
  power = 0
)
```

Here we (inefficiently) loop through all sample sizes and calculate power.

```
set.seed(20210729)
for(i in sample_sizes){
  pvalues = sim_format_b2(i, n_sims)
  sig = ifelse(pvalues < .05, 1, 0)
  power_df_2$power[power_df_2$N == i] <- sum(sig)/n_sims
}
```

Finally, we plot these effects to determine needed sample size.

```
power_df_2 %>%
  ggplot(aes(x = N, y = power)) +
  geom_line() +
  geom_point() +
  labs(
    x = "N per condition",
    y = "Simulated power"
  ) +
  theme_pubclean()
```



```r
#identify minimum sample size

power_df2_min = power_df_2 %>%
  filter(power > .95)

N_min2 = min(power_df2_min$N)
```

The simulation suggests that power would be over the threshold of .95 with a sample size of ∞ participants per condition.