Supplemental file

Last updated 2021-08-10

Contents

1 Cleaning

1.1 Workspace

```
library(here) # for working with files
library(tidyverse) # for cleaning
library(janitor) # for variable names
library(stringi) # for generating random strings
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
library(psych) # for correlation tests
library(broom.mixed) # for tidying mulitievel models
```

1.2 Time 1

```
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"))
```

1.2.1 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.

```
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.

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

Now we merge this back into the data.

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

1.2.2 Drop bots

1.2.2.1 Based on ID Prolific IDs must also be a certain length. We remove IDs that are not sufficiently long.

```
data = data %>%
  mutate(id_length = nchar(proid)) %>%
  filter(id_length > 20) %>%
  select(-id_length)
```

We removed 5 participants without valid Prolific IDs.

```
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 != "")
```

1.2.2.2 Based on language We removed 0 participants that do not speak english well or very well.

1.2.2.3 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.

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

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

#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
```

```
# 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.

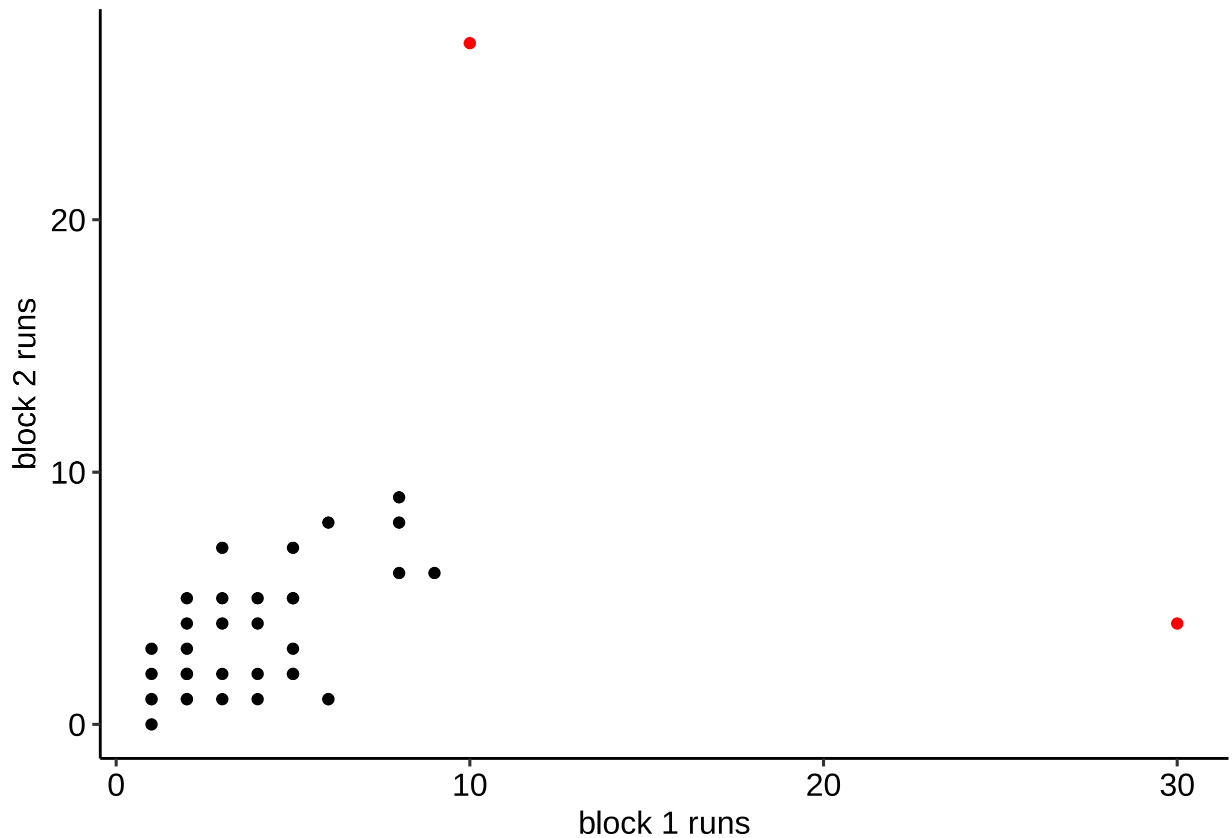


Figure 1: Maximum number of same consecutive responses in personality blocks.

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

rm(runs_data)
```

1.2.2.4 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 (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)
```

1.2.2.5 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."

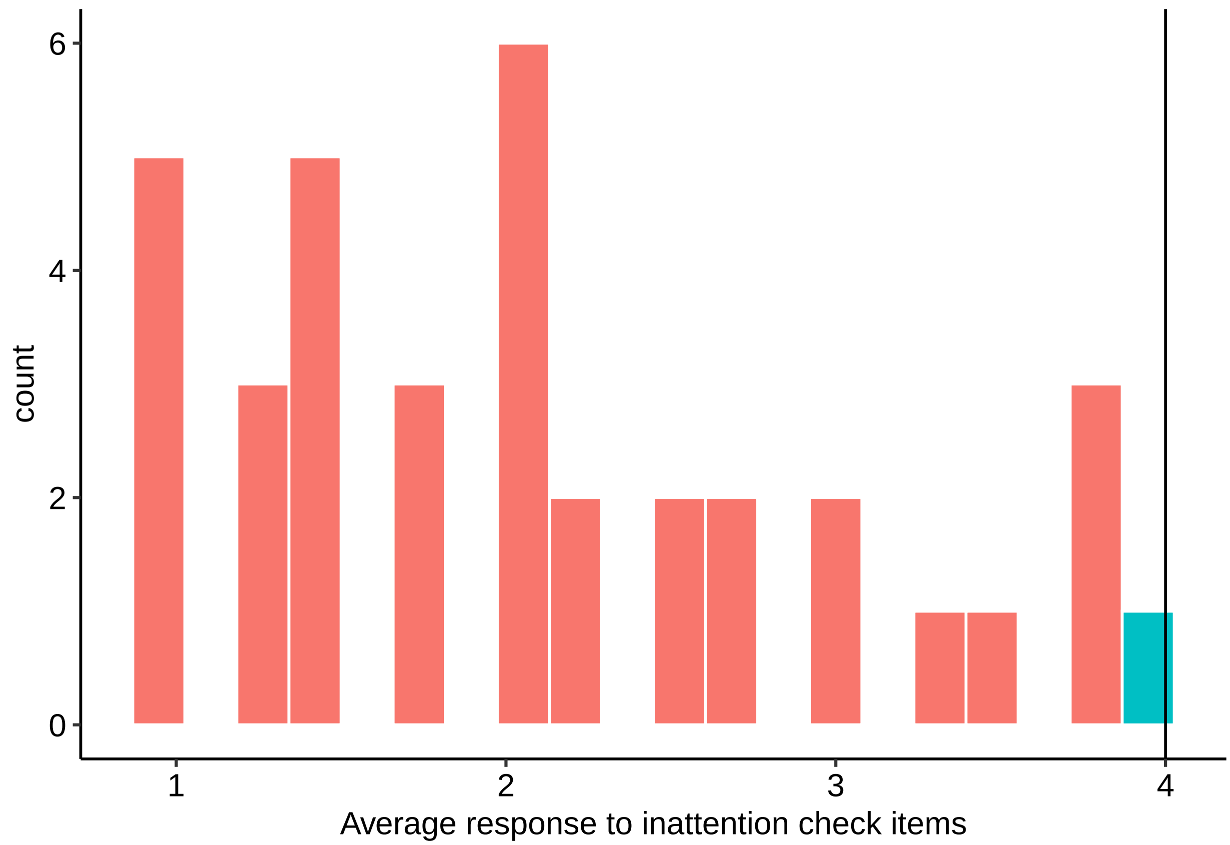


Figure 2: Average response to inattention check items

```
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))
```

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.

We create a variable which indicates the Block 1 condition of each participant. This is used in two places: first, in recruitinig participants at Time 2 (participants are given the same format at Time 2 as they received in Block 1), and second, in selecting the corret items during the test-retest analyses.

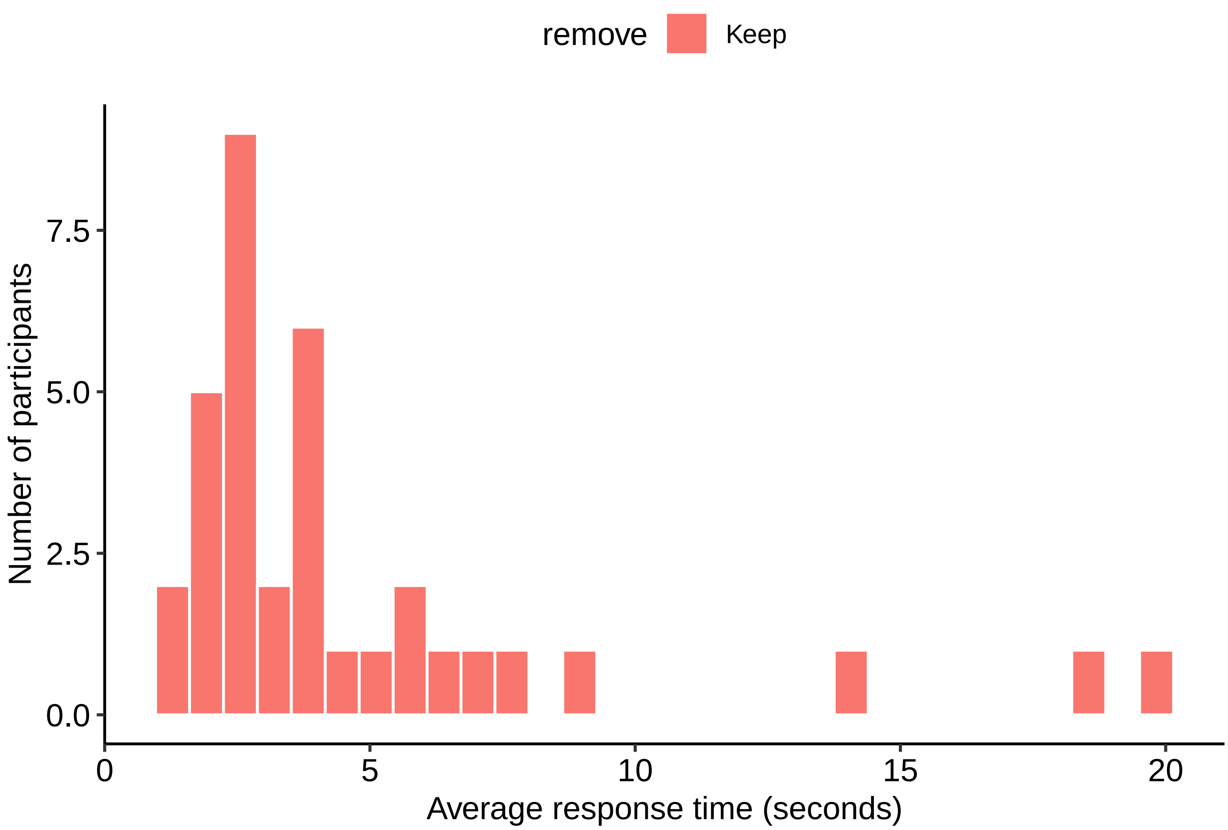


Figure 3: (#fig:timing_dist)Distribution of average time to respond to personality items.

```
data = data %>%
  mutate(condition = case_when(
    !is.na(outgoing_a) ~ "A",
    !is.na(outgoing_b) ~ "B",
    !is.na(outgoing_c) ~ "C",
    !is.na(outgoing_d) ~ "D",
  ))
```

At this point, we'll extract the Prolific ID numbers. These participants will be eligible to take the survey at Time 2.

```
data %>%
  select(proid, condition) %>%
  write_csv(file = here("data/elligible_proid"))
```

1.3 Time 2

```
data_path_2A = here("data/Wording 2A_July 29, 2021_14.49.text.csv")

data_labels_2A = read_csv(data_path_2A)

data_2A = read_csv(data_path_2A,
                skip = 3,
                col_names = names(data_labels_2A))
rm(data_labels_2A)
data_2A = clean_names(data_2A)
```

```
data_path_2B = here("data/Wording 2B_August 4, 2021_18.49.text.csv")

data_labels_2B = read_csv(data_path_2B)

data_2B = read_csv(data_path_2B,
                skip = 3,
                col_names = names(data_labels_2B))

rm(data_labels_2B)
data_2B = clean_names(data_2B)
```

```
names(data_2B) = str_replace(names(data_2B), "q763", "proid")
```

```
data_path_2C = here("data/Wording 2C_August 3, 2021_18.02.csv")

data_labels_2C = read_csv(data_path_2C)

data_2C = read_csv(data_path_2C,
                skip = 3,
                col_names = names(data_labels_2C))
rm(data_labels_2C)
data_2C = clean_names(data_2C)
```

```r
data_path_2D = here("data/Wording 2D_July 29, 2021_14.55.text.csv")

data_labels_2D = read_csv(data_path_2D)

data_2D = read_csv(data_path_2D,
                skip = 3,
                col_names = names(data_labels_2D))
rm(data_labels_2D)
data_2D = clean_names(data_2D)
```

```r
data_2 = data_2A %>%
  full_join(data_2B) %>%
  full_join(data_2C) %>%
  full_join(data_2D)
```

Remove the following columns.

```r
data_2 = data_2 %>%
  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"))
```

```r
data_2 = data_2 %>%
  select(-contains("outgoing_b_3i"))
```

1.3.1 Recode personality item responses to numeric

We recode the responses to personality items, which we downloaded as text strings. Here, all items end with `_3` and sometimes with `i`.

```r
p_items_2 = str_extract(names(data_2), "^[[:alpha:]]*_[abcd]_3(i)?$")
p_items_2 = p_items_2[!is.na(p_items_2)]

personality_items_2 = select(data_2, proid, all_of(p_items_2))
```

We apply the recoding function to all personality items.

```r
personality_items_2 = personality_items_2 %>%
  mutate(
    across(!c(proid), recode_p))
```

Now we merge this back into the data_2.

```r
data_2 = select(data_2, -all_of(p_items_2))
data_2 = full_join(data_2, personality_items_2)
```

1.3.2 Drop bots

1.3.2.1 Based on ID We also check that the ID in time 2 matches an ID in time 1.

```r
data_2 = data_2 %>%
  filter(proid %in% data$proid)
```

We removed 13 participants without valid Prolific IDs.

```r
data_2 = data_2 %>%
  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 != "")
```

1.3.2.2 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_2 %>%
  str_remove_all("_.") %>%
  unique()

# extract block 3 questions
block3 = data_2 %>%
  select(proid, all_of(p_items_2))

#rename variables
n = 0
for(i in adjectives){
  n = n+1
  names(block3) = str_replace(names(block3), i, paste0("trait", str_pad(n, 2, pad = "0")))
}


block3 = block3 %>%
  gather(item, response, -proid) %>%
  filter(!is.na(response)) %>%
  mutate(item = str_remove(item, "_3(i)?$")) %>%
  separate(item, into = c("item", "format")) %>%
  #select(-format) %>%
  spread(item, response)

block3_runs = numeric(length = nrow(block3))

for(i in 1:nrow(block3)){
  run = 0
  maxrun = 0
```

```r
  for(j in 3:ncol(block3)){
    if(block3[i,j] == block3[i, j-1]){
      run = run+1
      if(run > maxrun) maxrun = run
      } else{ run = 0}
  }
  block3_runs[i] = maxrun
}

#add to data_2 frame
block3$block3_runs = block3_runs
```

```r
#combine results
runs_data_2 = block3 %>%
  select(proid, block3_runs) %>%
  mutate(
    remove = case_when(
      block3_runs >= 17 ~ "Remove",
      TRUE ~ "Keep"
    ))
```

```r
#visualize
runs_data_2 %>%
  ggplot(aes(block3_runs)) +
  geom_histogram(aes(fill = remove), bins = 10,color = "white") +
  scale_color_manual() +
  guides(fill = "none") +
  labs(x = "Maximum number of repeated answers",
       y = "Participant count") +
  theme_pubr()
```

There were 0 participants who provided the same answer 17 or more times in a row. These participants were removed from the analyses.

```r
data_2 = data_2 %>%
  full_join(select(runs_data_2, proid, remove)) %>%
  filter(remove != "Remove") %>%
  select(-remove)

rm(runs_data_2)
```

1.3.2.3 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 (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.

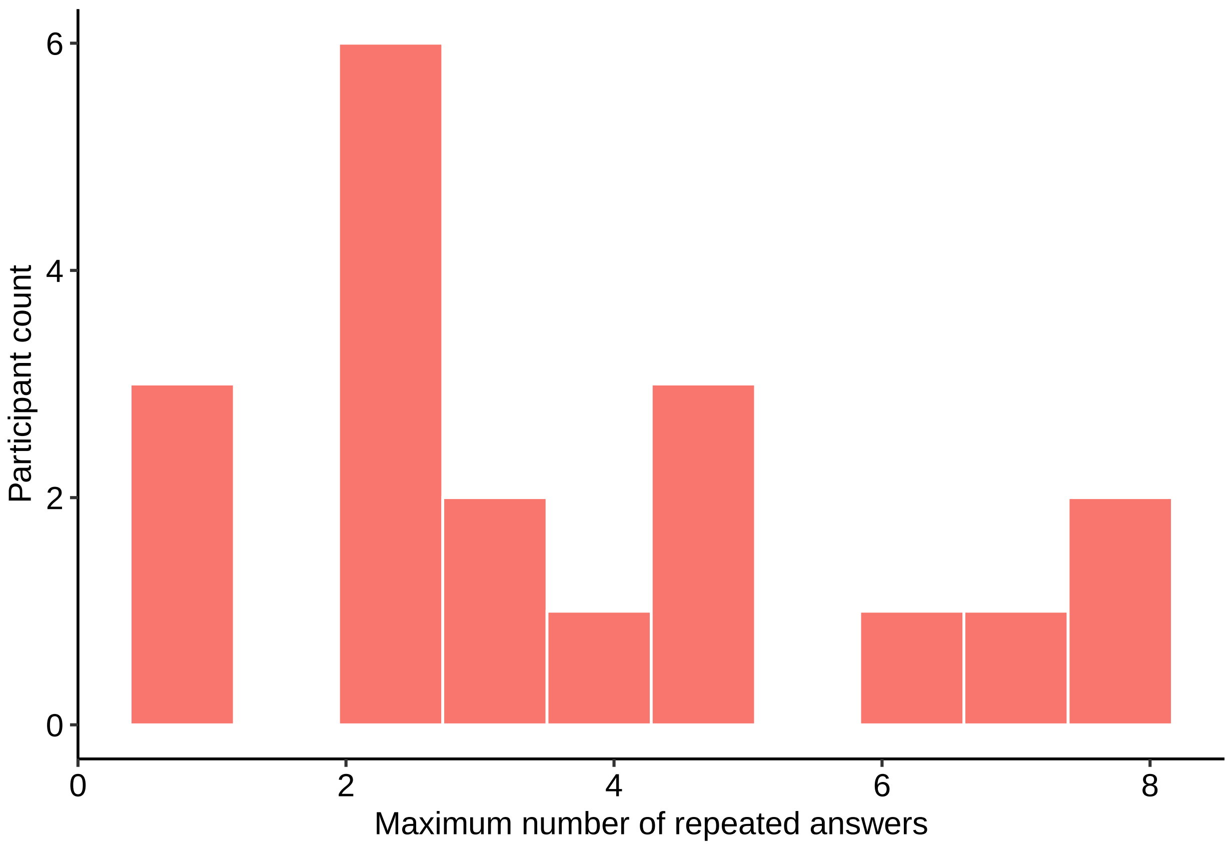


Figure 4: Maximum number of same consecutive responses in personality block 3.

```
in_average = data_2 %>%
  # 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()
```

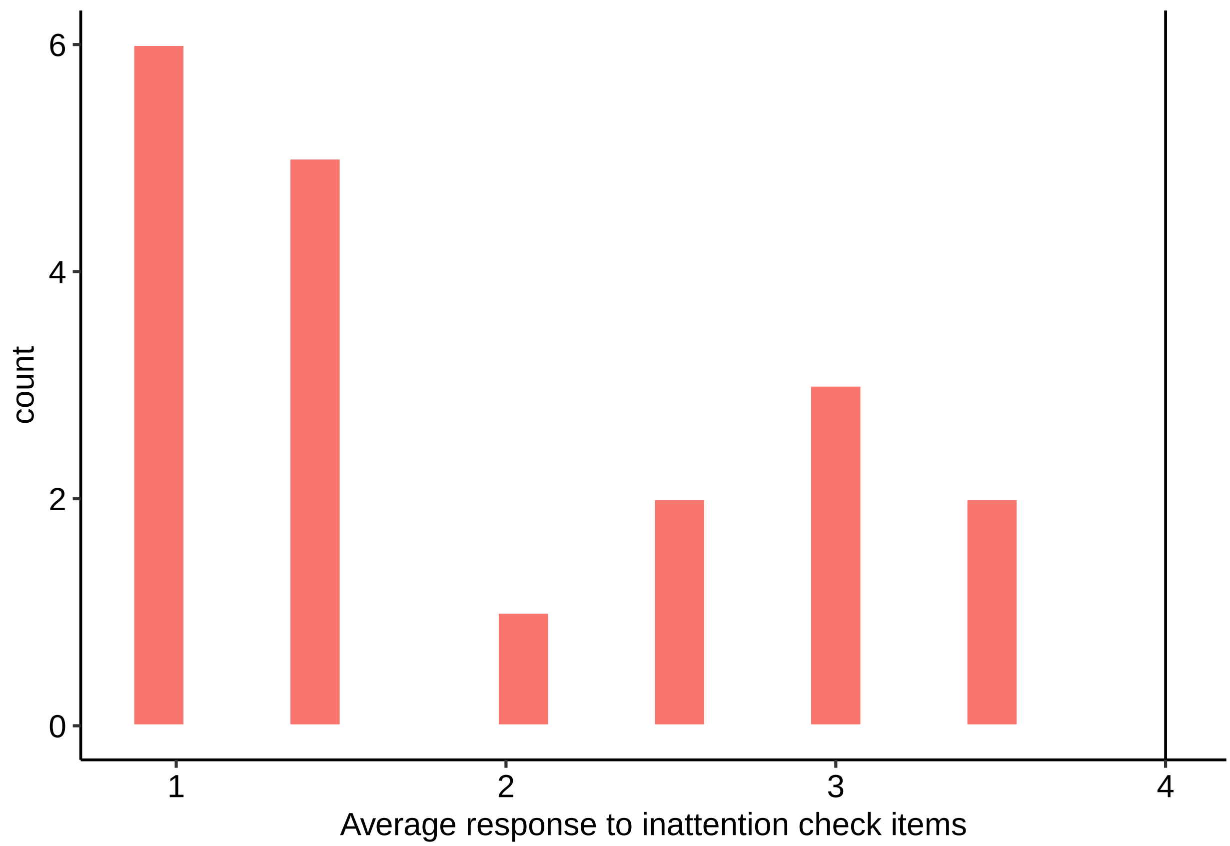


Figure 5: Average response to inattention check items

We remove 1 participants whose responses suggest inattention.

```
data_2 = data_2 %>%
  full_join(select(in_average, proid, remove)) %>%
  filter(remove != "Remove") %>%
  select(-remove)
```

1.3.2.4 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_2 = data_2 %>%
  select(proid, matches("t_[[:alpha:]]*_[abcd]_3(i)?_page_submit"))
```

Next we gather into long form and remove missing timing values

```
timing_data_2 = timing_data_2 %>%
  gather(variable, timing, -proid) %>%
  filter(!is.na(timing))
```

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

```
timing_data_2 %>%
  group_by(proid) %>%
  count() %>%
  ungroup() %>%
  summarise(min(n), max(n))
```

```
## # A tibble: 1 x 2
##   `min(n)` `max(n)`
##      <int>    <int>
## 1       31       31
```

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_2 = timing_data_2 %>%
  group_by(proid) %>%
  summarise(m_time = mean(timing)) %>%
  mutate(remove = case_when(
    m_time < 1 ~ "Remove",
    m_time > 30 ~ "Remove",
    TRUE ~ "Keep"
  ))
```

```
timing_data_2 %>%
  ggplot(aes(x = m_time, fill = remove)) +
  geom_histogram(color = "white") +
  labs(x = "Average response time (seconds)", y = "Number of participants") +
  theme_pubr()
```

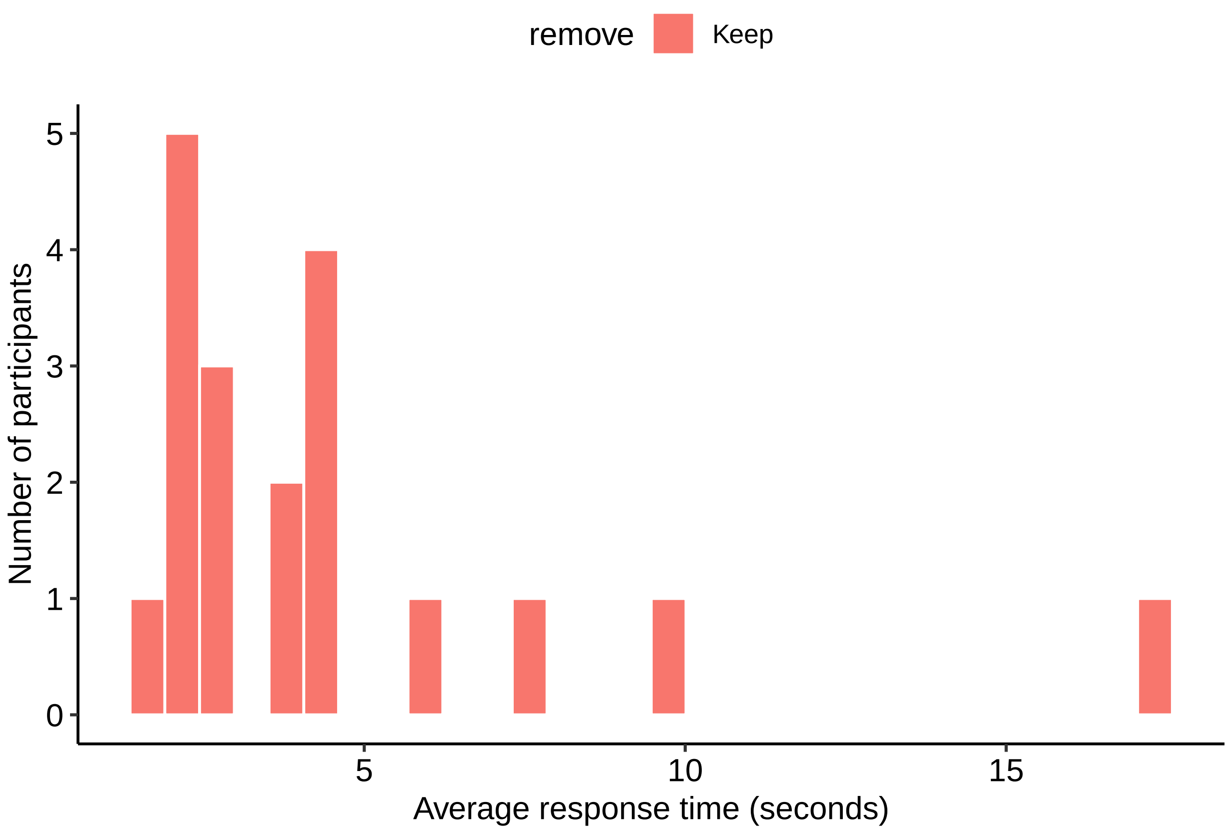


Figure 6: (#fig:timing_dist_2)Distribution of average time to respond to personality items in Block 3.

```r
data_2 = inner_join(data_2, filter(timing_data_2, remove == "Keep")) %>%
  select(-remove)
```

1.3.3 Merge all datasets together

```r
data_2 = data_2 %>%
  select(proid, very_delayed_recall, contains("_3")) %>%
  mutate(time2 = "yes")

data = data %>% full_join(data_2)
```

1.4 All data

1.4.1 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.

```r
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))
```

We also create a vector noting the items that are reverse scored. We use this later in tables, to help identify patterns when looking at analyses within-adjective.

```r
reverse = c("reckless", "moody", "worrying", "nervous", "careless", "impulsive")
```

1.4.2 Score memory task

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

```r
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 = ","))
```

1.4.2.1 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 ($SD = 2.73$),

```
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()
```

```
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()
```

1.4.2.2 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.

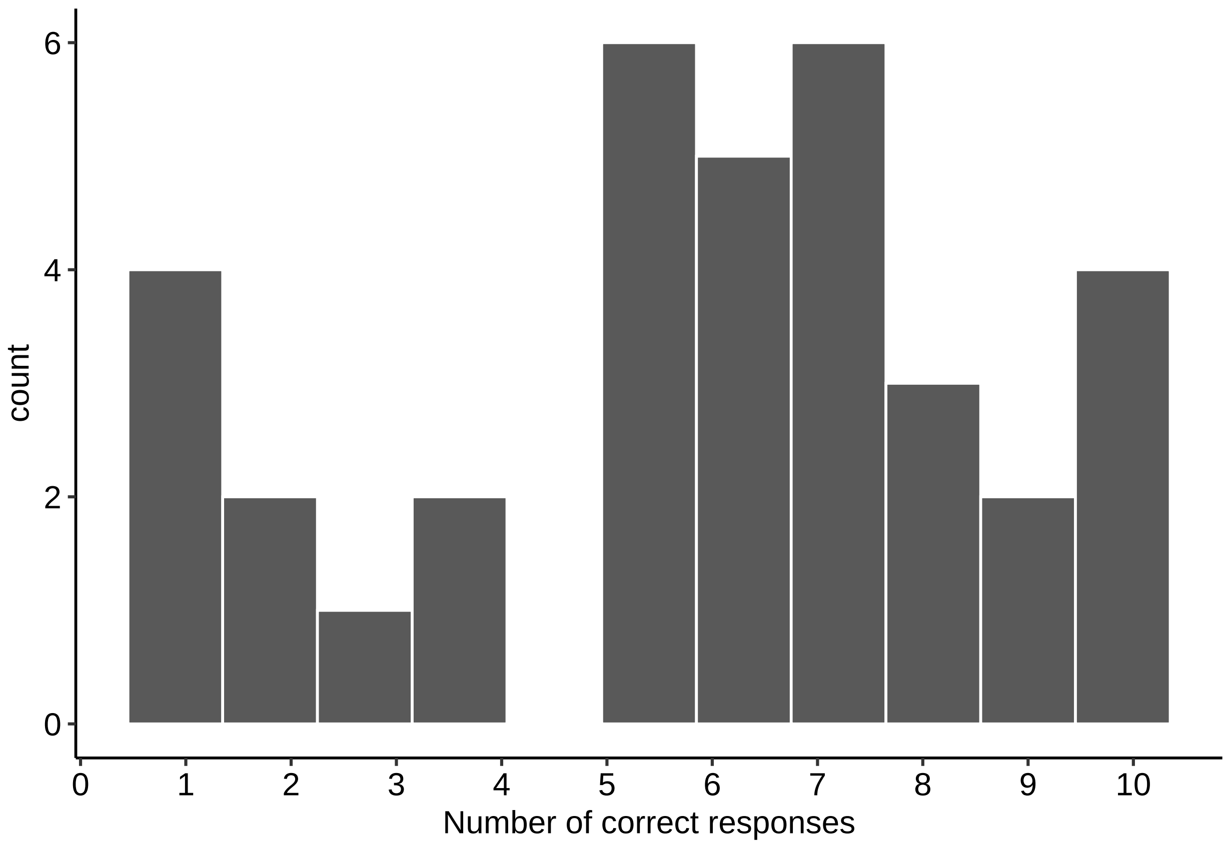


Figure 7: Correct responses on the memory task

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

```r
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, -newid)

data = inner_join(data, mem2)
```

```r
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()
```

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

1.4.2.3 Very-delayed recall Finally, we score the memory challenge posed at Time 2.

```r
mem3 = data %>%
  filter(time2 == "yes") %>%
  select(proid, mem_condition, very_delayed_recall) %>%
  mutate(newid = 1:nrow(.))
```

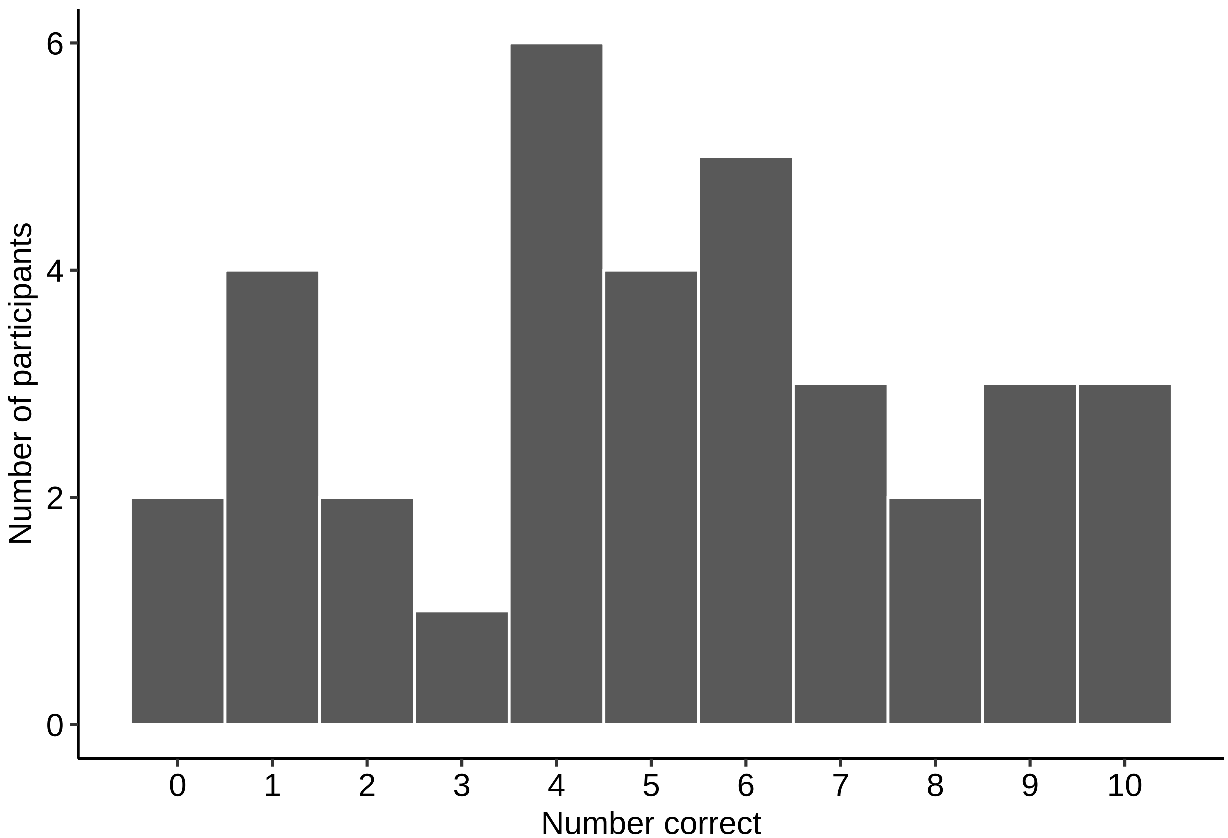


Figure 8: Distribution of delayed memory scores

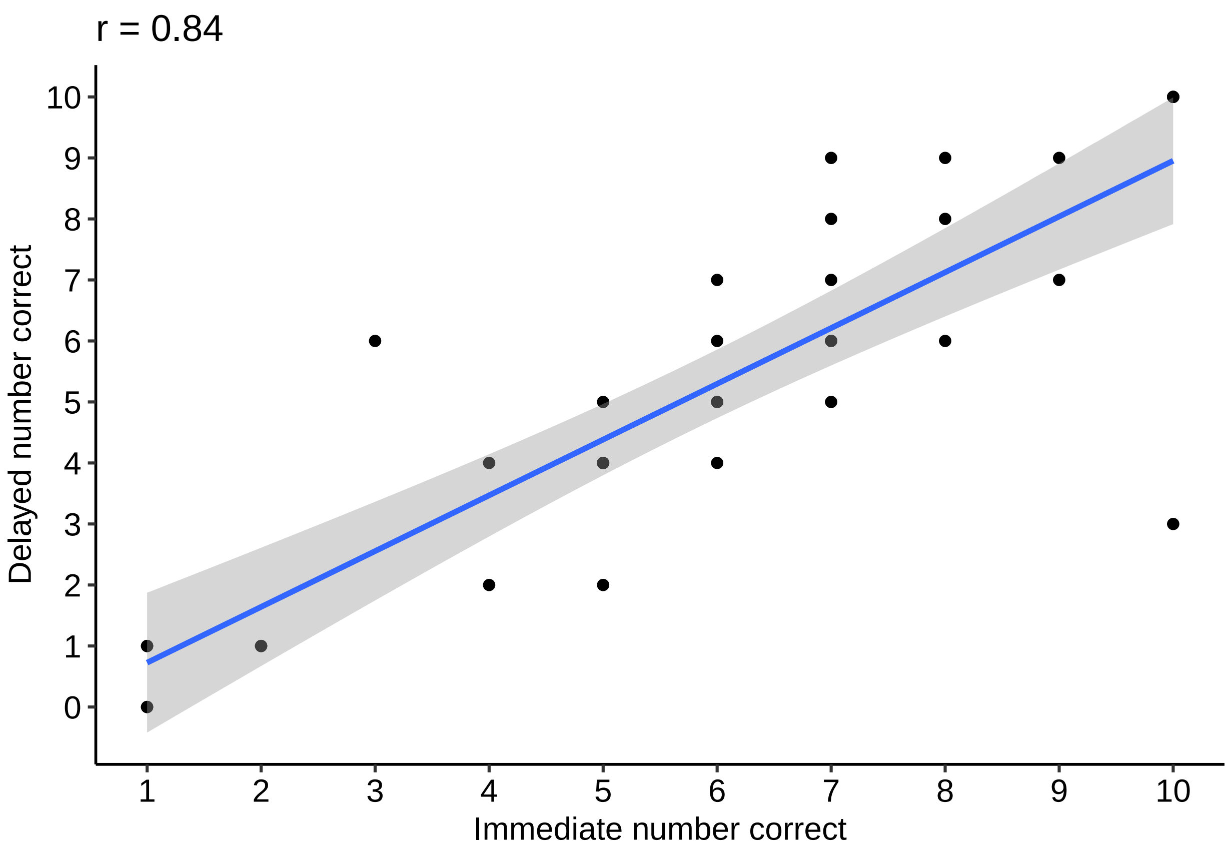


Figure 9: Relationship between immediate and delayed recall

```r
mem3 = mem3 %>%
  mutate(
    very_delayed_recall1 = map(very_delayed_recall, ~sapply(., amatch, correct1, maxDist = distance)),
    very_delayed_recall2 = map(very_delayed_recall, ~sapply(., amatch, correct2, maxDist = distance)),
    very_delayed_recall3 = map(very_delayed_recall, ~sapply(., amatch, correct3, maxDist = distance)),
    very_delayed_recall4 = map(very_delayed_recall, ~sapply(., amatch, correct4, maxDist = distance))
    ) %>%
  gather(variable, very_delayed_memory, very_delayed_recall1:very_delayed_recall4)

mem3 = mem3 %>%
  mutate(
      very_delayed_memory = map(very_delayed_memory, sapply,
                  FUN = function(x) ifelse(x >0, 1, 0)),
      # count the number of correct answers
      very_delayed_memory = map_dbl(very_delayed_memory, sum, na.rm=T))

mem3 = mem3 %>%
  group_by(proid) %>%
  filter(very_delayed_memory == max(very_delayed_memory)) %>%
  filter(row_number() == 1 ) %>%
  select(-very_delayed_recall, -variable, -newid)

data = full_join(data, mem3)
```

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

```r
data %>%
  ggplot(aes(x = memory, y = very_delayed_memory)) +
  geom_point() +
  geom_smooth(method = "lm") +
  scale_x_continuous("Immediate number correct", breaks = c(0:10)) +
  scale_y_continuous("Very delayed number correct", breaks = c(0:10)) +
  labs(title = paste0("r = ", printnum(cor(data$memory, data$delayed_memory, use = "pairwise")))) +
  theme_pubr()
```

```r
data %>%
  select(matches("memory$")) %>%
  corr.test
```

1.4.2.4 Correlations

```
## Call:corr.test(x = .)
## Correlation matrix
```

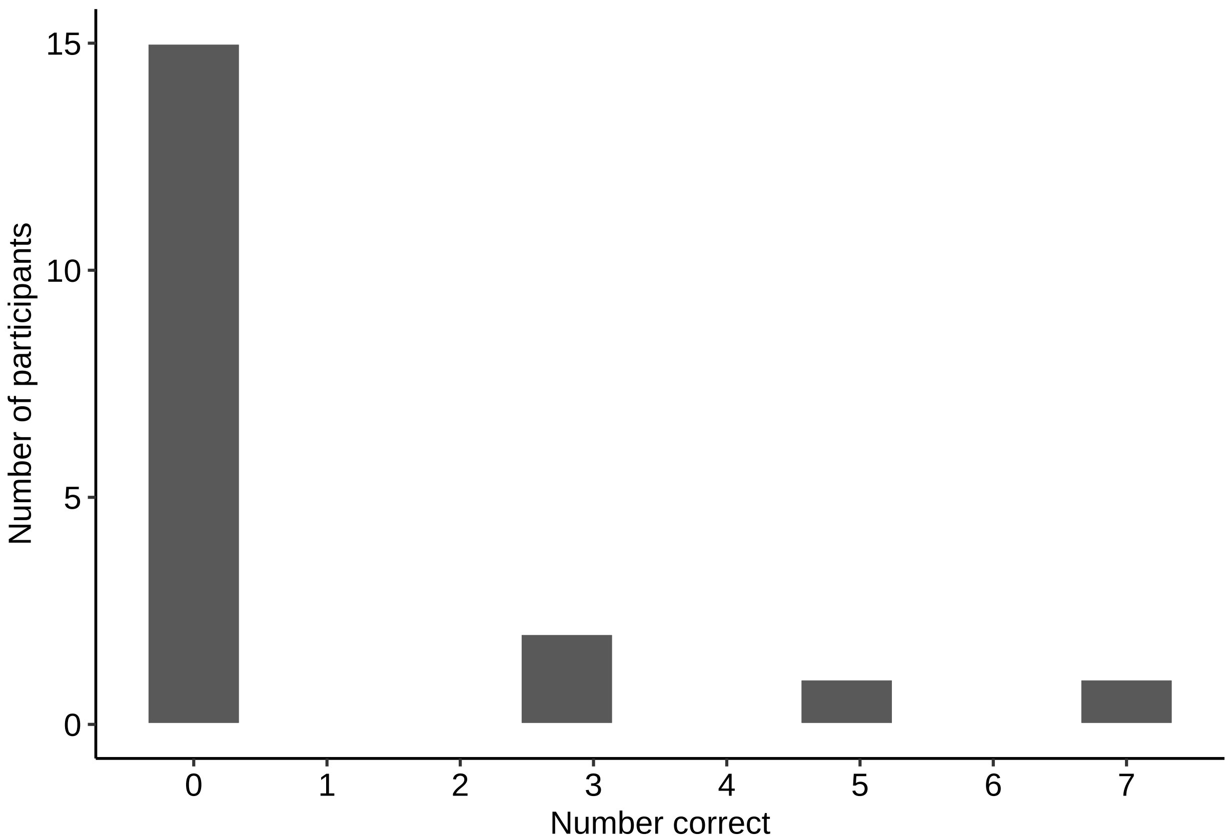


Figure 10: Distribution of delayed memory scores

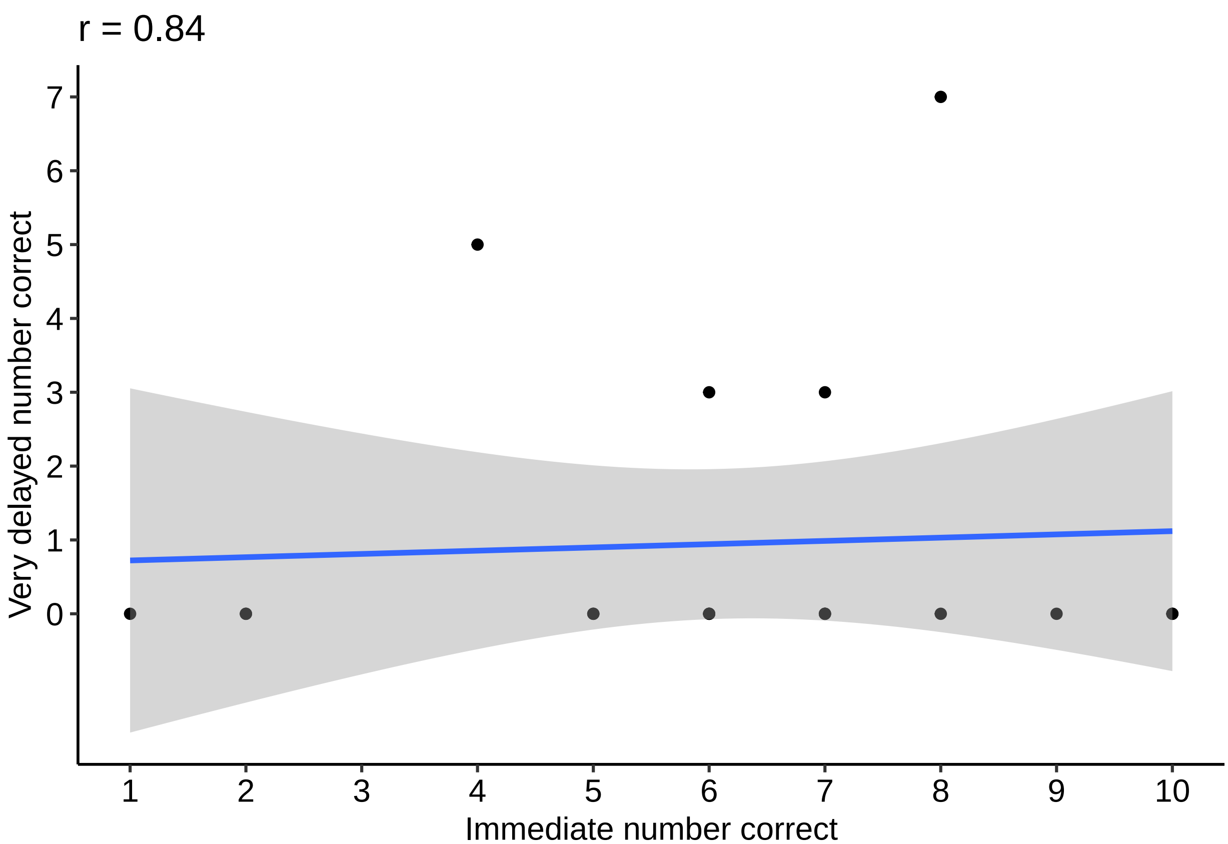


Figure 11: Relationship between immediate and delayed recall

```
##                     memory delayed_memory very_delayed_memory
## memory                1.00           0.84                0.05
## delayed_memory        0.84           1.00                0.05
## very_delayed_memory   0.05           0.05                1.00
## Sample Size
##                     memory delayed_memory very_delayed_memory
## memory                  35             35                  19
## delayed_memory          35             35                  19
## very_delayed_memory     19             19                  19
## Probability values (Entries above the diagonal are adjusted for multiple tests.)
##                     memory delayed_memory very_delayed_memory
## memory                0.00           0.00                   1
## delayed_memory        0.00           0.00                   1
## very_delayed_memory   0.82           0.83                   0
##
##  To see confidence intervals of the correlations, print with the short=FALSE option
```

1.4.3 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.)"),
    labels = c("Computer", "Mobile", "Tablet")
  ))
```

1.4.4 Long-form dataset

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 four formats: `[trait]_[abcd]` (for example, `talkative_a`), `[trait]_[abcd]_2` (for example, `talkative_a_2`), `[trait]_[abcd]_3` (e.g., `talkative_a_3`), or `[trait]_[abcd]_3i` (e.g., `talkative_a_3i`). We search for these items using regular expressions.

```
item_responses = str_subset(
  names(data),
  "^([[:alpha:]])+_[abcd](_2)?(_3)?(i)?$"
)
```

Similarly, we'll need to know how long it took participants to respond to these items. These variable names have one of four formats listed above followed by the string `page_submit`. We search for these items using regular expressions.

```
item_timing = str_subset(names(data), "t_([[:alpha:]])+_[abcd](_2)?(_3)?(i)?_page_submit$")
```

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

```
items_df = data %>%
  select(proid, condition, time2,
         memory, delayed_memory, very_delayed_memory,
         devicetype,
         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` or `_3`. We also identify whether it ends with `i`, indicating the item in block 3 started with "I". 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$")) %>%
  #identify block
  mutate(
    block = case_when(
      str_detect(item, "_2") ~ "2",
      str_detect(item, "_3") ~ "3",
      TRUE ~ "1"),
    item = str_remove(item, "_[23]")) %>%
  # identify presence of "I"
  mutate(i = case_when(
    str_detect(item, "i$") ~ "Present",
    TRUE ~ "Absent"),
    item = str_remove(item, "i$")) %>%
  separate(item, into = c("item", "format")) %>%
  spread(variable, value)
```

1.4.4.1 Remove 'human' and 'asleep' 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")
```

1.4.4.2 Label formatting conditions 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",
```

1.4.4.3 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()
```

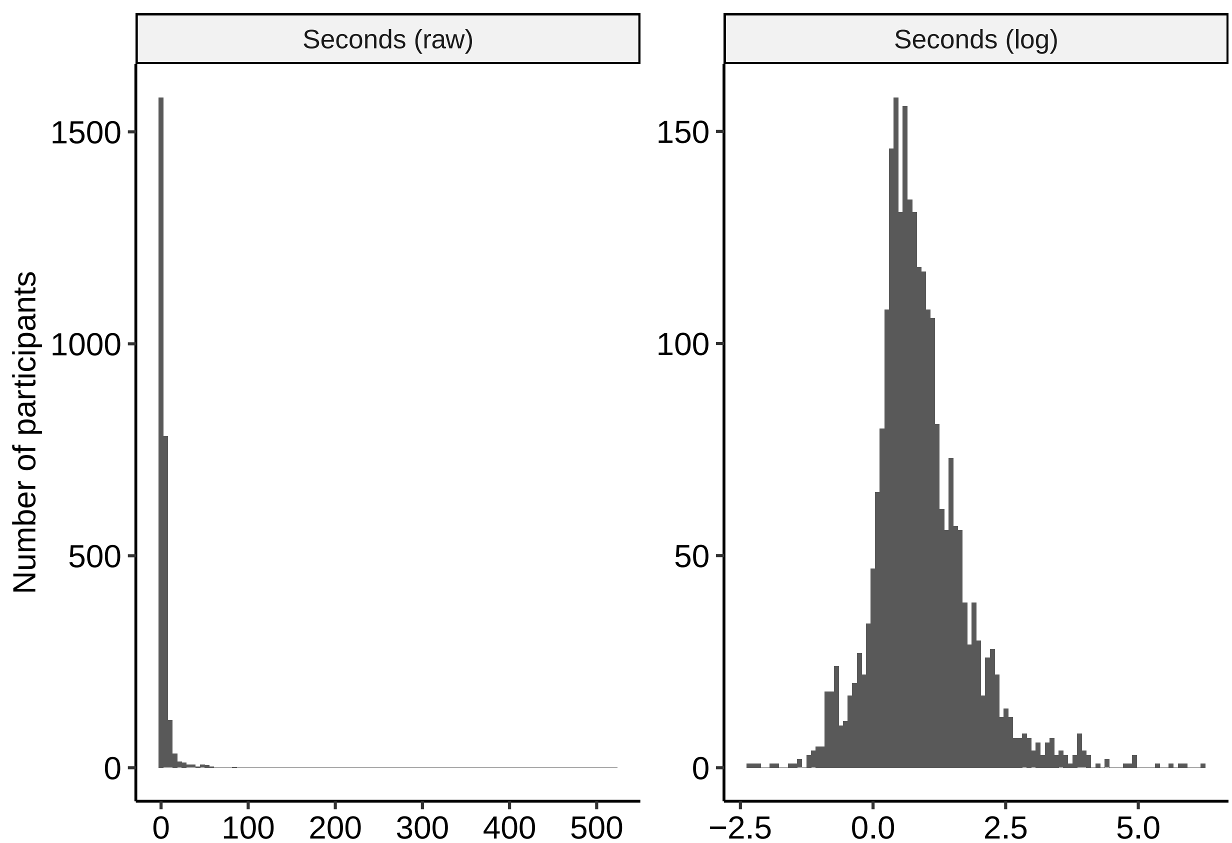


Figure 12: Distribution of seconds, raw and transformed.

1.4.4.4 New id numbers

We replace the Prolific ID numbers with randomly generated numbers. This allows us to share the data without the risk of identifying participants or breaking their confidentiality.

```
set.seed(202108)
original_id = unique(data$proid)
```

```r
new_id = stri_rand_strings(n = length(original_id), length = 10)

for(i in 1:length(original_id)){
  data$proid[data$proid == original_id[i]] = new_id[i]
  items_df$proid[data$proid == original_id[i]] = new_id[i]
}
```

2 Descriptives

Table 2: Descriptives of responses to Block 1

format	mean	sd	median	N_responses	N_participants
Adjective Only	4.70	1.30	5	319	11
Am Adjective	4.48	1.34	5	261	9
Tend to be Adjective	4.61	1.24	5	232	8
Am someone who tends to be Adjective	4.71	1.35	5	203	7

2.1 Block 1 personality

```
items_df %>%
  filter(block == "1") %>%
  group_by(format) %>%
  summarise(
    mean = mean(response),
    sd = sd(response),
    median = median(response),
    N_responses = n(),
    N_participants = length(unique(proid))
  ) %>%
  kable(booktabs = T, digits = c(0,2,2,0,0,0),
        caption = "Descriptives of responses to Block 1") %>%
  kable_styling()
```

```
items_df %>%
  filter(block == "1") %>%
  group_by(item, format) %>%
  summarise(
    mean = mean(response),
    sd = sd(response)
  ) %>%
  mutate(value = paste0(
    printnum(mean), " (", printnum(sd), ")"
  )) %>%
  select(-mean, -sd) %>%
  spread(format, value) %>%
  kable(booktabs = T) %>%
  kable_styling()
```

2.2 Block 2 personality

```
items_df %>%
  filter(block == "2") %>%
  group_by(format) %>%
  summarise(
    mean = mean(response),
    sd = sd(response),
```

item	Adjective Only	Am Adjective	Tend to be Adjective	Am someone who tends to be Adjective
active	5.45 (1.21)	5.00 (0.87)	4.75 (1.04)	4.86 (1.77)
adventurous	4.82 (0.60)	5.00 (0.87)	4.50 (0.93)	4.57 (1.40)
calm	5.45 (0.52)	4.67 (1.00)	4.75 (1.28)	5.00 (1.15)
careless	4.82 (1.40)	3.33 (1.94)	4.12 (1.25)	5.29 (1.11)
caring	5.36 (0.67)	4.78 (0.83)	5.00 (1.07)	5.43 (0.79)
cautious	4.91 (1.14)	4.67 (1.50)	5.00 (0.53)	4.43 (1.72)
creative	5.09 (0.94)	4.67 (1.22)	5.00 (0.93)	5.43 (0.79)
curious	4.64 (1.12)	4.89 (0.78)	4.62 (1.30)	4.57 (0.98)
friendly	5.55 (0.69)	5.33 (0.71)	5.25 (0.71)	5.29 (0.76)
hardworking	5.45 (0.69)	5.44 (0.73)	5.38 (1.06)	5.43 (0.53)
helpful	5.09 (0.94)	5.22 (0.83)	5.12 (0.64)	5.29 (1.11)
imaginative	5.00 (1.00)	5.22 (0.97)	5.25 (0.89)	5.57 (0.53)
impulsive	3.00 (1.18)	3.11 (1.17)	3.00 (1.51)	3.71 (1.80)
intelligent	5.09 (1.22)	4.78 (1.64)	5.25 (0.89)	5.43 (0.53)
lively	4.91 (0.83)	4.56 (0.73)	5.00 (1.20)	5.00 (1.41)
moody	3.91 (1.30)	2.89 (1.36)	3.38 (1.19)	3.86 (1.57)
nervous	4.09 (1.70)	3.56 (1.59)	3.88 (0.99)	4.00 (1.91)
organized	5.27 (0.79)	4.67 (1.12)	4.75 (1.28)	4.86 (1.07)
outgoing	4.91 (0.83)	4.89 (1.05)	4.38 (1.19)	4.71 (1.60)
reckless	4.18 (1.72)	4.11 (1.83)	4.38 (1.69)	5.14 (0.90)
responsible	5.64 (0.50)	5.22 (0.83)	5.50 (0.76)	5.43 (0.53)
softhearted	5.18 (0.98)	4.78 (0.97)	4.88 (0.99)	4.71 (1.80)
sophisticated	3.73 (1.27)	4.11 (1.05)	4.12 (1.73)	3.86 (1.07)
sympathetic	5.27 (1.01)	4.56 (1.01)	5.00 (0.76)	4.57 (1.27)
talkative	2.73 (1.01)	4.22 (1.72)	4.12 (1.46)	2.71 (1.38)
thorough	4.36 (1.29)	4.56 (1.13)	4.38 (1.06)	4.57 (0.98)
thrifty	3.64 (1.12)	3.89 (1.27)	4.75 (1.28)	3.43 (0.79)
warm	5.00 (1.48)	4.78 (0.83)	5.12 (0.99)	5.14 (0.69)
worrying	3.64 (1.43)	2.89 (1.69)	3.12 (1.25)	4.29 (1.80)

Table 3: Descriptives of responses to Block 2

format	mean	sd	median	N_responses	N_participants
Adjective Only	4.66	1.14	5	258	35
Am Adjective	4.57	1.44	5	260	35
Tend to be Adjective	4.64	1.34	5	250	35
Am someone who tends to be Adjective	4.59	1.33	5	247	35

```
    median = median(response),
    N_responses = n(),
    N_participants = length(unique(proid))
  ) %>%
  kable(booktabs = T, digits = c(0,2,2,0,0,0),
        caption = "Descriptives of responses to Block 2") %>%
  kable_styling()
```

```
items_df %>%
  filter(block == "2") %>%
  group_by(item, format) %>%
  summarise(
    mean = mean(response),
    sd = sd(response)
  ) %>%
  mutate(value = paste0(
    printnum(mean), " (", printnum(sd), ")"
  )) %>%
  select(-mean, -sd) %>%
  spread(format, value) %>%
  kable(booktabs = T) %>%
  kable_styling()
```

item	Adjective Only	Am Adjective	Tend to be Adjective	Am someone who tends to be Adjective
active	4.56 (1.67)	5.22 (1.30)	4.89 (1.27)	5.12 (0.35)
adventurous	5.00 (1.05)	4.86 (0.90)	5.22 (0.67)	4.11 (1.36)
calm	5.10 (0.57)	5.25 (0.71)	4.88 (0.83)	4.89 (1.05)
careless	4.50 (1.43)	3.62 (2.13)	4.62 (1.51)	4.89 (1.05)
caring	5.11 (0.60)	5.50 (0.76)	4.75 (0.46)	5.20 (0.79)
cautious	4.67 (0.71)	4.80 (0.92)	5.00 (0.53)	4.62 (1.30)
creative	5.00 (0.58)	5.50 (0.71)	5.10 (0.57)	4.88 (1.36)
curious	4.25 (1.58)	4.30 (1.64)	4.67 (1.32)	5.38 (0.74)
friendly	5.25 (0.71)	5.00 (1.31)	5.33 (0.71)	5.00 (0.67)
hardworking	5.44 (0.73)	4.60 (1.17)	6.00 (0.00)	5.12 (0.35)
helpful	5.33 (0.50)	5.40 (0.70)	5.43 (0.53)	5.56 (0.73)
imaginative	4.70 (0.67)	5.22 (0.44)	5.12 (1.13)	5.25 (0.71)
impulsive	3.89 (1.62)	3.22 (1.79)	3.75 (1.67)	2.89 (0.93)
intelligent	5.12 (0.64)	5.00 (1.32)	5.44 (0.53)	5.33 (0.71)
lively	4.75 (1.04)	5.00 (1.00)	5.00 (1.00)	4.00 (1.94)
moody	3.73 (0.79)	4.00 (1.94)	3.00 (1.91)	4.00 (1.69)
nervous	4.12 (1.46)	4.10 (1.52)	3.11 (1.83)	3.75 (1.67)
organized	4.62 (1.51)	4.70 (1.42)	5.50 (0.76)	4.89 (0.60)
outgoing	5.12 (0.99)	4.40 (1.35)	4.12 (1.55)	4.78 (0.97)
reckless	4.56 (1.33)	4.89 (1.54)	3.80 (1.87)	4.86 (1.46)
responsible	5.30 (0.48)	5.22 (0.83)	5.00 (0.58)	4.78 (1.72)
softhearted	4.90 (0.88)	4.78 (1.30)	5.62 (0.52)	5.00 (1.07)
sophisticated	3.89 (1.17)	4.14 (1.68)	3.40 (1.35)	4.33 (0.87)
sympathetic	5.00 (0.87)	4.78 (0.83)	4.22 (0.97)	5.00 (0.76)
talkative	3.67 (1.00)	2.22 (1.20)	3.88 (1.81)	3.11 (1.83)
thorough	4.62 (1.19)	4.89 (1.05)	4.30 (1.25)	4.88 (0.83)
thrifty	4.00 (1.41)	3.78 (1.20)	4.33 (1.12)	3.56 (1.51)
warm	5.33 (0.50)	5.00 (0.94)	5.11 (0.60)	5.29 (0.76)
worrying	3.56 (1.24)	2.43 (1.62)	4.45 (1.69)	2.88 (1.64)

3 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).

3.1 Effect of format (Block 1 data)

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),
```

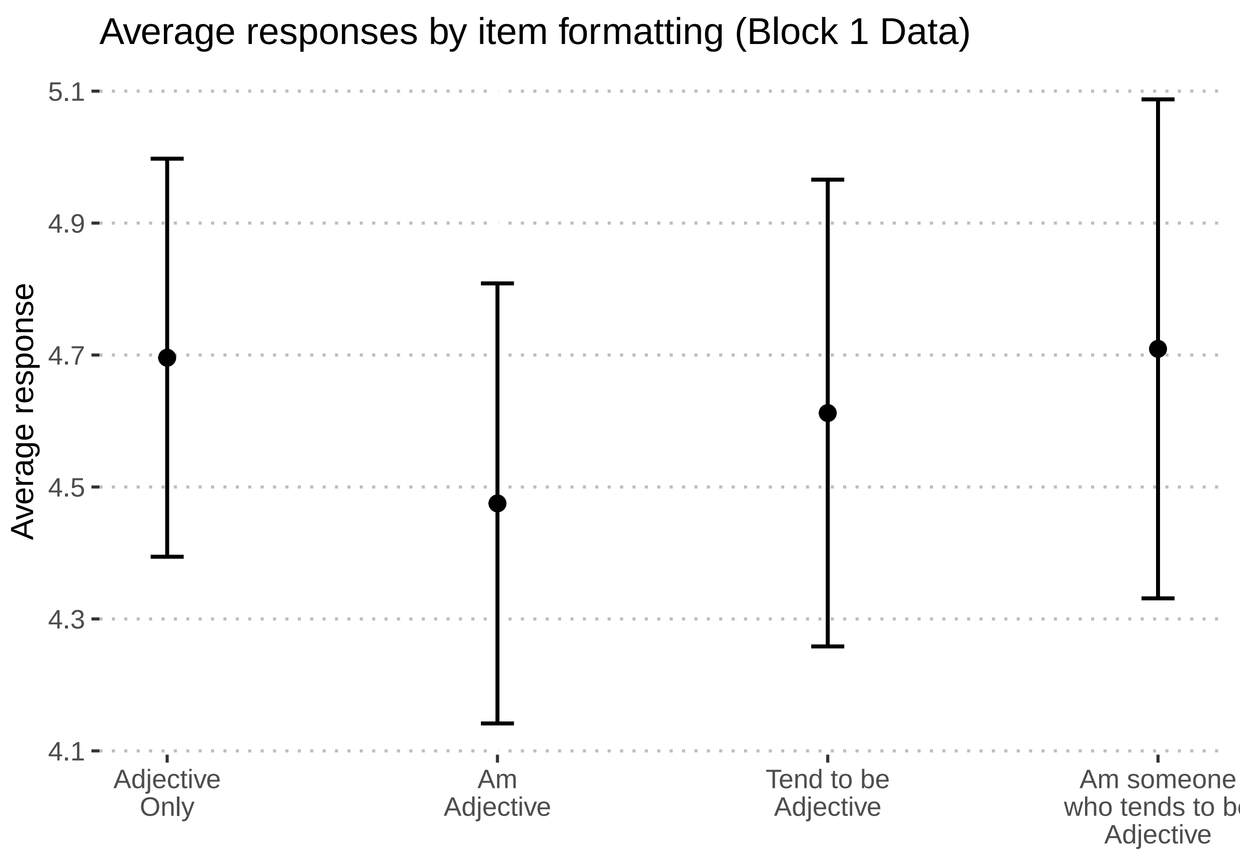


Figure 13: Predicted response on personality items by condition, using only Block 1 data.

```
                                    "\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()
```

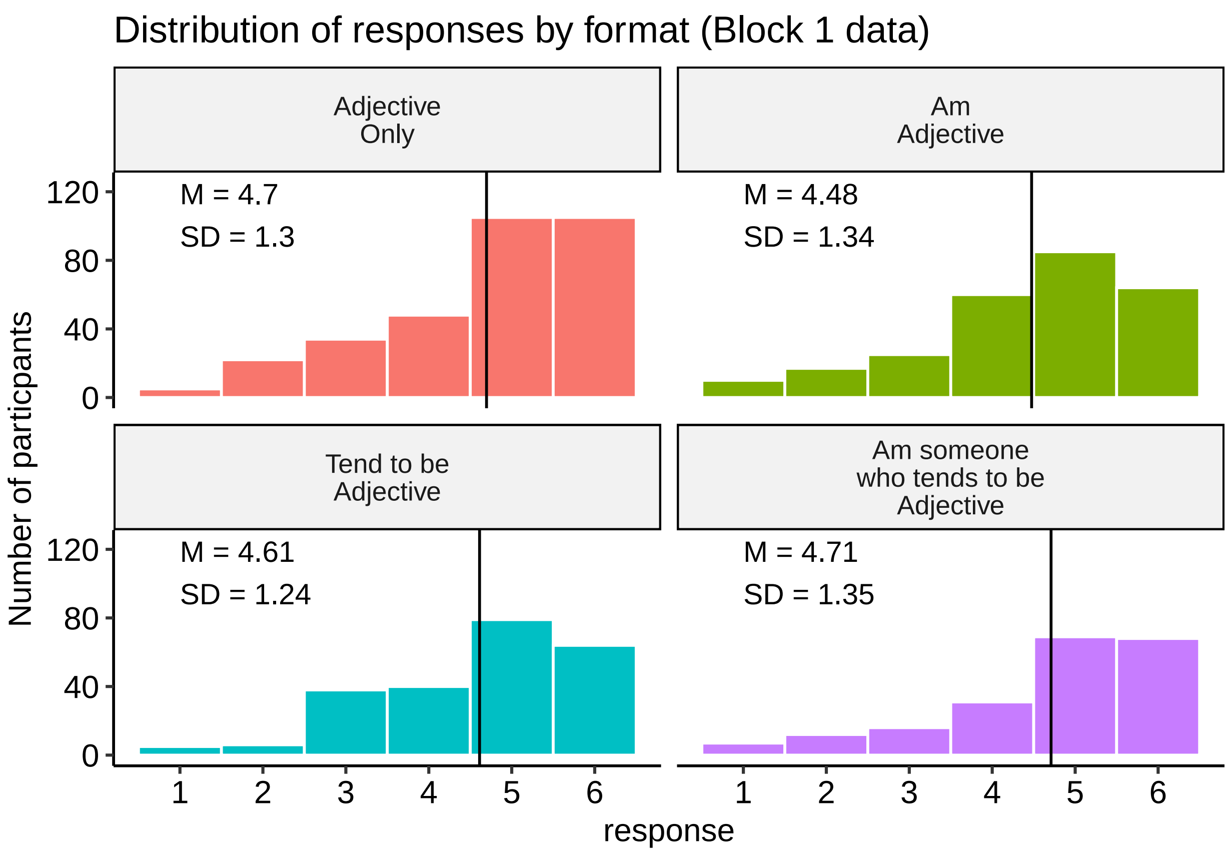


Figure 14: Distribution of responses by category, block 1 data only

3.1.1 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 %>%
```

```
  ungroup() %>%
  mutate(tidy = map(aov, broom::tidy)) %>%
  select(item, tidy) %>%
  unnest(cols = c(tidy)) %>%
  filter(term == "format") %>%
  mutate(reverse = case_when(
    item %in% reverse ~ "Y",
    TRUE ~ "N"
  )) %>%
  mutate(p.adj = p.adjust(p.value, method = "holm"))

summary_by_item_b1 %>%
  arrange(reverse, item) %>%
  select(item, reverse, sumsq, meansq, df, statistic, p.value, p.adj) %>%
  kable(digits = c(0,0,2,2,2,2,3,3), booktabs = T,
         col.names = c("Item", "Reverse\nScored?", "SS", "MS", "df", "F", "raw", "adj"),
        caption = "Format effects on response by item (block 1 data only)") %>%
  kable_styling()
```

3.1.2 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!

3.1.3 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()
```

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

Item	Reverse Scored?	SS	MS	df	F	raw	adj
active	N	2.80	0.93	3	0.61	0.611	1
adventurous	N	1.34	0.45	3	0.50	0.682	1
calm	N	3.77	1.26	3	1.29	0.295	1
caring	N	2.47	0.82	3	1.17	0.337	1
cautious	N	1.55	0.52	3	0.32	0.814	1
creative	N	2.35	0.78	3	0.79	0.507	1
curious	N	0.52	0.17	3	0.15	0.927	1
friendly	N	0.52	0.17	3	0.34	0.796	1
hardworking	N	0.03	0.01	3	0.02	0.997	1
helpful	N	0.20	0.07	3	0.08	0.968	1
imaginative	N	1.40	0.47	3	0.58	0.630	1
intelligent	N	1.86	0.62	3	0.44	0.725	1
lively	N	1.15	0.38	3	0.36	0.782	1
organized	N	2.20	0.73	3	0.66	0.583	1
outgoing	N	1.58	0.53	3	0.40	0.755	1
responsible	N	0.87	0.29	3	0.65	0.588	1
softhearted	N	1.25	0.42	3	0.30	0.828	1
sophisticated	N	1.08	0.36	3	0.21	0.887	1
sympathetic	N	3.42	1.14	3	1.10	0.363	1
talkative	N	18.53	6.18	3	3.19	0.037	1
thorough	N	0.33	0.11	3	0.08	0.968	1
thrifty	N	8.09	2.70	3	2.06	0.126	1
warm	N	0.71	0.24	3	0.20	0.897	1
careless	Y	18.23	6.08	3	2.77	0.058	1
impulsive	Y	2.65	0.88	3	0.45	0.716	1
moody	Y	6.21	2.07	3	1.14	0.350	1
nervous	Y	1.54	0.51	3	0.20	0.893	1
reckless	Y	5.14	1.71	3	0.65	0.587	1
worrying	Y	8.95	2.98	3	1.25	0.307	1

Table 5: 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 - 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 - Am someone who tends to be Adjective	1.51	0.70	31	2.15	0.192
Tend to be Adjective - Am someone who tends to be Adjective	1.41	0.72	31	1.96	0.192

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

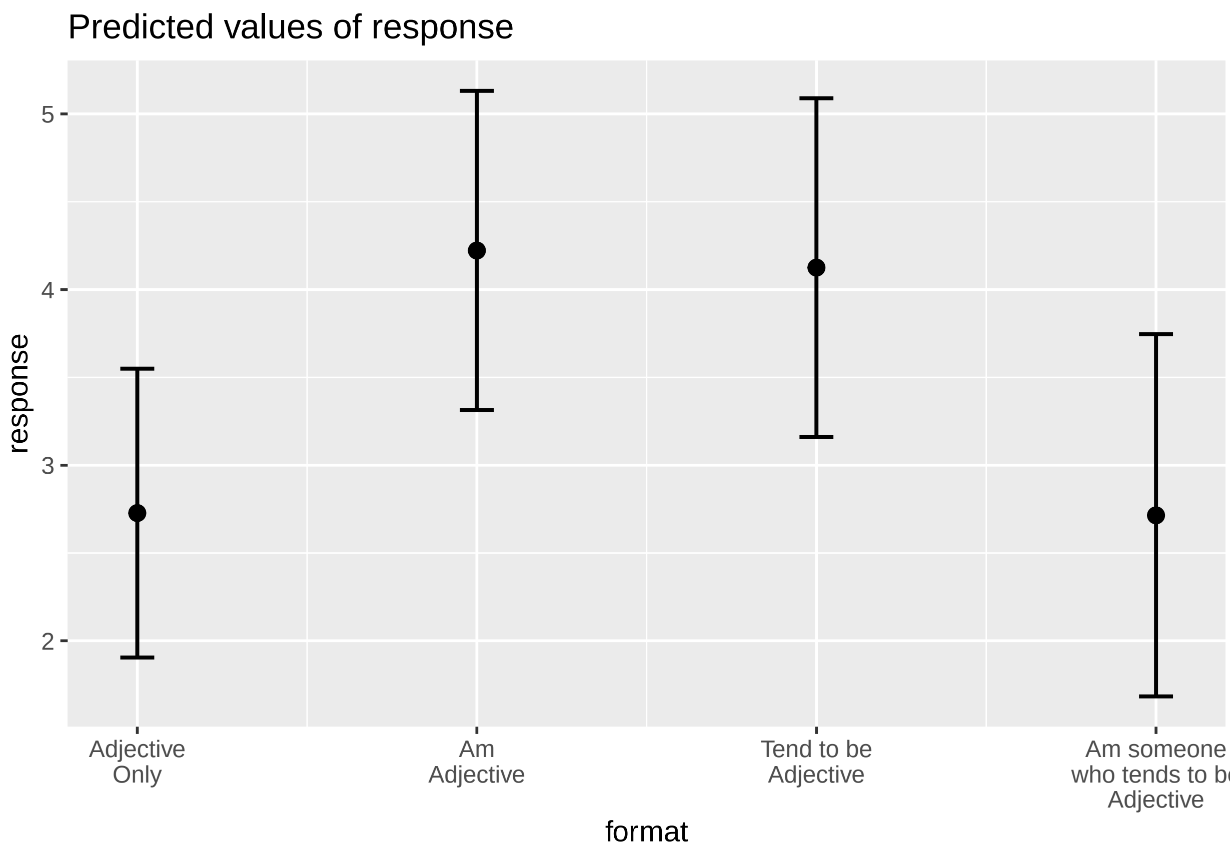


Figure 15: Average response to "talkative" by format (block 1 data only)

3.2 Effect of format (Block 1 and 2)

We used a multilevel model, nesting response within participant to account for dependence. Our primary predictor was format. Here, we use data from blocks 1 and 2.

```
items_12 = items_df %>% filter(block %in% c("1","2"))
```

```
mod.format_b2 = lmer(response~format + (1|proid),
                  data = items_12)
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()
```

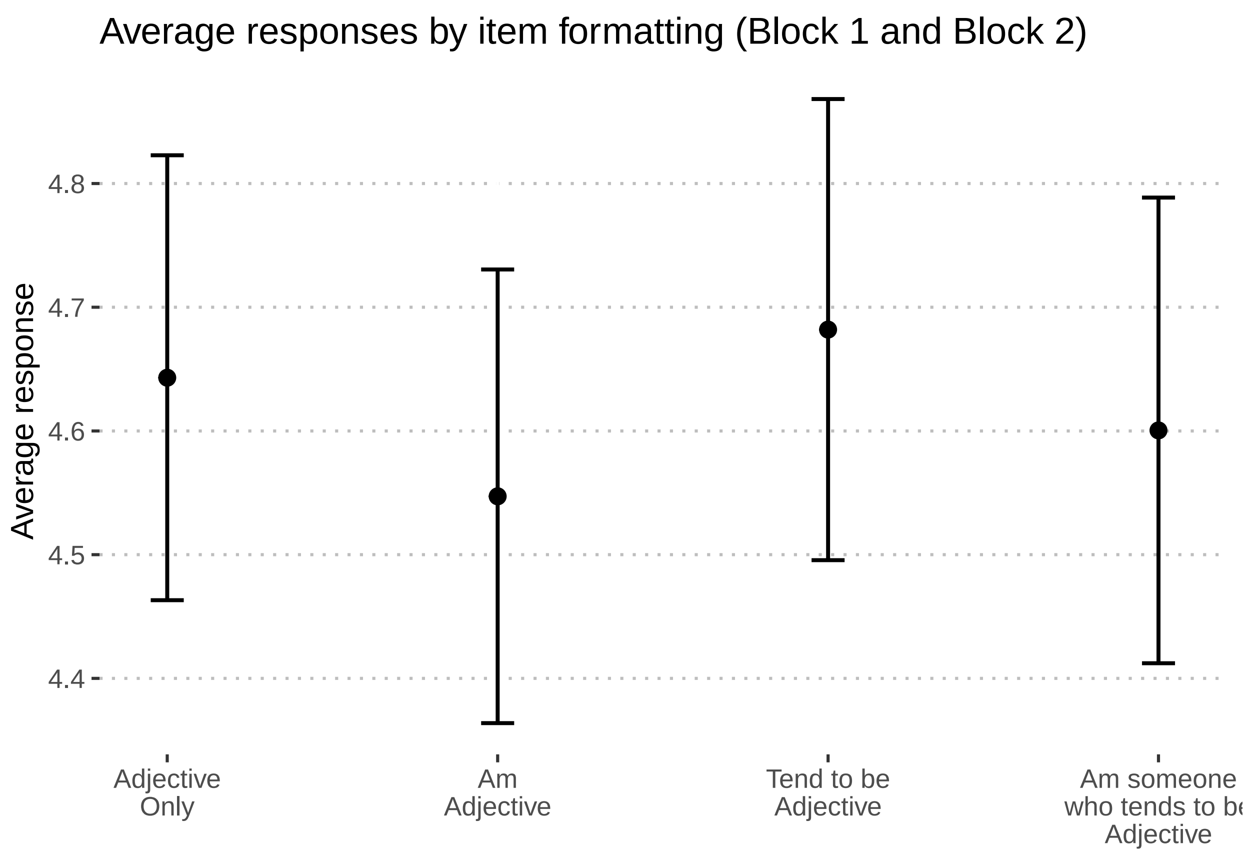


Figure 16: Predicted response on personality items by condition, using only Block 1 data.

```
means_by_group = items_12 %>%
  group_by(format) %>%
  summarise(m = mean(response),
            s = sd(response))

items_12 %>%
  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()
```

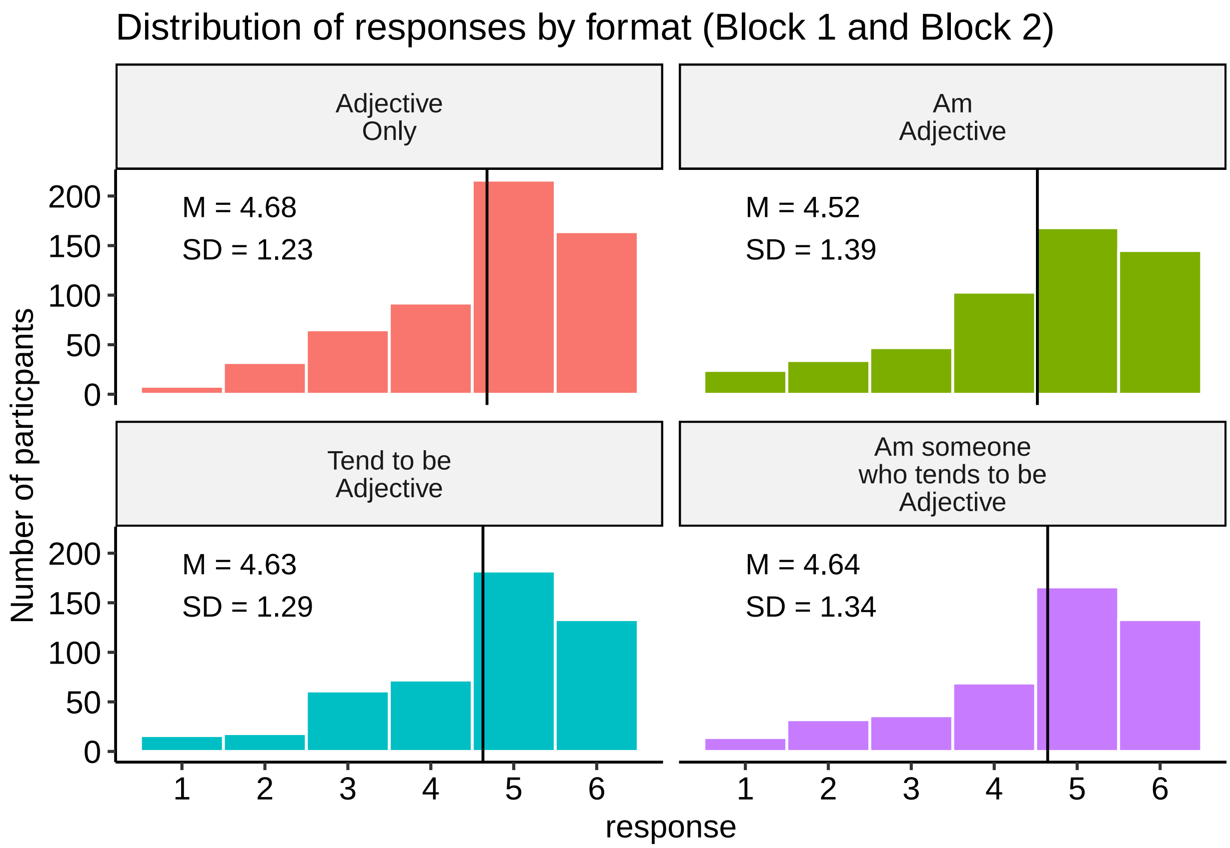


Figure 17: Distribution of responses by category, block 1 and block 2

3.2.1 One model for each adjective

We can also repeat this analysis separately for each trait. We use the `anova` function to estimate the variability due to format and print the corresponding F-test.

```
mod_by_item_b2 = items_12 %>%
  group_by(item) %>%
  nest() %>%
  mutate(mod = map(data, ~lmer(response~format + (1|proid), data = .))) %>%
  mutate(aov = map(mod, anova))
```

To present these results, we use the `tidy` function to summarise the findings and extract just the F-test associated with the format variable. We calculate adjusted p-values using a Holm correction. We also create a column that indicates whether the item was reverse-scored; we use this to sort the table, in case a pattern emerges.

```
summary_by_item_b2 = mod_by_item_b2 %>%
  ungroup() %>%
  mutate(tidy = map(aov, broom::tidy)) %>%
  select(item, tidy) %>%
  unnest(cols = c(tidy)) %>%
  filter(term == "format") %>%
  mutate(reverse = case_when(
    item %in% reverse ~ "Y",
    TRUE ~ "N"
  )) %>%
  mutate(p.adj = p.adjust(p.value, method = "holm"))

summary_by_item_b2 %>%
  arrange(reverse, item) %>%
  select(item, reverse, sumsq, meansq, NumDF, DenDF, statistic, p.value, p.adj) %>%
  kable(digits = c(0,0,2,2,0,2,2,3,3),
        col.names = c("Item", "Reverse\nScored?", "SS", "MS", "df1", "df2", "F", "raw", "adj"),
        booktabs = T, caption = "Format effects on response by item (block 1 data only)") %>%
  kable_styling() %>%
    add_header_above(c(" " = 7, "p-value" = 2))
```

3.2.2 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!

3.2.3 Careless

```
careless_model_b2 = items_12 %>%
  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,
        digits = c(0,2,2,2,2,3),
```

Table 6: Format effects on response by item (block 1 data only)

Item	Reverse Scored?	SS	MS	df1	df2	F	p-value raw	p-value adj
active	N	0.76	0.25	3	40.00	1.01	0.398	1.000
adventurous	N	1.51	0.50	3	48.65	1.09	0.361	1.000
calm	N	0.13	0.04	3	45.02	0.13	0.940	1.000
caring	N	1.16	0.39	3	52.66	1.28	0.291	1.000
cautious	N	2.30	0.77	3	54.97	0.98	0.407	1.000
creative	N	0.14	0.05	3	49.22	0.13	0.943	1.000
curious	N	2.35	0.78	3	52.00	1.05	0.377	1.000
friendly	N	0.94	0.31	3	46.85	1.45	0.239	1.000
hardworking	N	2.10	0.70	3	52.03	2.06	0.117	1.000
helpful	N	0.90	0.30	3	60.54	0.82	0.488	1.000
imaginative	N	1.03	0.34	3	55.07	0.86	0.465	1.000
intelligent	N	1.72	0.57	3	51.16	1.15	0.340	1.000
lively	N	1.08	0.36	3	42.77	0.89	0.456	1.000
organized	N	0.22	0.07	3	37.39	0.41	0.750	1.000
outgoing	N	0.71	0.24	3	39.34	0.83	0.484	1.000
responsible	N	1.51	0.50	3	56.50	0.82	0.487	1.000
softhearted	N	1.95	0.65	3	57.46	0.74	0.531	1.000
sophisticated	N	0.05	0.02	3	44.04	0.04	0.989	1.000
sympathetic	N	0.49	0.16	3	51.48	0.51	0.676	1.000
talkative	N	6.36	2.12	3	43.86	1.95	0.135	1.000
thorough	N	2.23	0.74	3	40.16	2.72	0.057	1.000
thrifty	N	6.39	2.13	3	51.69	3.30	0.027	0.794
warm	N	0.10	0.03	3	52.22	0.08	0.968	1.000
careless	Y	8.29	2.76	3	53.32	2.84	0.047	1.000
impulsive	Y	1.38	0.46	3	41.41	0.85	0.473	1.000
moody	Y	1.21	0.40	3	42.77	0.69	0.566	1.000
nervous	Y	1.35	0.45	3	46.22	0.45	0.716	1.000
reckless	Y	1.14	0.38	3	48.60	0.36	0.782	1.000
worrying	Y	1.72	0.57	3	41.67	0.63	0.602	1.000

Table 7: 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 - 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 - Am someone who tends to be Adjective	-1.26	0.45	51.43	-2.82	0.041
Tend to be Adjective - Am someone who tends to be Adjective	-0.58	0.45	49.48	-1.31	0.591

```
      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"))
```

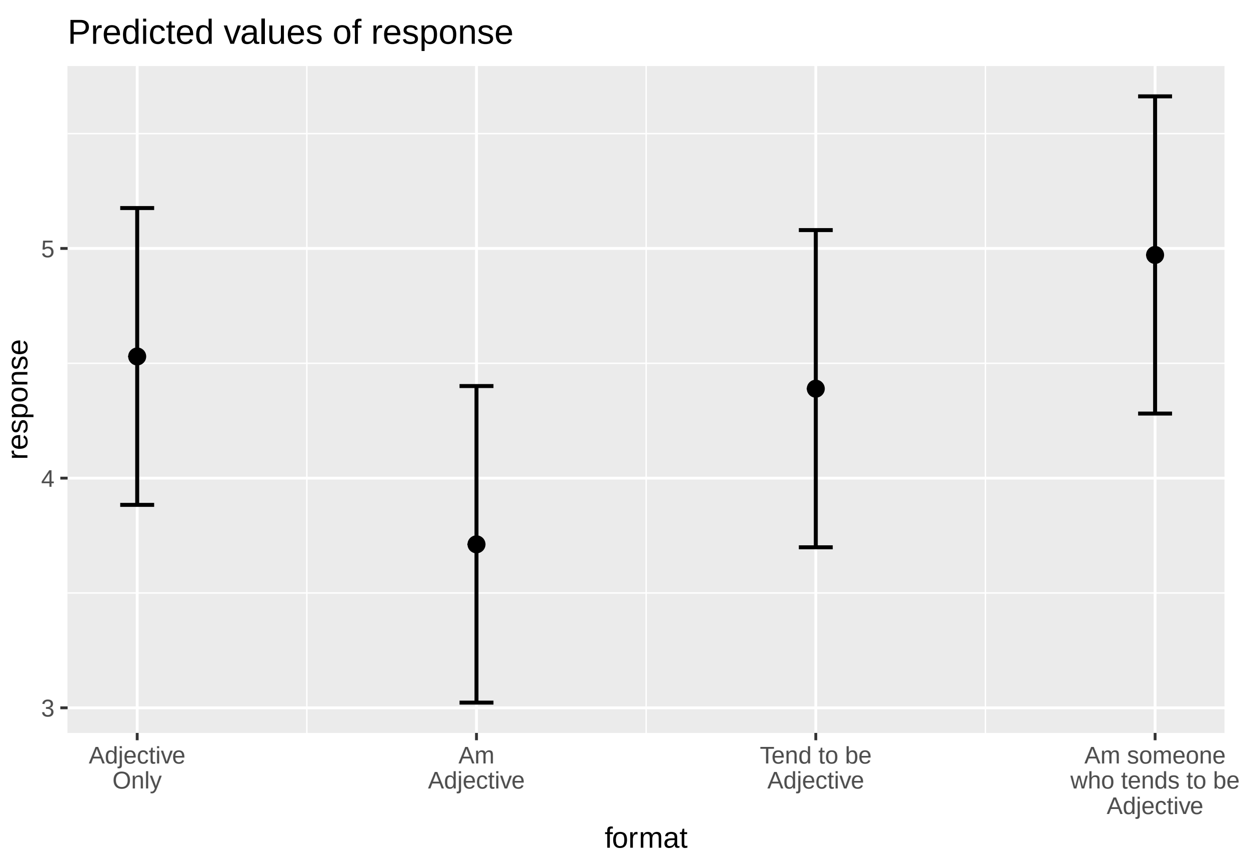


Figure 18: Average response to “careless” by format (Block 1 and Block 2)

Table 8: 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 - 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 - Am someone who tends to be Adjective	-0.08	0.37	54.54	-0.21	1.000
Tend to be Adjective - Am someone who tends to be Adjective	0.83	0.33	43.22	2.49	0.084

3.2.4 Thrifty

```
thrifty_model_b2 = items_12 %>%
  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"))
```

3.3 Account for memory effects (Blocks 1 and 2)

```
mod.format_mem = lmer(response~format*delayed_memory + (1|proid),
                      data = items_12)
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_12
```

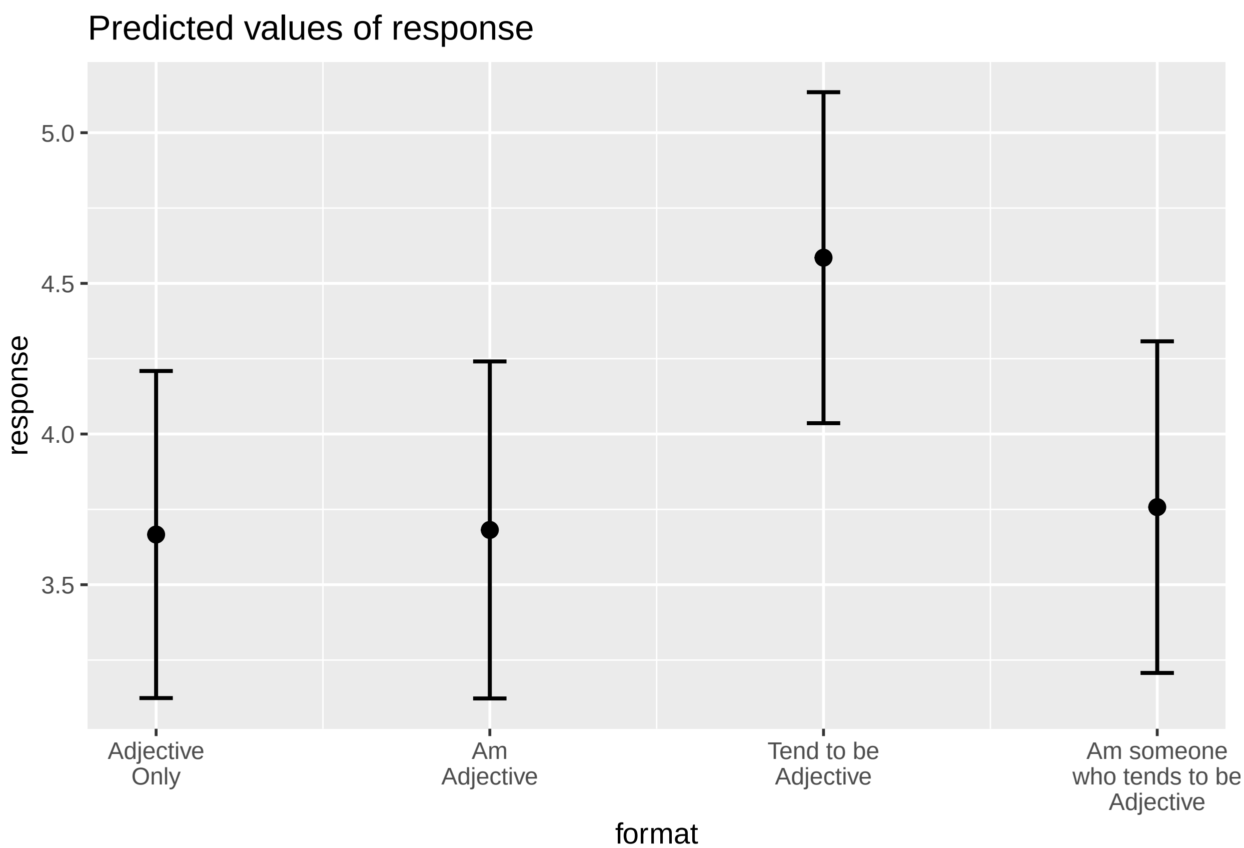


Figure 19: Average response to “thrifty” by format (Block 1 and Block 2)

```
##
## 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
## formatAm 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
## formatAm 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
## formatAm 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
## formatAm someone\nwho tends to be\nAdjective:delayed_memory  3.082e-02
##                                                               df t value
## (Intercept)                                            6.417e+01  24.900
## formatAm\nAdjective                                    1.990e+03  -2.028
## formatTend to be\nAdjective                            1.946e+03  -0.612
## formatAm someone\nwho tends to be\nAdjective           1.999e+03  -1.083
## delayed_memory                                         6.577e+01   0.197
## formatAm\nAdjective:delayed_memory                     1.971e+03   1.683
## formatTend to be\nAdjective:delayed_memory             1.927e+03   0.964
## formatAm someone\nwho tends to be\nAdjective:delayed_memory  2.006e+03   0.990
##                                                       Pr(>|t|)
## (Intercept)                                             <2e-16 ***
## formatAm\nAdjective                                     0.0427 *
## formatTend to be\nAdjective                             0.5408
## formatAm someone\nwho tends to be\nAdjective            0.2787
## delayed_memory                                          0.8447
## formatAm\nAdjective:delayed_memory                      0.0925 .
## formatTend to be\nAdjective:delayed_memory              0.3350
## formatAm someone\nwho tends to be\nAdjective:delayed_memory   0.3224
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
```

```
##              (Intr) frmtAA frTtbA frAswttbA dlyd_m frAA:_ fTtbA:
## frmtAmAdjct -0.449
## frmtTndtbAd -0.442  0.475
## frmtAswttbA -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
## frAswttbA:_  0.363 -0.402 -0.410 -0.872    -0.422  0.487  0.462
```

```r
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()
```

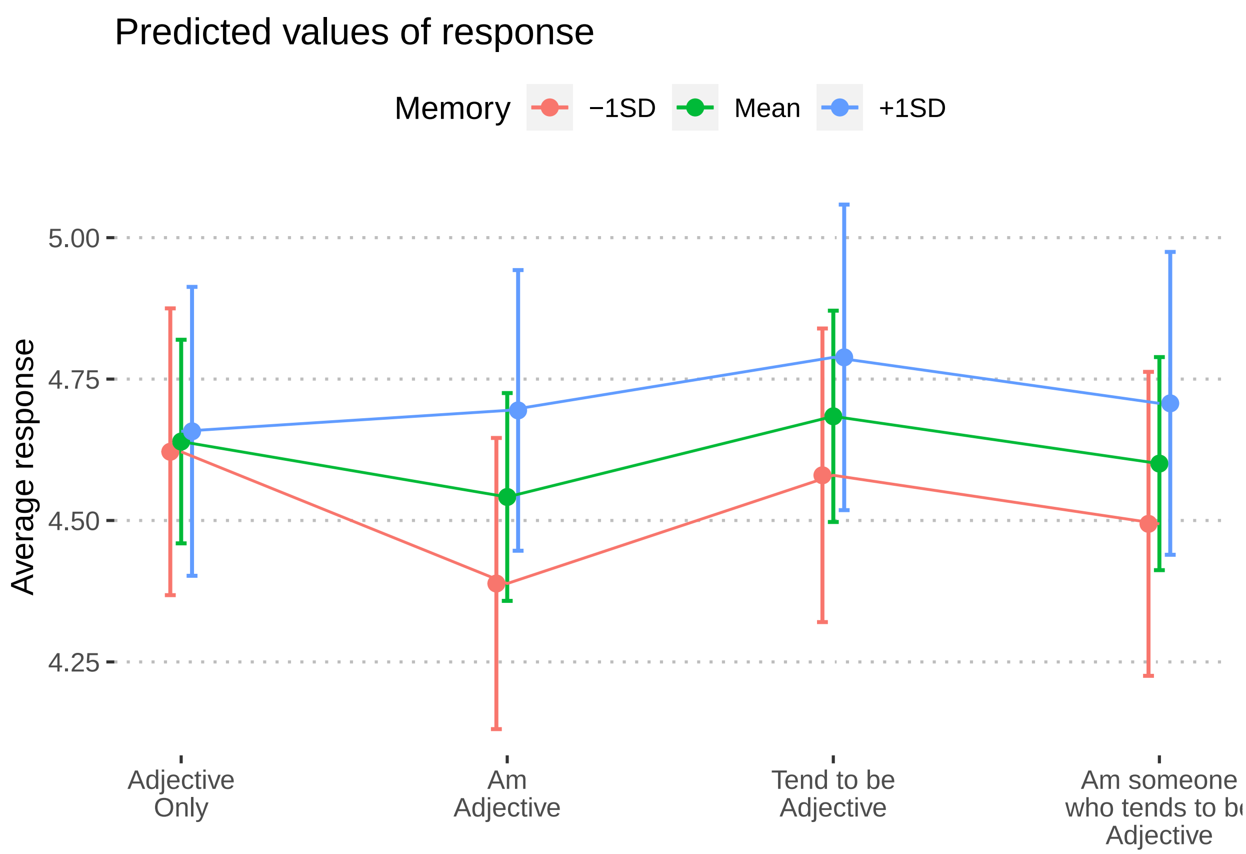


Figure 20: Predicted response on personality items by condition after controlling for delayed_memory.

3.3.1 One model for each adjective

```
mod_by_item_mem = items_12 %>%
  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 %>%
  ungroup() %>%
  mutate(tidy = map(aov, broom::tidy)) %>%
  select(item, tidy) %>%
  unnest(cols = c(tidy)) %>%
  filter(term == "format:delayed_memory") %>%
  mutate(reverse = case_when(
    item %in% reverse ~ "Y",
    TRUE ~ "N"
  )) %>%
  mutate(p.adj = p.adjust(p.value, method = "holm"))

summary_by_item_mem %>%
  arrange(reverse, item) %>%
  select(item, reverse, sumsq, meansq, df, statistic, p.value, p.adj) %>%
  kable(digits = c(0,0,2,2,2,2,3,3),
         col.names = c("Item", "Reverse\nScored?", "SS", "MS", "df", "F", "raw", "adj"),
        booktabs = T) %>%
  kable_styling() %>%
  add_header_above(c(" " = 6, "p-value" = 2))
```

3.3.2 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)
```

3.4 Inclusion of "I" (Block 1 and Block 3)

We used a multilevel model, nesting response within participant to account for dependence. Our primary predictors are format and also the presence of the word "I". Here, we use data from blocks 1 and 3.

```
items_13 = items_df %>%
  filter(block %in% c("1","3")) %>%
  filter(condition != "A") %>%
  filter(time2 == "yes")
```

						p-value	
Item	Reverse Scored?	SS	MS	df	F	raw	adj
active	N	0.29	0.10	3	0.06	0.980	1
adventurous	N	1.17	0.39	3	0.39	0.759	1
calm	N	0.15	0.05	3	0.06	0.980	1
caring	N	0.16	0.05	3	0.08	0.968	1
cautious	N	0.42	0.14	3	0.12	0.948	1
creative	N	1.12	0.37	3	0.44	0.727	1
curious	N	1.28	0.43	3	0.27	0.843	1
friendly	N	1.83	0.61	3	1.01	0.394	1
hardworking	N	0.74	0.25	3	0.37	0.771	1
helpful	N	1.31	0.44	3	0.73	0.536	1
imaginative	N	0.65	0.22	3	0.35	0.790	1
intelligent	N	2.94	0.98	3	0.95	0.421	1
lively	N	1.21	0.40	3	0.28	0.840	1
organized	N	5.06	1.69	3	1.53	0.216	1
outgoing	N	1.72	0.57	3	0.40	0.753	1
responsible	N	2.53	0.84	3	1.11	0.353	1
softhearted	N	0.46	0.15	3	0.13	0.943	1
sophisticated	N	4.21	1.40	3	0.89	0.451	1
sympathetic	N	1.11	0.37	3	0.40	0.754	1
talkative	N	8.53	2.84	3	1.23	0.306	1
thorough	N	3.87	1.29	3	1.13	0.344	1
thrifty	N	0.61	0.20	3	0.13	0.940	1
warm	N	1.65	0.55	3	0.69	0.560	1
careless	Y	3.74	1.25	3	0.55	0.653	1
impulsive	Y	7.37	2.46	3	1.13	0.343	1
moody	Y	10.15	3.38	3	1.76	0.165	1
nervous	Y	4.40	1.47	3	0.57	0.638	1
reckless	Y	7.65	2.55	3	1.01	0.394	1
worrying	Y	1.77	0.59	3	0.22	0.880	1

```
mod.format_b3 = lmer(response~format*i + (1|proid),
                    data = items_13)
anova(mod.format_b3)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##          Sum Sq Mean Sq NumDF  DenDF F value Pr(>F)
## format   3.2586 1.62932     2  12.41  1.0262 0.3869
## i        2.6021 2.60209     1 796.86  1.6389 0.2008
## format:i 1.5784 0.78918     2 797.04  0.4971 0.6085
```

```
plot_b3 = plot_model(mod.format_b3, type = "pred", terms = c("format", "i"))
plot_b3 +
  geom_line() +
  labs(x = NULL,
       y = "Average response",
       title = "Average responses by item formatting (Block 1 and Block 2)") +
  theme_pubclean()
```

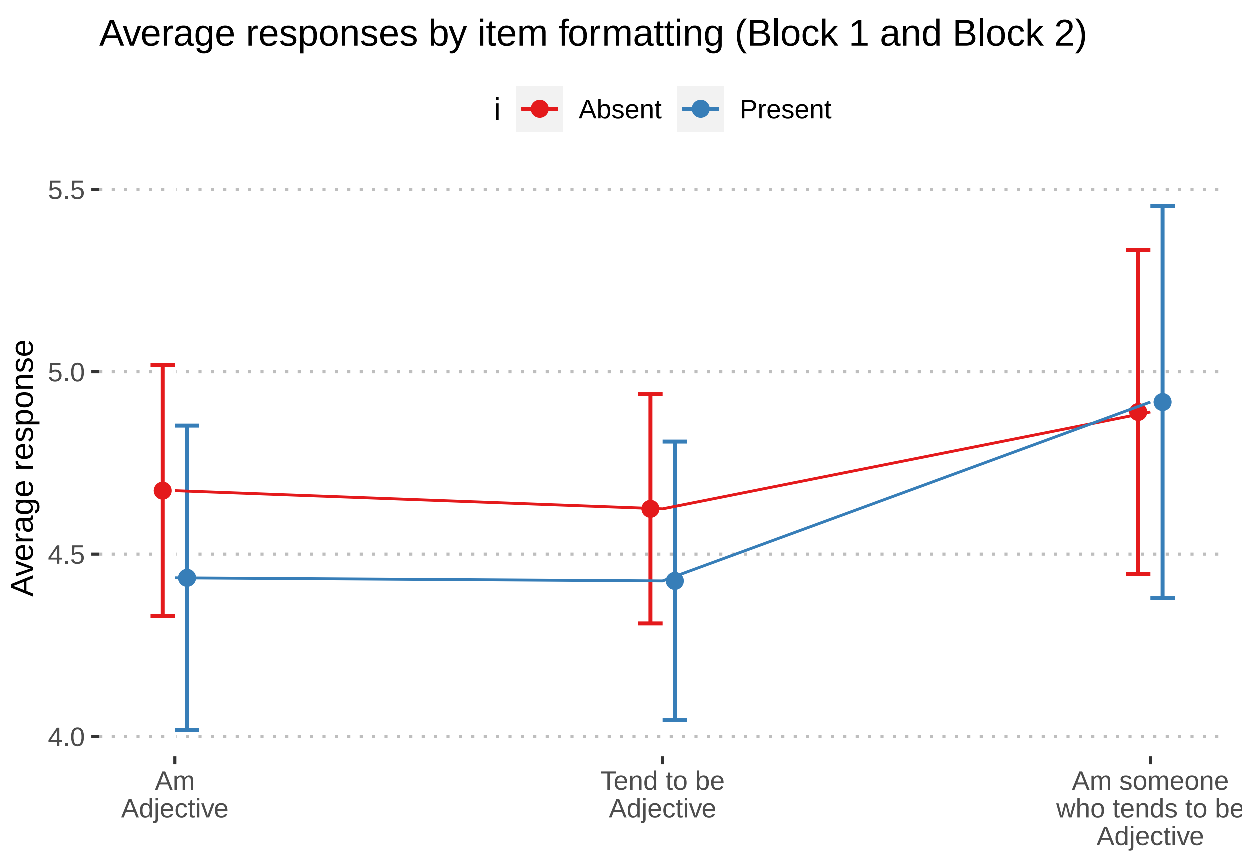


Figure 21: Predicted response on personality items by condition, using only Block 1 data.

```
means_by_group = items_13 %>%
  group_by(format, i) %>%
  summarise(m = mean(response),
```

```
            s = sd(response))

items_13 %>%
  ggplot(aes(x = response, fill = i)) +
  geom_histogram(bins = 6, color = "white") +
  geom_vline(aes(xintercept = m), data = means_by_group) +
  geom_text(aes(x = 1,
                y = 100,
                label = paste("M =", round(m,2),
                              "\nSD =", round(s,2))),
            data = means_by_group,
            hjust =0,
            vjust = 1) +
  facet_grid(i~format, scales = "free") +
  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()
```

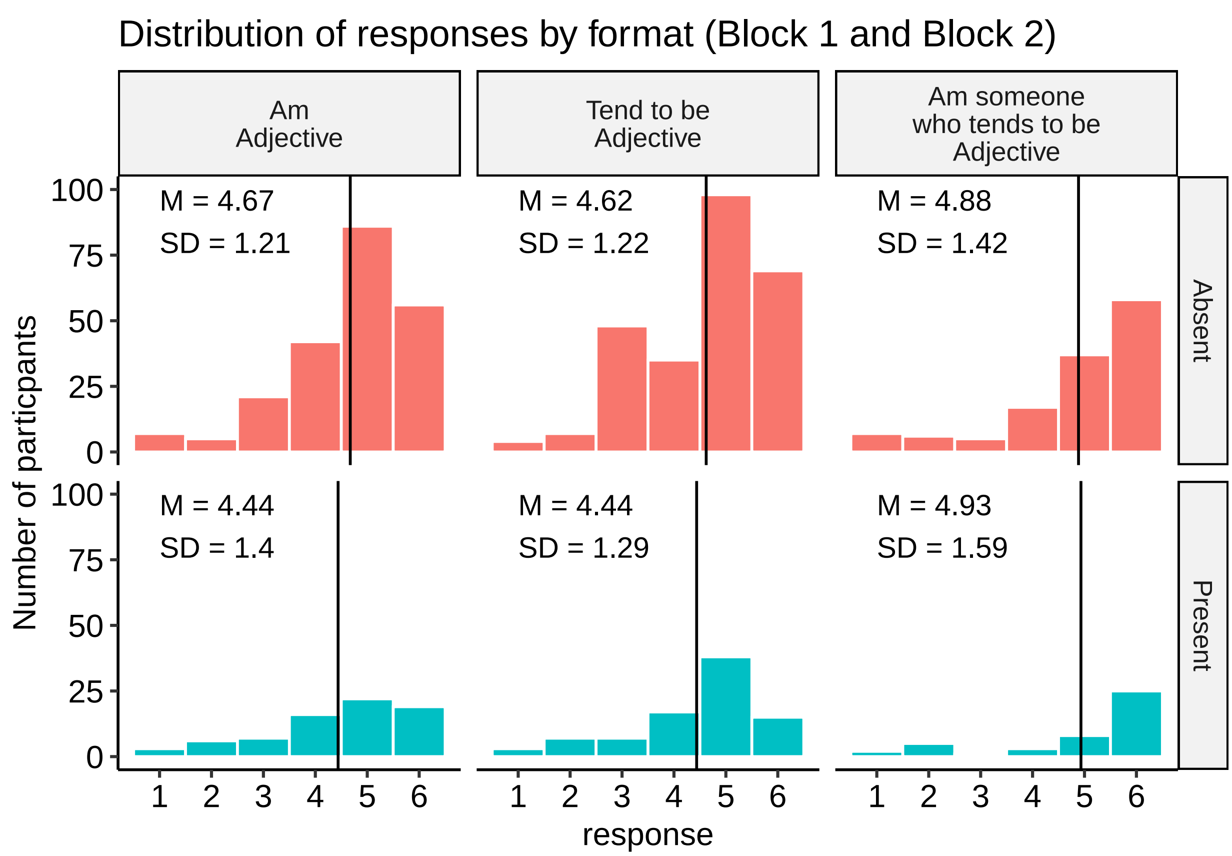


Figure 22: Distribution of responses by category, block 1 and block 2

3.4.1 One model for each adjective

```
mod_by_item_i = items_13 %>%
  group_by(item) %>%
  nest() %>%
  mutate(mod = map(data, ~lmer(response~format*i + (1|proid), data = .))) %>%
  mutate(aov = map(mod, anova))

summary_by_item_i = mod_by_item_i %>%
  ungroup() %>%
  mutate(tidy = map(aov, broom::tidy)) %>%
  select(item, tidy) %>%
  unnest(cols = c(tidy)) %>%
  filter(term == "format:i") %>%
  mutate(reverse = case_when(
    item %in% reverse ~ "Y",
    TRUE ~ "N"
  )) %>%
  mutate(p.adj = p.adjust(p.value, method = "holm"))

summary_by_item_i %>%
  arrange(reverse, item) %>%
  select(item, reverse, sumsq, meansq, NumDF, DenDF, statistic, p.value, p.adj) %>%
  kable(digits = c(0,0,2,2,0,2,2,3,3),
        col.names = c("Item", "Reverse\nScored?", "SS", "MS", "df1", "df2", "F", "raw", "adj"),
        booktabs = T, caption = "Interaction of format and \"I\" (block 1 and 3 data)") %>%
  kable_styling() %>%
    add_header_above(c(" " = 7, "p-value" = 2))
```

3.4.2 Pairwise t-tests for significant ANOVAs

Here we identify the specific items with significant differences.

```
sig_item_i = summary_by_item_i %>%
  filter(p.value < .05)

sig_item_i = sig_item_i$item
sig_item_i
```

```
## [1] "curious"     "lively"      "responsible" "sympathetic"
```

3.4.3 Curious

```
curious_model_i = items_13 %>%
  filter(item == "curious") %>%
  lmer(response~format*i + (1|proid),
                               data = .)

curious_model_i %>%
```

Table 9: Interaction of format and "I" (block 1 and 3 data)

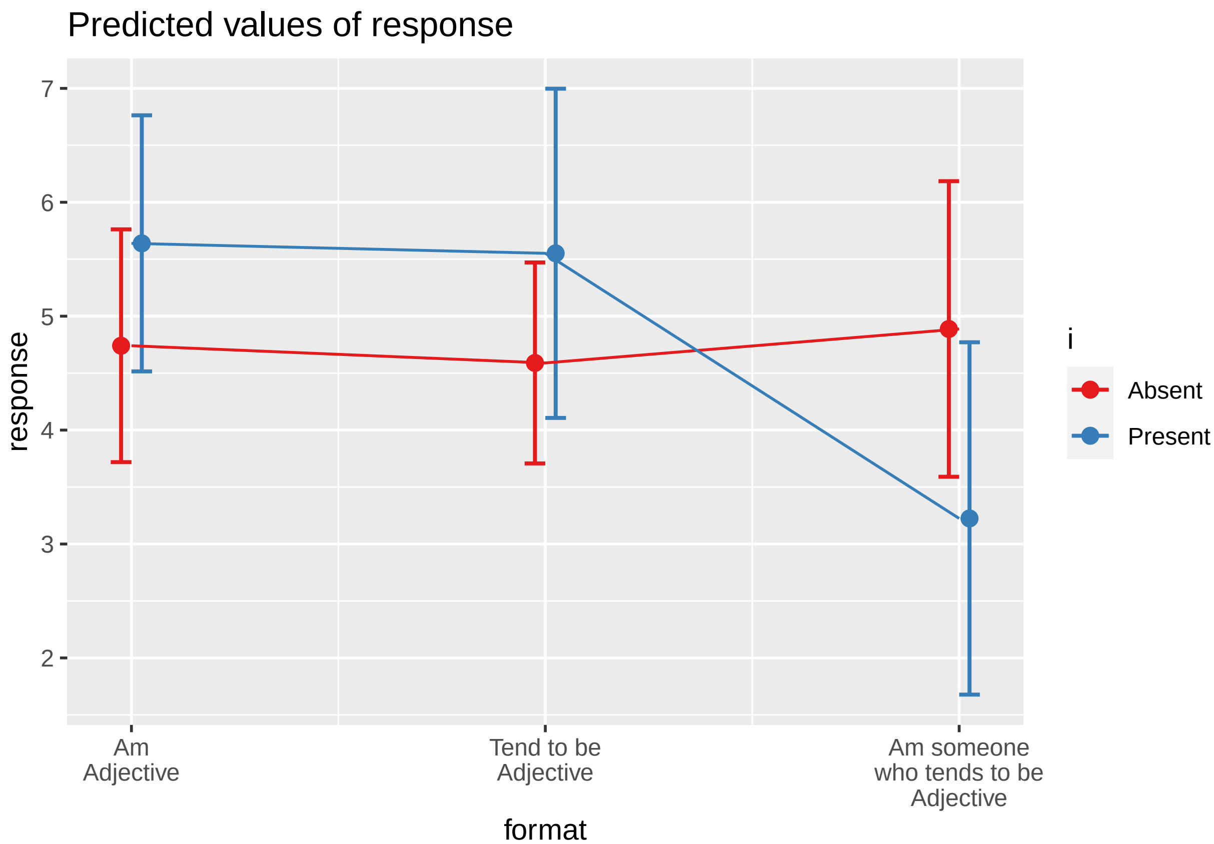
							p-value	
Item	Reverse Scored?	SS	MS	df1	df2	F	raw	adj
active	N	0.66	0.33	2	12.78	0.66	0.535	1.000
adventurous	N	1.80	0.90	2	17.26	2.06	0.158	1.000
calm	N	0.23	0.11	2	18.38	0.12	0.883	1.000
caring	N	0.15	0.08	2	22.00	0.24	0.786	1.000
cautious	N	1.52	0.76	2	12.76	1.42	0.277	1.000
creative	N	0.11	0.05	2	13.01	0.22	0.803	1.000
curious	N	5.47	2.73	2	13.80	4.54	0.031	0.827
friendly	N	0.19	0.10	2	22.00	0.30	0.741	1.000
hardworking	N	0.15	0.07	2	12.44	0.65	0.539	1.000
helpful	N	0.58	0.29	2	16.06	2.40	0.122	1.000
imaginative	N	0.69	0.35	2	14.68	1.96	0.177	1.000
intelligent	N	0.16	0.08	2	14.37	0.34	0.715	1.000
lively	N	2.91	1.45	2	13.39	6.92	0.009	0.250
organized	N	0.42	0.21	2	13.64	0.52	0.608	1.000
outgoing	N	0.26	0.26	1	13.85	0.90	0.358	1.000
responsible	N	1.83	0.92	2	15.53	3.67	0.050	1.000
softhearted	N	2.85	1.42	2	14.50	1.32	0.298	1.000
sophisticated	N	2.06	1.03	2	13.69	1.90	0.187	1.000
sympathetic	N	3.31	1.65	2	12.58	5.38	0.021	0.575
talkative	N	0.14	0.07	2	12.24	0.11	0.898	1.000
thorough	N	0.13	0.07	2	15.66	0.19	0.828	1.000
thrifty	N	0.93	0.47	2	14.23	0.58	0.570	1.000
warm	N	0.39	0.19	2	15.80	0.32	0.734	1.000
careless	Y	8.80	4.40	2	20.04	2.51	0.106	1.000
impulsive	Y	0.27	0.14	2	12.86	0.17	0.844	1.000
moody	Y	0.78	0.39	2	14.44	0.34	0.718	1.000
nervous	Y	1.38	0.69	2	18.91	0.37	0.693	1.000
reckless	Y	1.52	0.76	2	17.03	0.58	0.570	1.000
worrying	Y	1.72	0.86	2	15.50	0.78	0.474	1.000

Table 10: Interaction of format and "I"

Term	Estimate	SE	t	df	p
(Intercept)	4.74	0.52	9.09	14.77	0.000
Tend to be Adjective	-0.15	0.69	-0.22	13.62	0.830
Am someone who tends to be Adjective	0.15	0.84	0.17	14.25	0.864
I	0.90	0.53	1.68	12.58	0.117
Tend to be Adjective:I	0.06	0.89	0.07	14.24	0.944
Am someone who tends to be Adjective:I	-2.56	0.92	-2.80	13.09	0.015

```
  tidy() %>%
  mutate(term = str_replace(term, "\n", " "),
         term = str_replace(term, "format", ""),
         term = str_replace(term, "iPresent", "I")) %>%
  filter(is.na(group)) %>%
  select(-effect, -group) %>%
  kable(booktabs = T,
        digits = c(0,2,2,2,2,3),
        caption = "Interaction of format and \"I\"",
        col.names = c("Term", "Estimate", "SE",  "t", "df", "p")) %>%
  kable_styling()
```

```
plot_model(curious_model_i, type = "pred",
           terms = c("format", "i")) +
  geom_line()
```



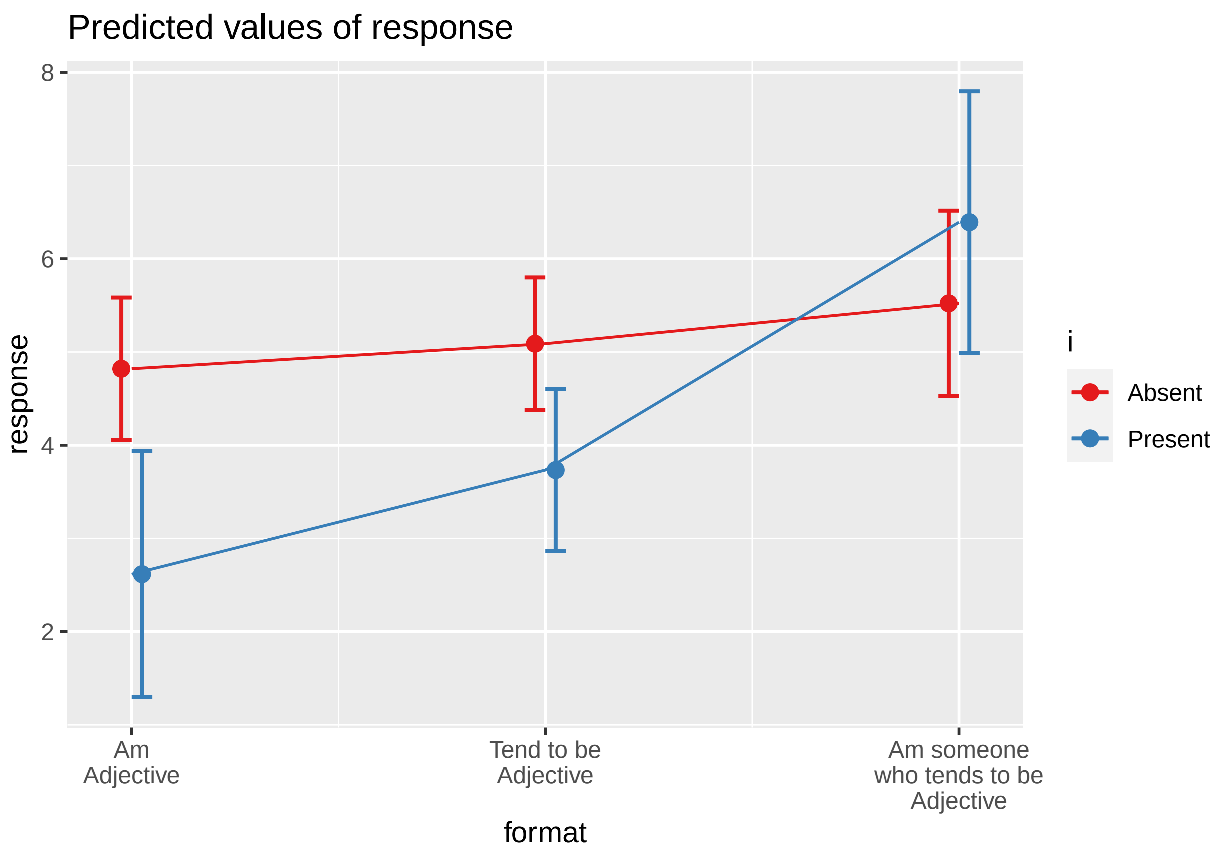
3.4.4 Lively

```
lively_model_i = items_13 %>%
  filter(item == "lively") %>%
  lmer(response~format*i + (1|proid),
                               data = .)

lively_model_i %>%
  tidy() %>%
  mutate(term = str_replace(term, "\n", " "),
         term = str_replace(term, "format", ""),
         term = str_replace(term, "iPresent", "I")) %>%
  filter(is.na(group)) %>%
  select(-effect, -group) %>%
  kable(booktabs = T,
        digits = c(0,2,2,2,2,3),
        caption = "Interaction of format and \"I\"",
        col.names = c("Term", "Estimate", "SE",  "t", "df", "p")) %>%
  kable_styling()
```

```
plot_model(lively_model_i, type = "pred",
           terms = c("format", "i")) +
  geom_line()
```

Table 11: Interaction of format and "I"

Term	Estimate	SE	t	df	p
(Intercept)	4.82	0.39	12.37	11.21	0.000
Tend to be Adjective	0.27	0.53	0.50	11.59	0.623
Am someone who tends to be Adjective	0.70	0.64	1.10	11.43	0.296
I	-2.20	0.61	-3.59	13.74	0.003
Tend to be Adjective:I	0.85	0.71	1.19	13.50	0.254
Am someone who tends to be Adjective:I	3.07	0.87	3.52	13.54	0.004



3.4.5 Responsible

```
responsible_model_i = items_13 %>%
  filter(item == "responsible") %>%
  lmer(response~format*i + (1|proid),
                              data = .)

responsible_model_i %>%
  tidy() %>%
  mutate(term = str_replace(term, "\n", " "),
         term = str_replace(term, "format", ""),
         term = str_replace(term, "iPresent", "I")) %>%
  filter(is.na(group)) %>%
```

Table 12: Interaction of format and "I"

Term	Estimate	SE	t	df	p
(Intercept)	5.63	0.27	21.21	16.29	0.000
Tend to be Adjective	-0.17	0.36	-0.49	15.64	0.634
Am someone who tends to be Adjective	-0.07	0.41	-0.18	13.70	0.859
I	0.16	0.34	0.48	13.25	0.638
Tend to be Adjective:I	-1.05	0.47	-2.23	13.91	0.043
Am someone who tends to be Adjective:I	0.47	0.70	0.68	16.66	0.509

```
  select(-effect, -group) %>%
  kable(booktabs = T,
        digits = c(0,2,2,2,2,3),
        caption = "Interaction of format and \"I\"",
        col.names = c("Term", "Estimate", "SE",  "t", "df", "p")) %>%
  kable_styling()
```

```
plot_model(responsible_model_i, type = "pred",
           terms = c("format", "i")) +
  geom_line()
```

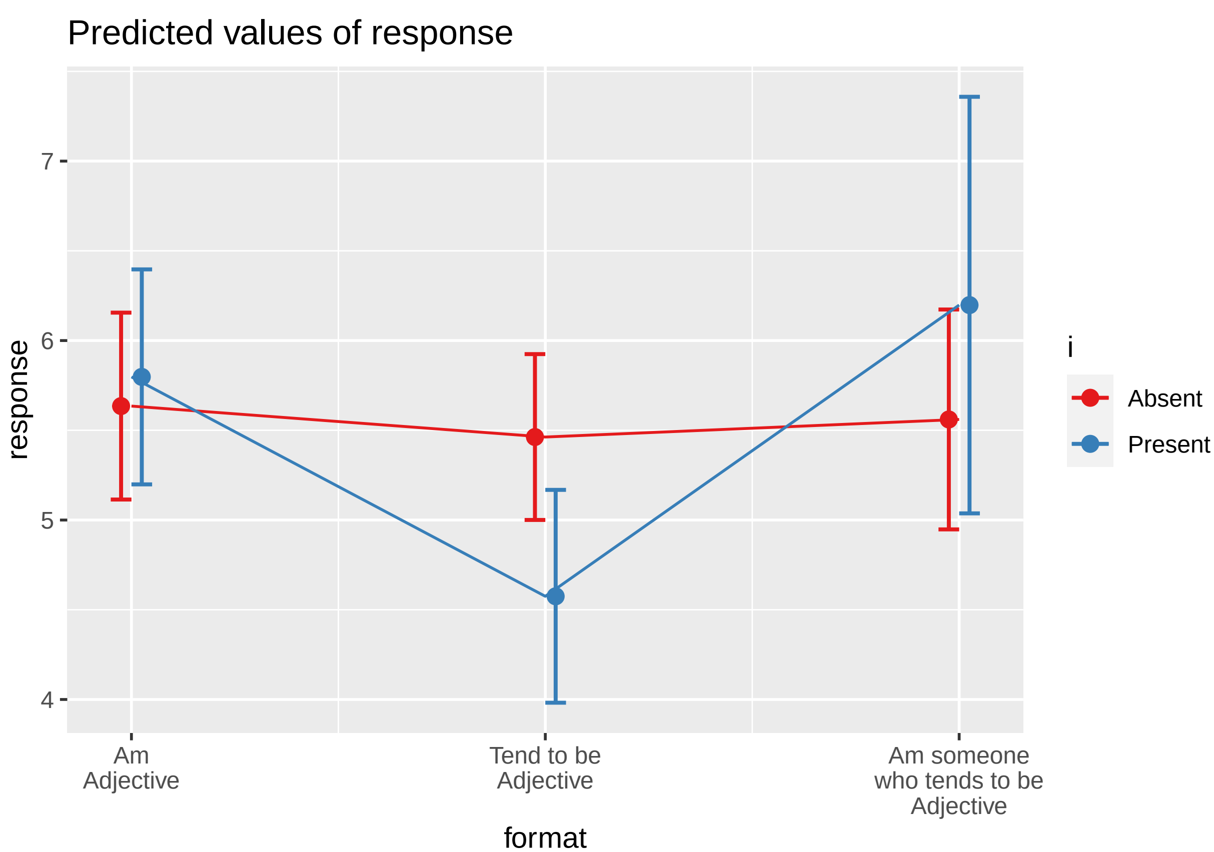


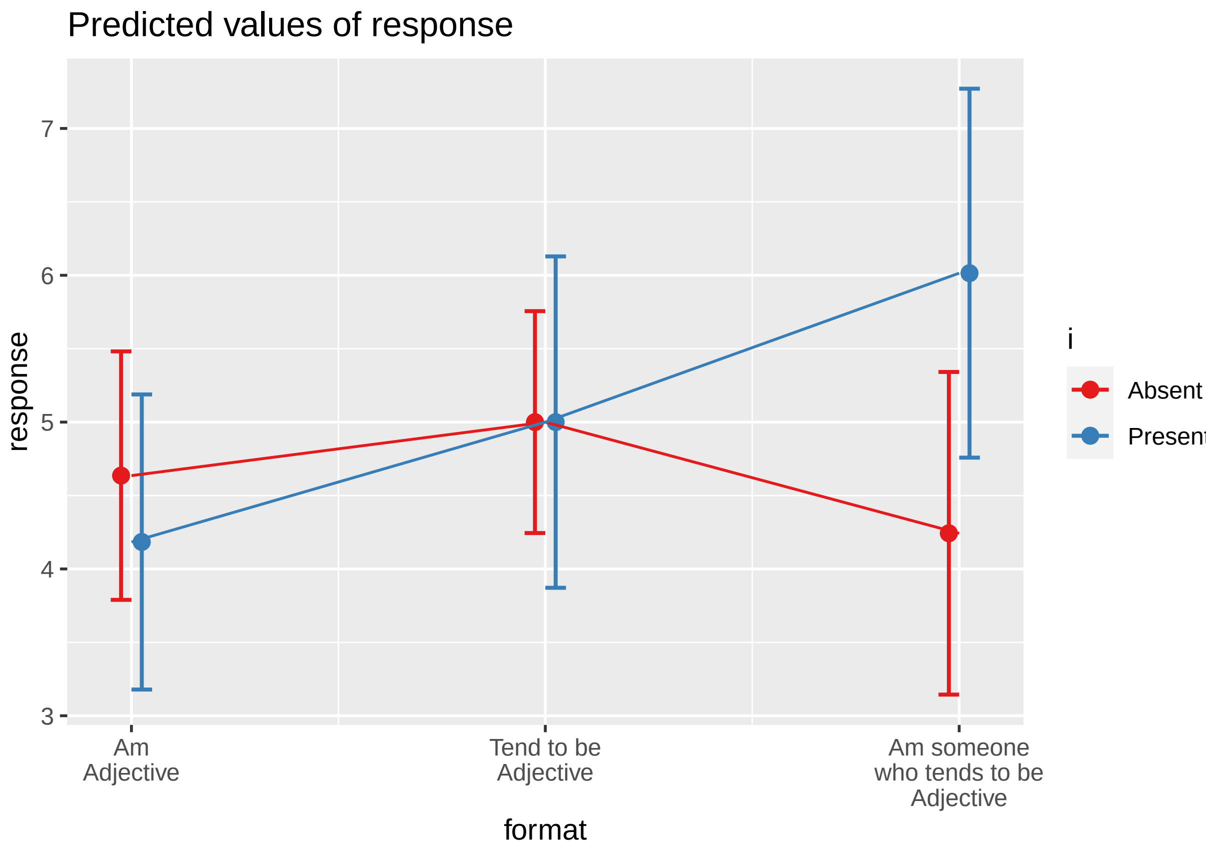
Table 13: Interaction of format and "I"

Term	Estimate	SE	t	df	p
(Intercept)	4.64	0.43	10.74	12.24	0.000
Tend to be Adjective	0.36	0.58	0.63	11.79	0.541
Am someone who tends to be Adjective	-0.39	0.71	-0.56	12.41	0.589
I	-0.45	0.44	-1.03	12.31	0.321
Tend to be Adjective:I	0.45	0.68	0.66	12.94	0.519
Am someone who tends to be Adjective:I	2.22	0.69	3.21	12.15	0.007

3.4.6 Sympathetic

```
sympathetic_model_i = items_13 %>%
  filter(item == "sympathetic") %>%
  lmer(response~format*i + (1|proid),
                              data = .)

sympathetic_model_i %>%
  tidy() %>%
  mutate(term = str_replace(term, "\n", " "),
         term = str_replace(term, "format", ""),
         term = str_replace(term, "iPresent", "I")) %>%
  filter(is.na(group)) %>%
  select(-effect, -group) %>%
  kable(booktabs = T,
        digits = c(0,2,2,2,2,3),
        caption = "Interaction of format and \"I\"",
        col.names = c("Term", "Estimate", "SE",  "t", "df", "p")) %>%
  kable_styling()
```

```
plot_model(sympathetic_model_i, type = "pred",
           terms = c("format", "i")) +
  geom_line()
```



Predicted values of response
response
7
6
5
4
3
Am
Adjective
Tend to be
Adjective
Am someone
who tends to be
Adjective
format
i
Absent
Present

4 Does the test-retest reliability of personality items change as a function of item wording?

We also plan to evaluate test-retest reliability within formats (within session and over two weeks); we expect slightly higher test-retest reliability for item wording formats that are more specific – formats #3 and #4 above vs the use of adjectives alone. In other words, we expect equal or lower retest reliability for the adjectives than for longer phrases. We will also consider the effect of performance on the word recall task on retest reliability .

4.1 Prep dataset

The data structure needed for these analsyes is in wide-format. That is, we require one column for each time point. In addition, we hope to examine reliability *within* format, which requires selecting only the response options which match the original, Block 1, assessment.

```
items_df = items_df %>%
  mutate(condition = tolower(condition)) %>%
  mutate(condition = factor(condition,
                            levels = c("a", "b", "c", "d"),
                            labels = c("Adjective\nOnly", "Am\nAdjective", "Tend to be\nAdjective", "Am

items_matchb1 = items_df %>%
  filter(format == condition) %>%
  mutate(block = paste0("block_", block)) %>%
  select(-timing, -seconds_log, -i) %>%
  spread(block, response)
```

We standardize responses within each block – this allows us to use a regression framework yet interpret the slopes as correlations.

```
items_matchb1 = items_matchb1 %>%
  mutate(across(
    starts_with("block"), ~(.-mean(., na.rm=T))/sd(., na.rm = T)
  ))
```

We also standardize the memory scores for ease of interpretation.

```
items_matchb1 = items_matchb1 %>%
  mutate(across(
    ends_with("memory"), ~(.-mean(., na.rm=T))/sd(., na.rm = T)
  ))
```

4.2 Test-retest reliability (all items pooled)

To estimate the reliability coefficients, we use a multilevel model, predicting the latter block from the earlier one. These models nest responses within participant, allowing us to estimate standard errors which account for the dependency of scores.

```
tr_mod1_b1b2 = lmer(block_2 ~ block_1 + (1 |proid), data = items_matchb1)
tr_mod1_b1b3 = lmer(block_3 ~ block_1 + (1 |proid), data = items_matchb1)

tab_model(tr_mod1_b1b2, tr_mod1_b1b3, show.re.var = F)
```

block 2

block 3

Predictors

Estimates

CI

p

Estimates

CI

p

(Intercept)

-0.02

-0.14 – 0.09

0.678

-0.04

-0.15 – 0.06

0.450

block_1

0.77

0.69 – 0.85

<0.001

0.67

0.60 – 0.73

<0.001

ICC

0.15

0.06

N

35 proid

19 proid

Observations

237

551

Marginal R2 / Conditional R2

0.599 / 0.658

0.421 / 0.456

4.3 Test-retest reliability (all items pooled, by format)

We fit these same models, except now we moderate by format, to determine whether the test-retest reliabilty differs as a function of item wording.

```
tr_mod2_b1b2 = lmer(block_2 ~ block_1*condition + (1 |proid),
                    data = items_matchb1)
tr_mod2_b1b3 = lmer(block_3 ~ block_1*condition + (1 |proid),
                    data = items_matchb1)

tab_model(tr_mod2_b1b2, tr_mod2_b1b3, show.re.var = F)
```

We also extract the simple slopes estimates of these models, which allow us to more explicitly identify and compare the test-retest correlations.

4.3.1 Block 1/Block 2

```
emtrends(tr_mod2_b1b2, pairwise ~ condition, var = "block_1")
```

```
## $emtrends
##   condition                              block_1.trend     SE  df lower.CL
##  Adjective\nOnly                                 0.605 0.0728 213    0.462
##  Am\nAdjective                                   0.850 0.0710 224    0.710
##  Tend to be\nAdjective                           0.809 0.0869 227    0.637
##  Am someone\nwho tends to be\nAdjective          0.877 0.1044 189    0.671
##  upper.CL
##     0.749
##     0.990
##     0.980
##     1.083
##
## Degrees-of-freedom method: kenward-roger
## Confidence level used: 0.95
##
## $contrasts
##    contrast                                                        estimate    SE
##  Adjective\nOnly - Am\nAdjective                                    -0.2452 0.102
##  Adjective\nOnly - Tend to be\nAdjective                            -0.2036 0.113
##  Adjective\nOnly - Am someone\nwho tends to be\nAdjective           -0.2720 0.127
##  Am\nAdjective - Tend to be\nAdjective                               0.0416 0.112
##  Am\nAdjective - Am someone\nwho tends to be\nAdjective             -0.0268 0.126
##  Tend to be\nAdjective - Am someone\nwho tends to be\nAdjective     -0.0684 0.136
##   df t.ratio p.value
##  219 -2.413  0.0777
##  228 -1.796  0.2777
##  219 -2.137  0.1448
##  229  0.371  0.9825
##  214 -0.212  0.9966
##  210 -0.503  0.9582
##
## Degrees-of-freedom method: kenward-roger
## P value adjustment: tukey method for comparing a family of 4 estimates
```

4.3.2 Block 1/Block 3

```
emtrends(tr_mod2_b1b3, pairwise ~ condition, var = "block_1")
```

```
## $emtrends
##   condition                            block_1.trend     SE  df lower.CL
##  Adjective\nOnly                               0.677 0.0649 535    0.550
##  Am\nAdjective                                 0.659 0.0621 539    0.537
##  Tend to be\nAdjective                         0.683 0.0673 278    0.550
##  Am someone\nwho tends to be\nAdjective        0.621 0.0751 543    0.473
##  upper.CL
##     0.805
##     0.781
##     0.815
##     0.768
##
## Degrees-of-freedom method: kenward-roger
## Confidence level used: 0.95
##
## $contrasts
##    contrast                                                  estimate     SE
##  Adjective\nOnly - Am\nAdjective                              0.01849 0.0899
##  Adjective\nOnly - Tend to be\nAdjective                     -0.00562 0.0935
##  Adjective\nOnly - Am someone\nwho tends to be\nAdjective     0.05634 0.0993
##  Am\nAdjective - Tend to be\nAdjective                       -0.02411 0.0916
##  Am\nAdjective - Am someone\nwho tends to be\nAdjective       0.03785 0.0975
##  Tend to be\nAdjective - Am someone\nwho tends to be\nAdjective  0.06196 0.1009
##   df t.ratio p.value
##  543  0.206  0.9969
##  413 -0.060  0.9999
##  541  0.567  0.9417
##  441 -0.263  0.9936
##  543  0.388  0.9801
##  457  0.614  0.9275
##
## Degrees-of-freedom method: kenward-roger
## P value adjustment: tukey method for comparing a family of 4 estimates
```

4.4 Test-retest reliability (all items pooled, by memory)

Here we fit models moderated by memory – that it, perhaps the test-retest coefficient is affected by the memory of the participant.

```
tr_mod3_b1b2 = lmer(block_2 ~ block_1*delayed_memory +
                      (1 |proid),
                    data = items_matchb1)
tr_mod3_b1b3 = lmer(block_3 ~ block_1*very_delayed_memory +
                      (1 |proid),
                    data = items_matchb1)

tab_model(tr_mod3_b1b2, tr_mod3_b1b3, show.re.var = F)
```

We also extract the simple slopes estimates of these models, which allow us to more explicitly identify and compare the test-retest correlations.

4.4.1 Block 1/Block 2

```
mem_list = list(delayed_memory = c(-1,0,1))

emtrends(tr_mod3_b1b2,
         pairwise~delayed_memory,
         var = "block_1",
         at = mem_list)
```

```
## $emtrends
##  delayed_memory block_1.trend     SE  df lower.CL upper.CL
##              -1         0.811 0.0580 232    0.697    0.925
##               0         0.762 0.0418 231    0.679    0.844
##               1         0.712 0.0690 231    0.576    0.848
##
## Degrees-of-freedom method: kenward-roger
## Confidence level used: 0.95
##
## $contrasts
##  contrast estimate     SE  df t.ratio p.value
##  (-1) - 0   0.0495 0.0481 232 1.029   0.5592
##  (-1) - 1   0.0990 0.0962 232 1.029   0.5592
##  0 - 1      0.0495 0.0481 232 1.029   0.5592
##
## Degrees-of-freedom method: kenward-roger
## P value adjustment: tukey method for comparing a family of 3 estimates
```

4.4.2 Block 1/Block 3

This chunk is turned off due to low coverage. Be sure to turn on with real data.

```
mem_list = list(very_delayed_memory = c(-1,0,1))

emtrends(tr_mod3_b1b3,
         pairwise~very_delayed_memory,
         var = "block_1",
         at = mem_list)
```

```
## $emtrends
##  very_delayed_memory block_1.trend     SE  df lower.CL upper.CL
##                   -1         0.678 0.0485 545    0.582    0.773
##                    0         0.665 0.0341 546    0.598    0.732
##                    1         0.653 0.0553 547    0.544    0.762
##
## Degrees-of-freedom method: kenward-roger
## Confidence level used: 0.95
##
## $contrasts
```

```
##  contrast estimate     SE  df t.ratio p.value
##  (-1) - 0   0.0123 0.0393 547 0.312   0.9478
##  (-1) - 1   0.0245 0.0786 547 0.312   0.9478
##  0 - 1      0.0123 0.0393 547 0.312   0.9478
##
## Degrees-of-freedom method: kenward-roger
## P value adjustment: tukey method for comparing a family of 3 estimates
```

4.5 Test-retest reliability (items separated, by format)

To assess test-retest reliability for each item, we can rely on more simple correlation analyses, as each participant only contributed one response to each item in each block. We first not the sample size coverage for these comparisons:

```
items_matchb1 %>%
  group_by(item, condition) %>%
  count() %>%
  ungroup() %>%
  full_join(expand_grid(item = unique(items_matchb1$item),
                        condition = unique(items_matchb1$condition))) %>%
  mutate(n = ifelse(is.na(n), 0, n)) %>%
  summarise(
    min = min(n),
     max = max(n),
    mean = mean(n),
    median = median(n)
  )
```

```
## # A tibble: 1 x 4
##     min   max  mean median
##   <dbl> <dbl> <dbl>  <dbl>
## 1     0    11   8.5      8
```

```
items_matchb1 %>%
  group_by(item, condition) %>%
  count() %>%
  ungroup() %>%
  full_join(expand_grid(item = unique(items_matchb1$item),
                        condition = unique(items_matchb1$condition))) %>%
  mutate(n = ifelse(is.na(n), 0, n)) %>%
  ggplot(aes(x = n)) +
  geom_histogram(bins = 50) +
  labs(x = "Sample size",
       y = "Number of tests") +
  facet_wrap(~condition)
```

```
items_cors = items_matchb1 %>%
  select(item, condition, contains("block")) %>%
  group_by(item, condition) %>%
  nest() %>%
   mutate(cors = map(data, psych::corr.test, use = "pairwise"),
```

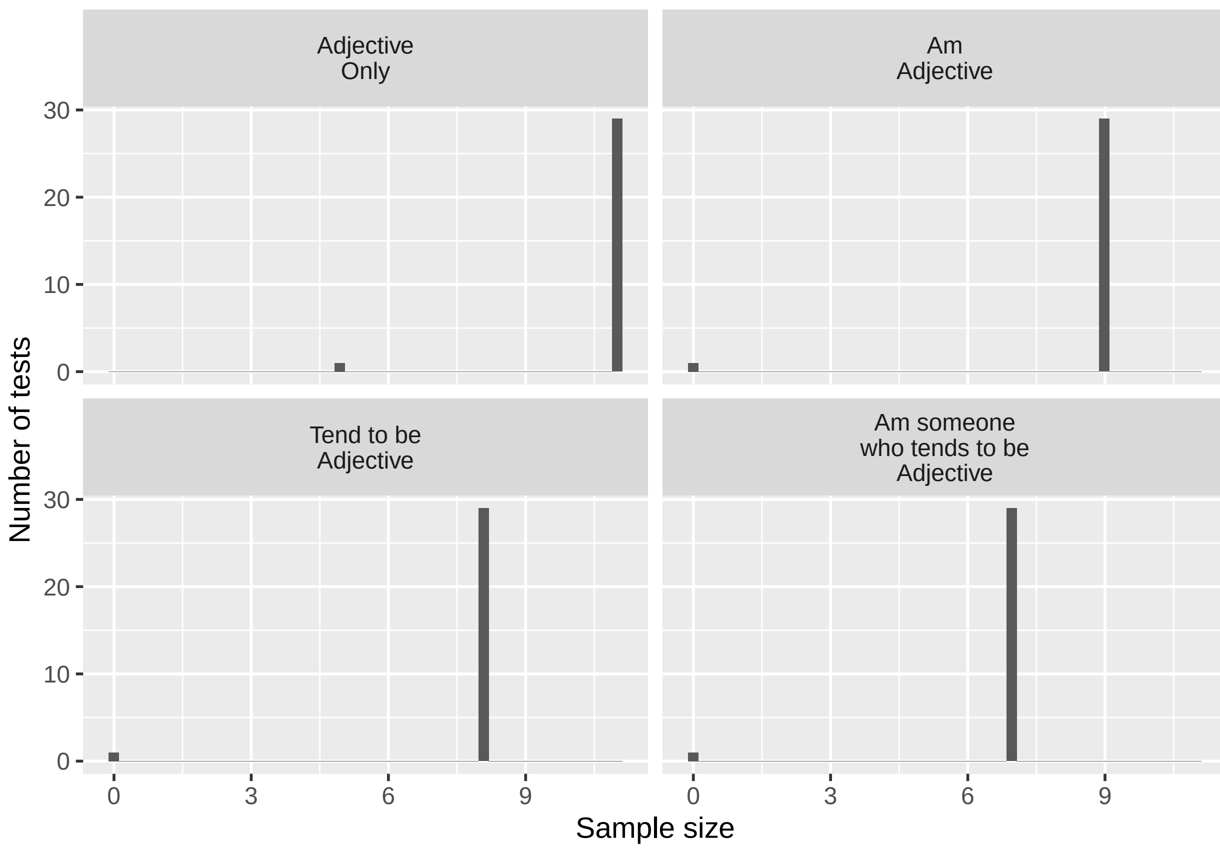


Figure 23: Sample sizes for item-level test-retest correlations.

```
         cors = map(cors, print, short = F),
         cors = map(cors, ~.x %>% mutate(comp = rownames(.)))) %>%
  select(item, condition, cors) %>%
  unnest(cols = c(cors))
```

```
items_cors %>%
  mutate(raw.r = printnum(raw.r),
         raw.r = case_when(
           is.na(raw.p) ~ NA_character_,
           raw.p < .05 ~ paste0(raw.r, "*"),
           TRUE ~ raw.r)) %>%
  select(item, condition, comp, raw.r) %>%
  mutate(reverse = case_when(
    item %in% reverse ~ "Y",
    TRUE ~ "N"
  )) %>%
  filter(comp != "blc_2-blc_3") %>%
  mutate(condition = case_when(
    condition == "Adjective\nOnly" ~ "a",
    condition == "Am\nAdjective" ~ "b",
    condition == "Tend to be\nAdjective" ~ "c",
    condition == "Am someone\nwho tends to be\nAdjective" ~ "d"
  )) %>%
  unite(comp, condition, comp) %>%
  spread(comp, raw.r) %>%
  arrange(reverse, item) %>%
  kable(caption = "Test-retest correlations for each item and condition. Preregistration note: given the
        col.names = c("Item", "Reverse scored?", rep(c("5 min", "2 weeks"), 4)),
        booktabs = T) %>%
  kable_styling() %>%
  add_header_above(c(" " = 2, "Adjective Only" = 2, "Am Adjective" = 2,
                     "Tend to be" = 2, "Am someone who tends to be" = 2))
```

```
items_cors %>%
  mutate(comp_num = case_when(
    comp == "blc_1-blc_2" ~ 1,
    comp == "blc_1-blc_3" ~ 2,
    comp == "blc_2-blc_3" ~ NA_real_,
  )) %>%
  filter(!is.na(comp_num)) %>%
  ggplot(aes(x = comp_num, y = raw.r, color = condition)) +
  geom_jitter(width = .1) +
  scale_x_continuous(breaks = c(1:2),
                     labels = c("1-2", "1-3")) +
  labs(x = NULL, y = "Correlation") +
  theme_pubclean()
```

Table 14: Test-retest correlations for each item and condition. Preregistration note: given the low sample size for the pilot data, we are missing observations for many of these comparisons. Correlations which could not be computed are blank in this table, but we expect them to be reported in the final manuscript.

		Adjective Only		Am Adjective		Tend to be		Am someone who tends to be	
Item	Reverse scored?	5 min	2 weeks	5 min	2 weeks	5 min	2 weeks	5 min	2 weeks
active	N	0.87	-0.25	0.93	0.96*	1.00*	0.75		-0.50
adventurous	N	0.50	-0.13		0.00		0.17	0.97	-0.87
calm	N	0.58	0.61		0.87		0.40		-0.19
caring	N	-0.50	0.92*		-0.61	1.00*	0.34		
cautious	N		0.64		0.32		-0.42		0.91
creative	N	0.87	0.53	1.00*	0.50	0.50	0.71		
curious	N		0.71	0.30	0.17		0.55		0.87
friendly	N		0.67		0.61		-0.54	1.00*	
hardworking	N		-0.25		1.00*		0.50		
helpful	N	0.00	0.00	1.00*	0.41		0.25		
imaginative	N	0.90	0.87		0.25	0.98	0.87*		
impusive	N								
intelligent	N		0.80		0.61		0.70		0.50
lively	N		0.98*		0.66		0.69		
organized	N	0.97*	0.87		0.42		0.98*		
outgoing	N		0.88	0.77	0.80		0.95*		
responsible	N		1.00*		0.61		0.48		-0.50
softhearted	N		0.94*		0.91*		0.49		0.98
sophisticated	N	1.00*	-0.40	0.96*	0.61		0.62		0.97
sympathetic	N		0.53	0.50	0.69	0.50		0.43	1.00*
talkative	N		0.12		0.91*		0.90*		-0.19
thorough	N		0.76	0.50	-0.13	0.50	0.81		-0.50
thrifty	N	0.93	0.69	0.91	0.71		0.90*		0.00
warm	N		1.00*		-0.65		0.81		-0.50
careless	Y	0.87	0.15	0.69	0.89*		0.16		
impulsive	Y		0.75		0.78	0.96	0.87*		0.62
moody	Y	0.79	0.94*		0.87		0.10		0.84
nervous	Y		0.54	-1.00*	0.59		-0.58		-0.87
reckless	Y	0.87	0.94*		0.59	-0.18	-0.08		-0.87
worrying	Y	0.00	0.96*		0.64		0.80		

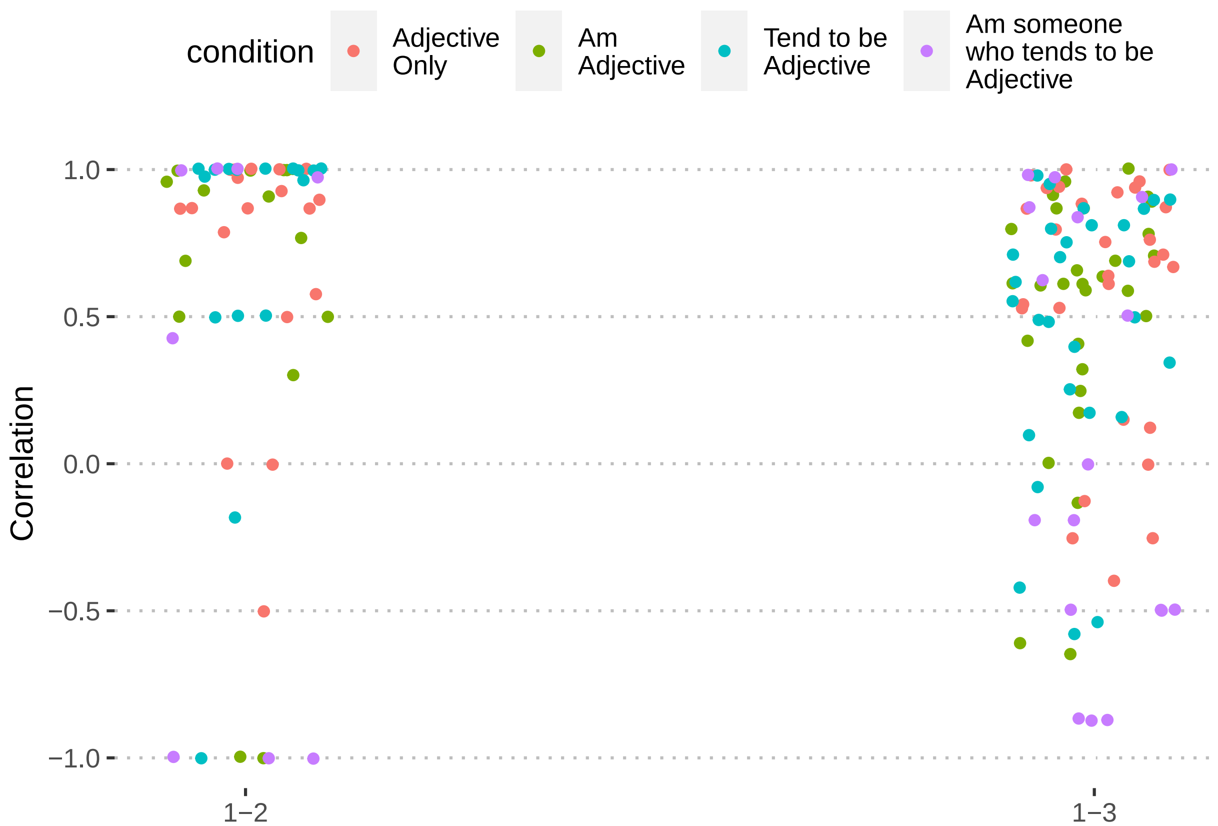


Figure 24: Test-retest correlations of specific items across word format. Each dot represents the test-retest correlation within a specific item.

5 How does format affect timining of responses?

5.1 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) +
```

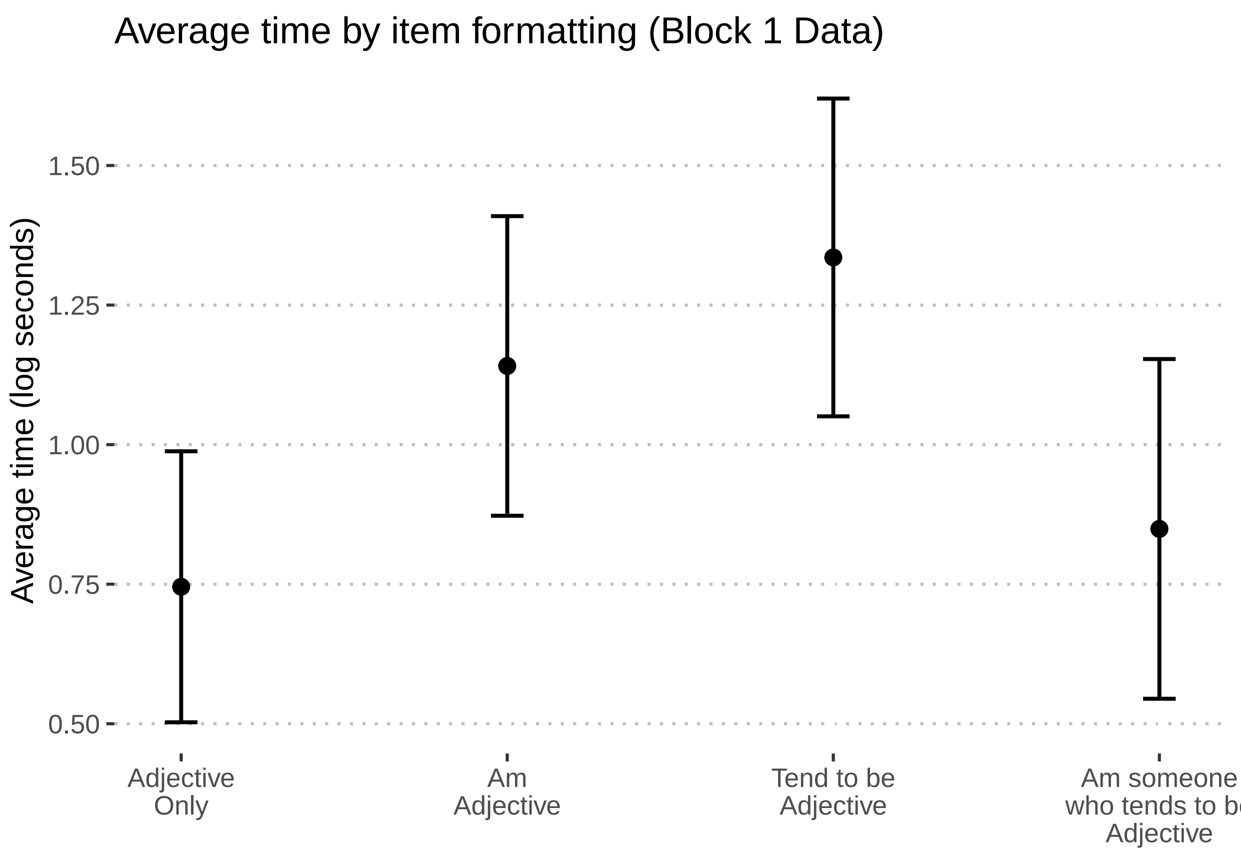


Figure 25: Predicted seconds (log) on personality items by condition, using only Block 1 data.

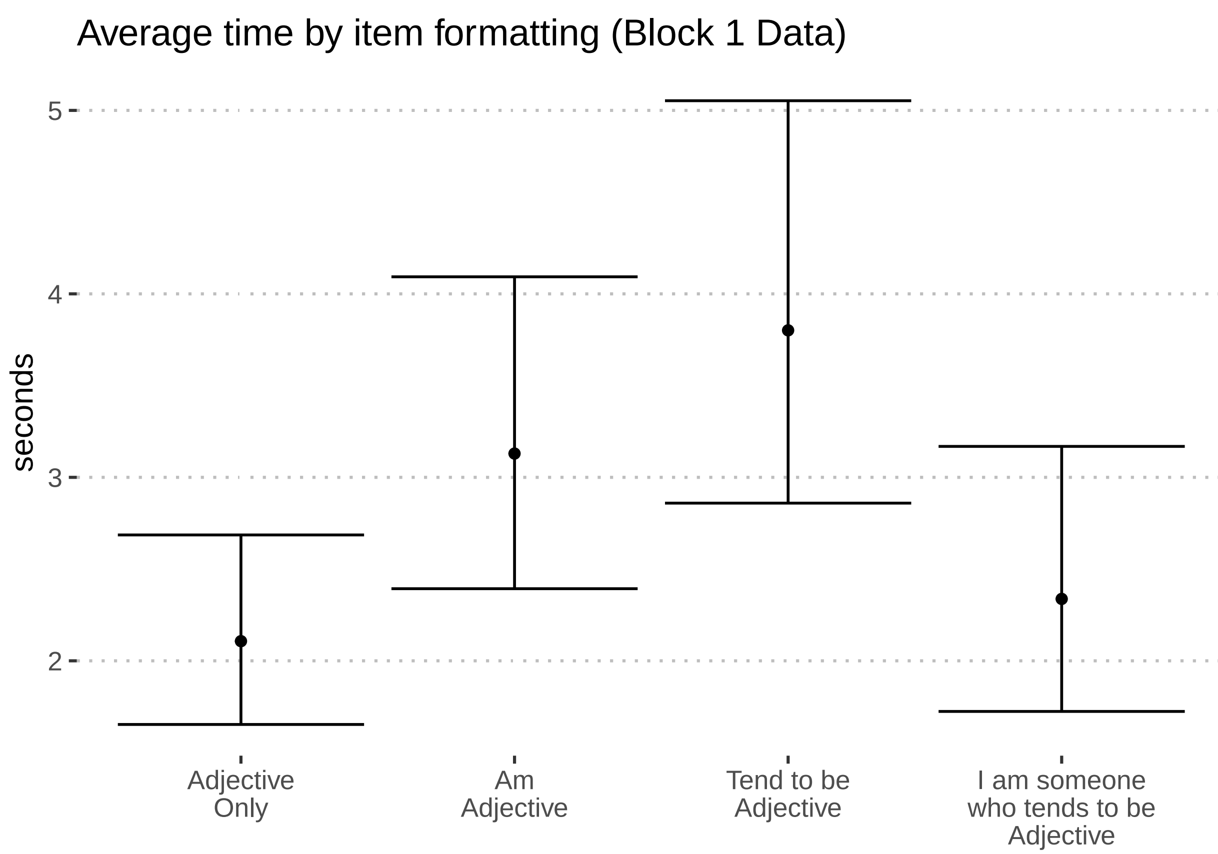


Figure 26: Predicted seconds on personality items by condition, using only Block 1 data.

```
  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()
```

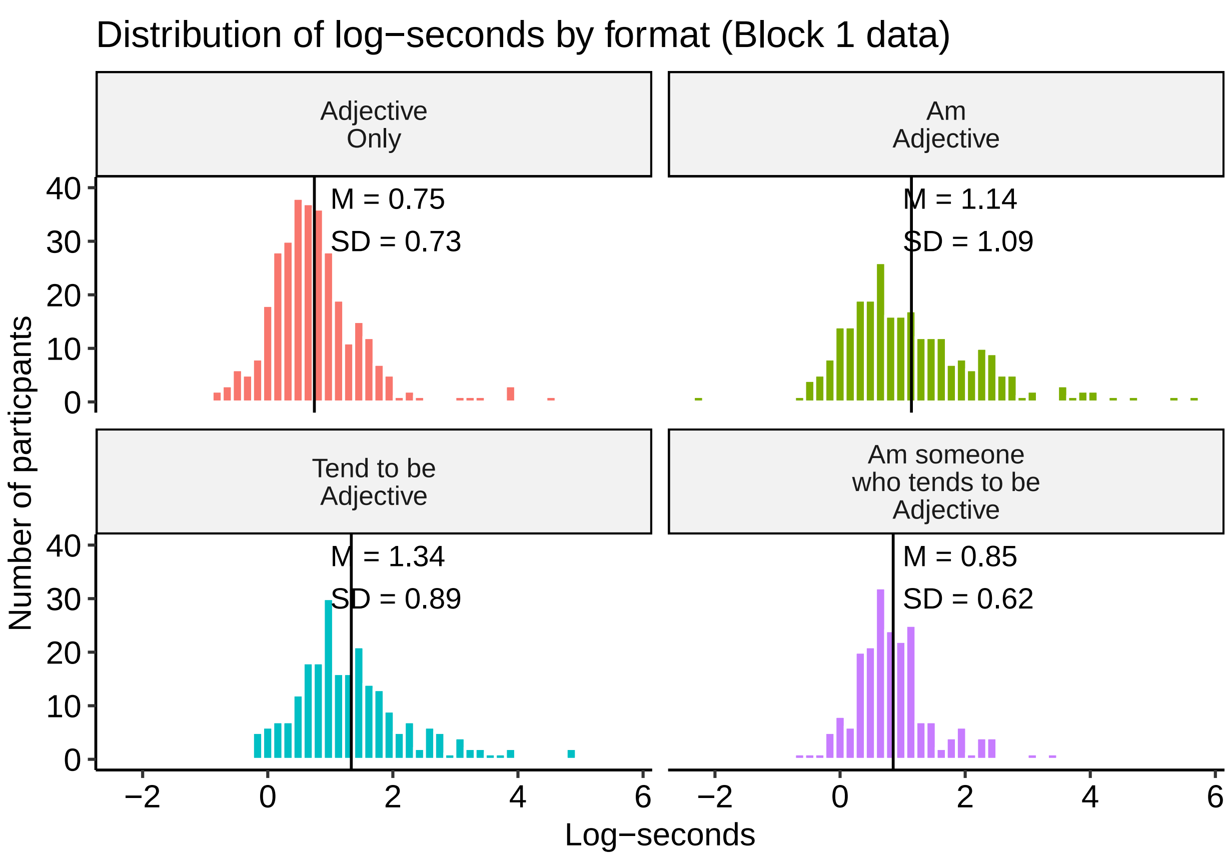


Figure 27: Distribution of time by category, block 1 data only

```
pairs(emmeans(mod.format_b1, "format"), adjust = "holm") %>%
  kable(booktabs = T, digits = c(0, 2,2,1,2,3)) %>%
  kable_styling()
```

5.1.0.1 Pairwise comparisons

contrast	estimate	SE	df	t.ratio	p.value
Adjective Only - Am Adjective	-0.40	0.18	31	-2.14	0.160
Adjective Only - Tend to be Adjective	-0.59	0.19	31	-3.09	0.025
Adjective Only - Am someone who tends to be Adjective	-0.10	0.20	31	-0.52	0.675
Am Adjective - Tend to be Adjective	-0.19	0.20	31	-0.97	0.675
Am Adjective - Am someone who tends to be Adjective	0.29	0.21	31	1.41	0.505
Tend to be Adjective - Am someone who tends to be Adjective	0.49	0.21	31	2.29	0.146

5.1.1 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()
```

5.1.2 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!

Table 15: 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 16: 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 - 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 - Am someone who tends to be Adjective	-0.26	0.23	31	-1.13	0.803
Tend to be Adjective - Am someone who tends to be Adjective	-0.20	0.24	31	-0.84	0.812

5.1.3 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()
```

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

5.1.4 Caring

```
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()
```

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

5.1.5 Soft-hearted

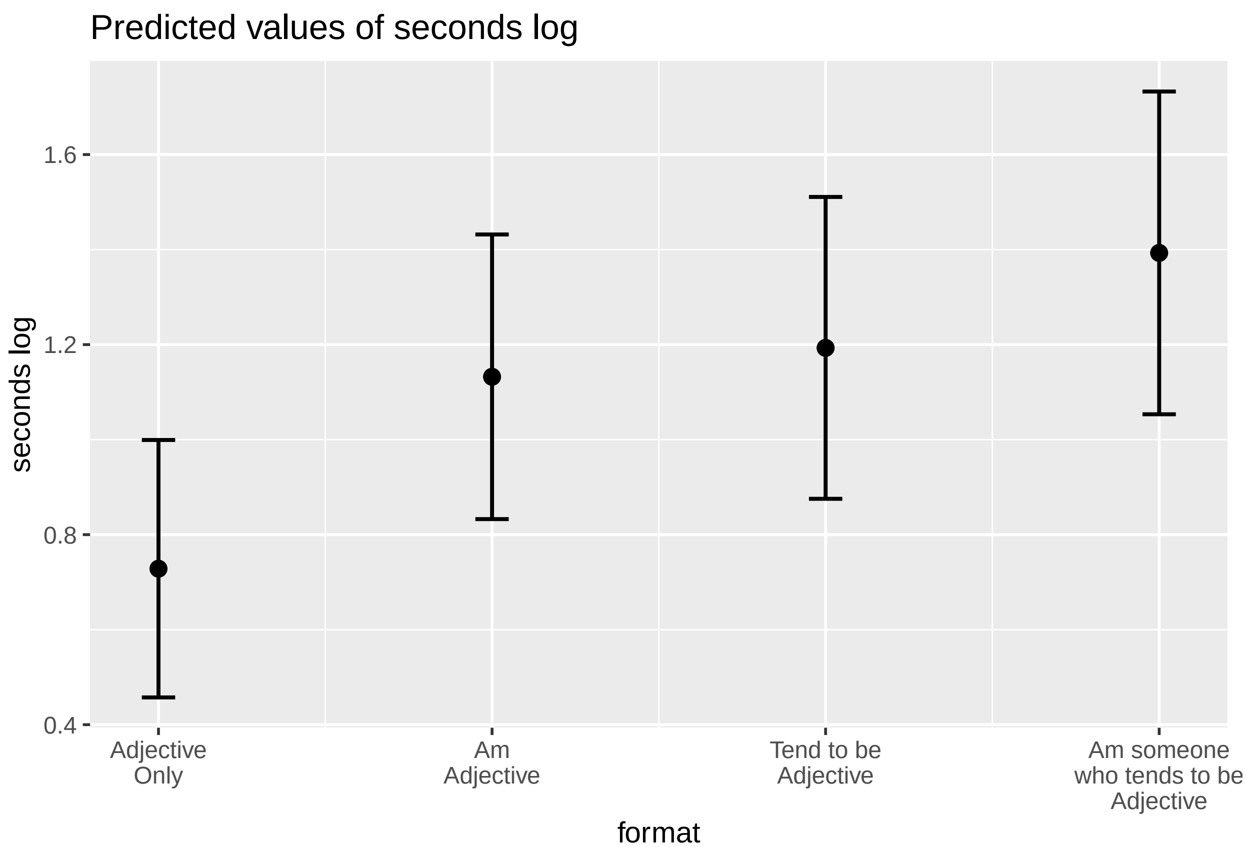


Figure 28: Average log-seconds to “helpful” by format (block 1 data only)

Table 17: 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 - 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 - Am someone who tends to be Adjective	0.71	0.33	31	2.15	0.189
Tend to be Adjective - Am someone who tends to be Adjective	0.34	0.34	31	0.99	0.750

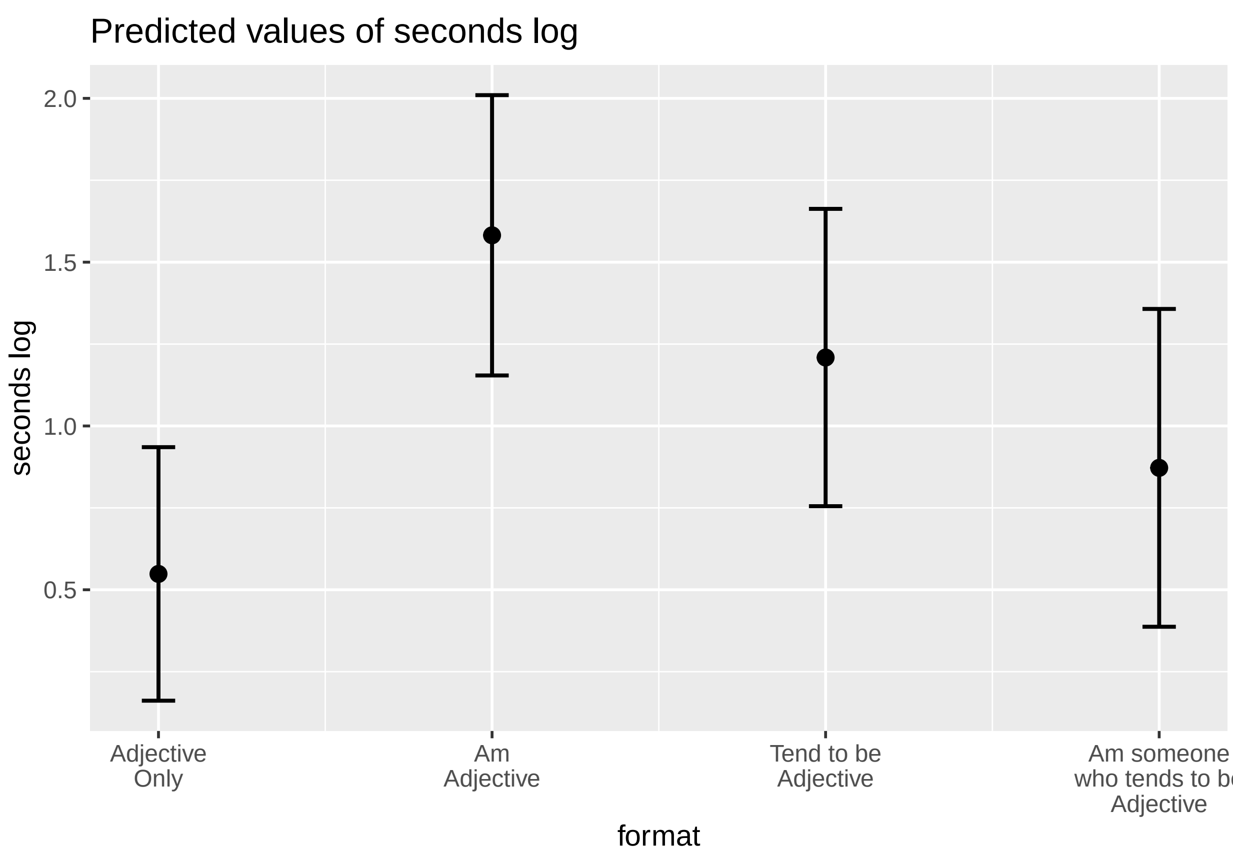


Figure 29: Average log-seconds to “caring” by format (block 1 data only)

Table 18: 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 - 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 - Am someone who tends to be Adjective	0.95	0.35	31	2.69	0.046
Tend to be Adjective - Am someone who tends to be Adjective	0.90	0.36	31	2.47	0.057

```
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"))
```

5.1.6 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"))
```

5.1.7 Sympathetic

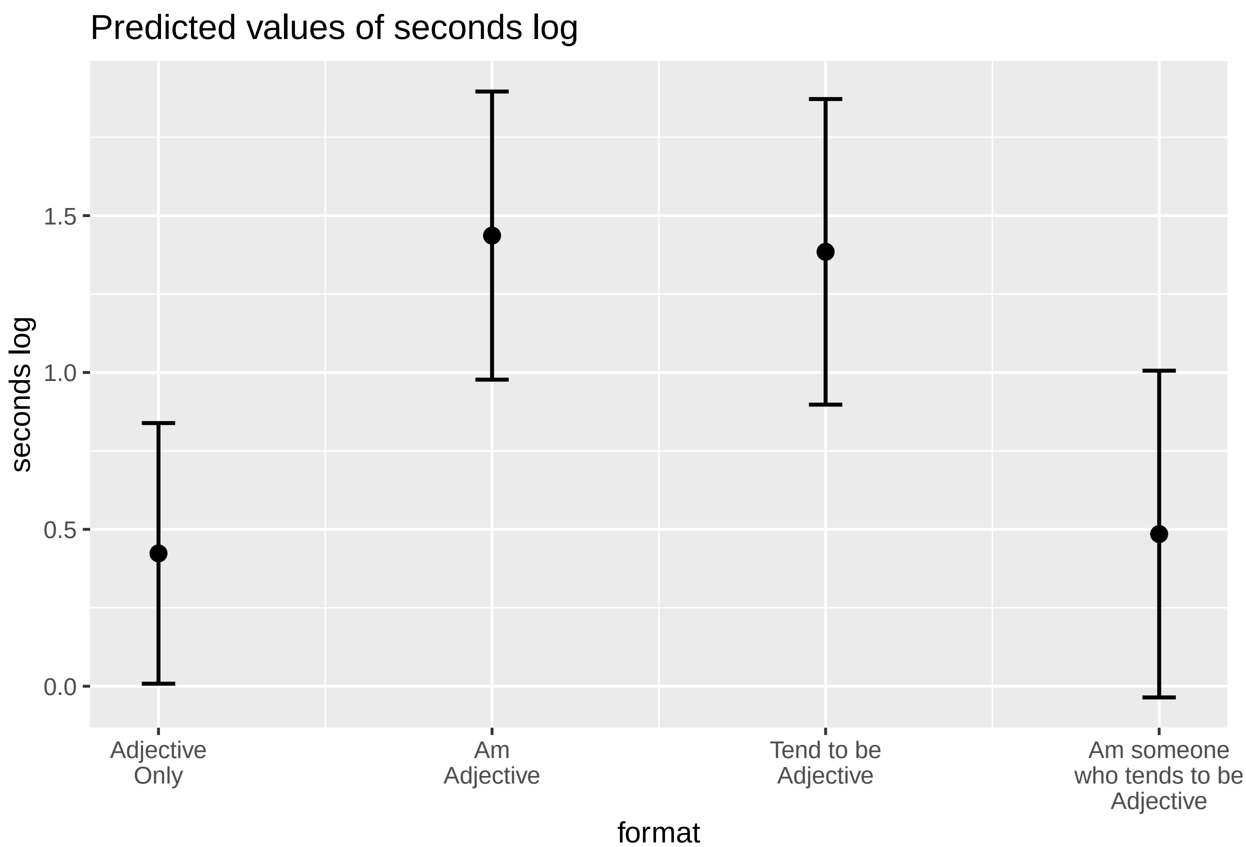


Figure 30: Average log-seconds to “softhearted” by format (block 1 data only)

Table 19: 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 - 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 - Am someone who tends to be Adjective	1.08	0.47	31	2.30	0.142
Tend to be Adjective - Am someone who tends to be Adjective	0.92	0.48	31	1.91	0.197

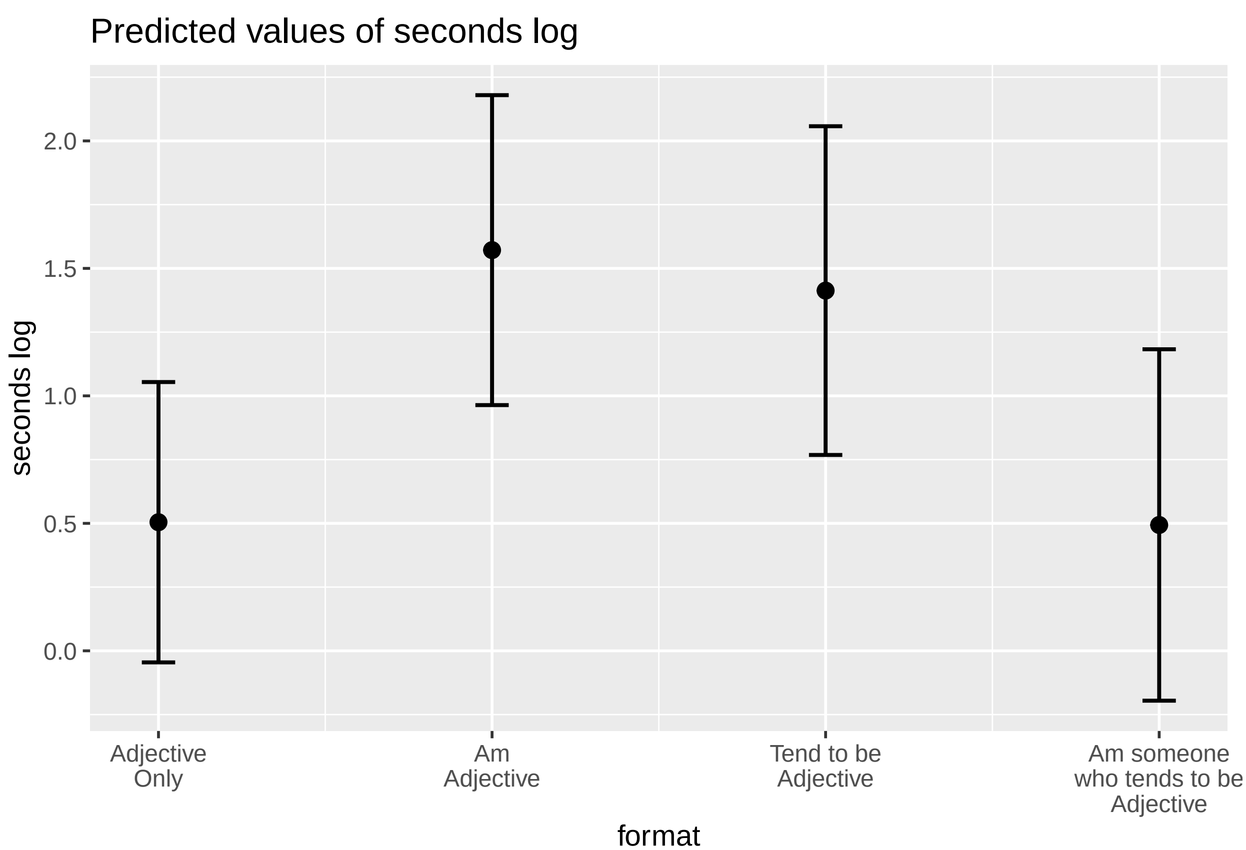


Figure 31: Average log-seconds to "calm" by format (block 1 data only)

Table 20: 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 - 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 - Am someone who tends to be Adjective	0.69	0.42	31	1.66	0.320
Tend to be Adjective - Am someone who tends to be Adjective	1.62	0.43	31	3.79	0.004

```r
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()
```

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

5.1.8 Adventurous

```r
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()
```

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

5.2 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 data from blocks 1 and 2.

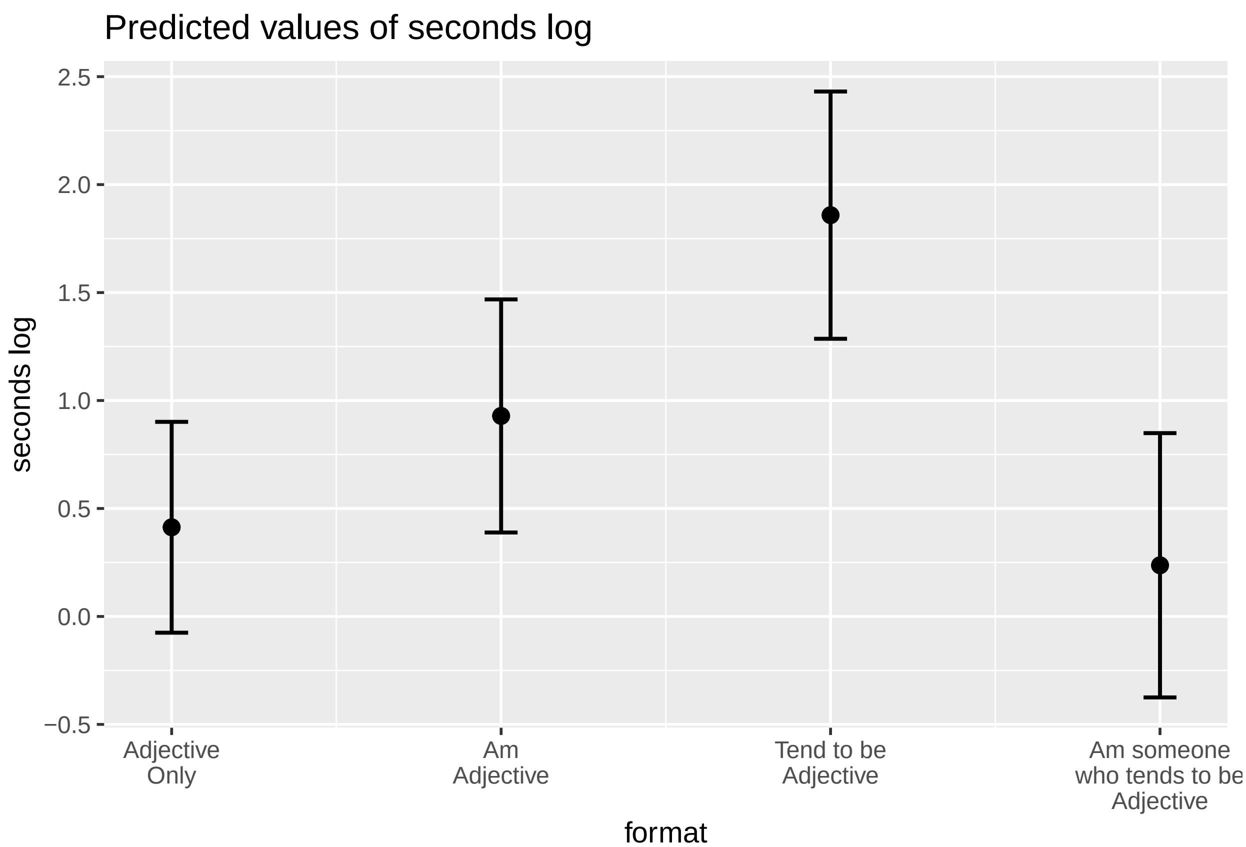


Figure 32: Average log-seconds to “sympathetic” by format (block 1 data only)

Table 21: 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 - 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 - Am someone who tends to be Adjective	-0.14	0.30	31	-0.48	1.000
Tend to be Adjective - Am someone who tends to be Adjective	0.55	0.31	31	1.79	0.332

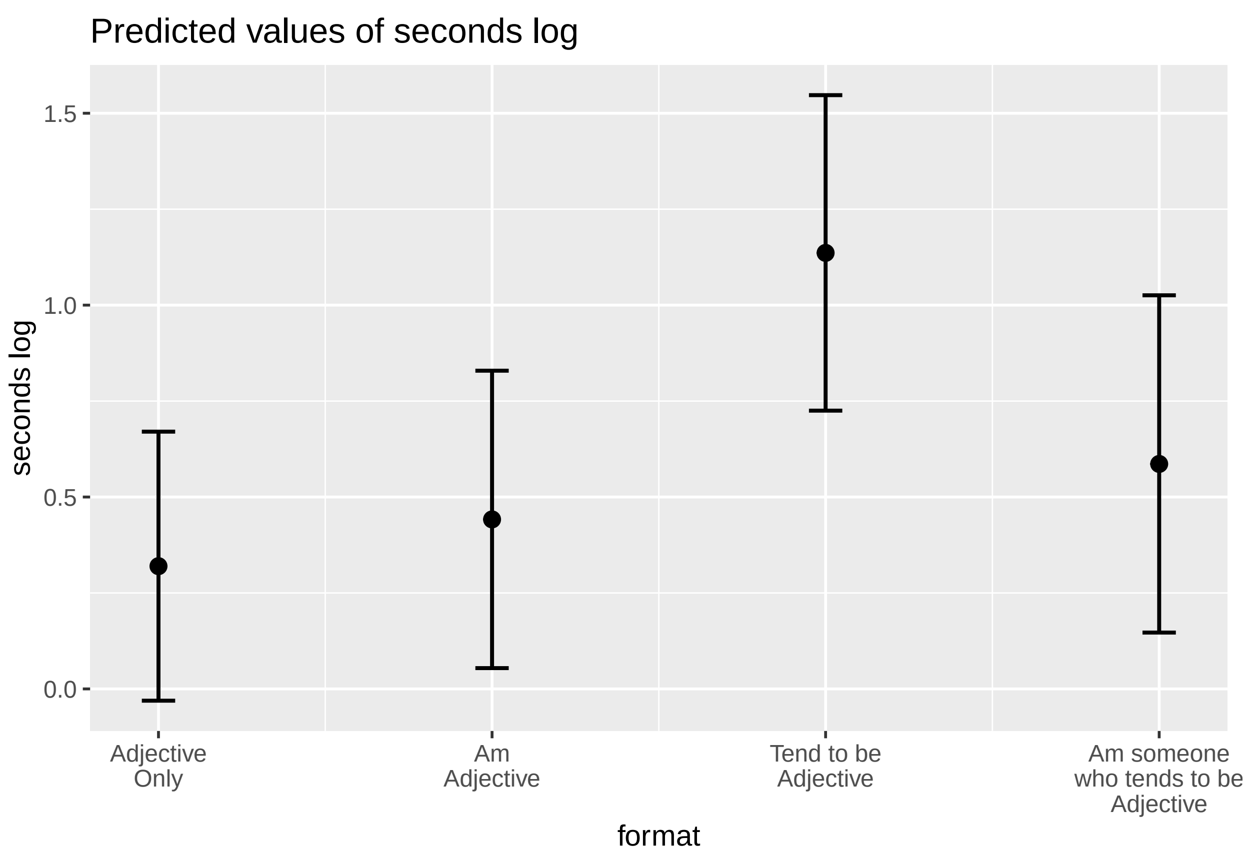


Figure 33: Average log-seconds to “adventurous” by format (block 1 data only)

```r
items_12 = items_df %>% filter(block %in% c("1","2"))
```

```r
mod.format_b2 = lmer(seconds_log~format + (1|proid),
                     data = items_12)
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
```

```r
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()
```

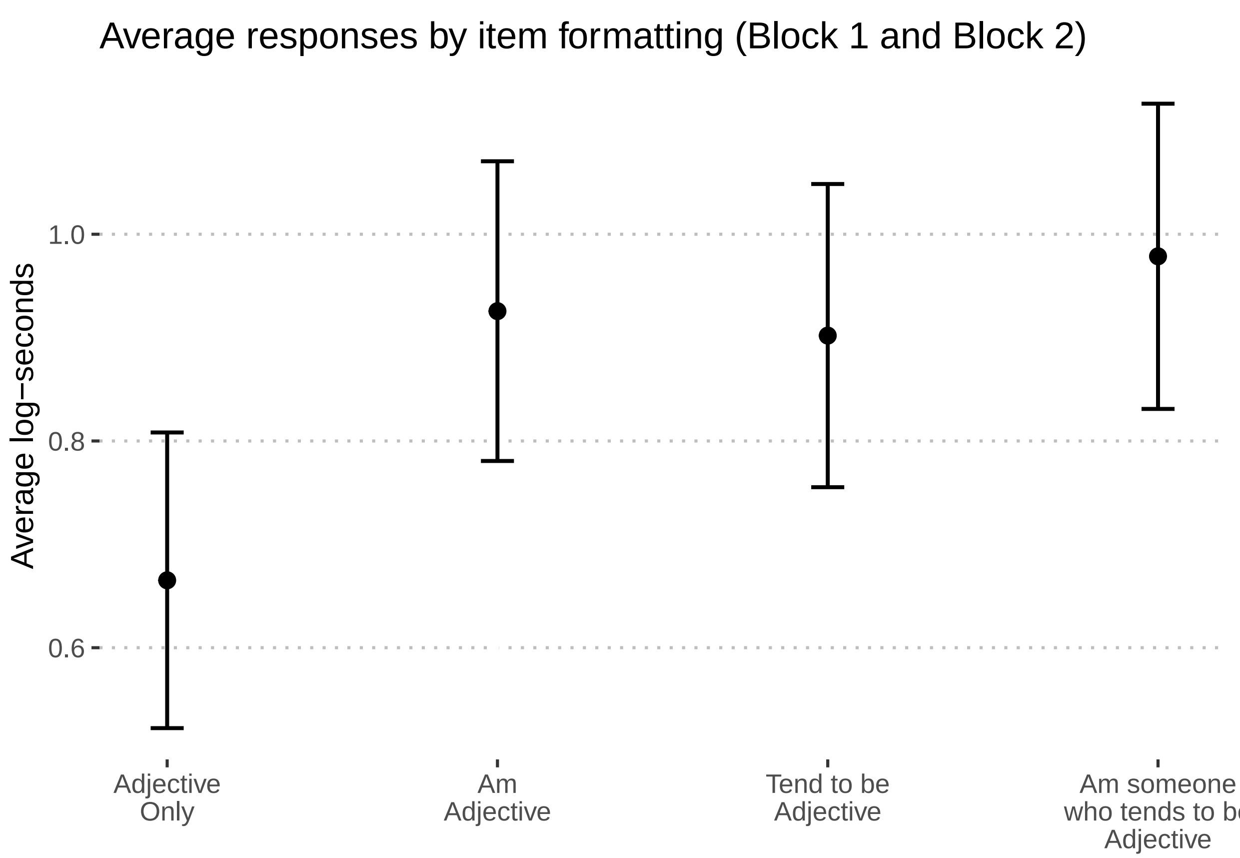


Figure 34: Predicted log-seconds on personality items by condition, using only Block 1 data.

```
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()
```

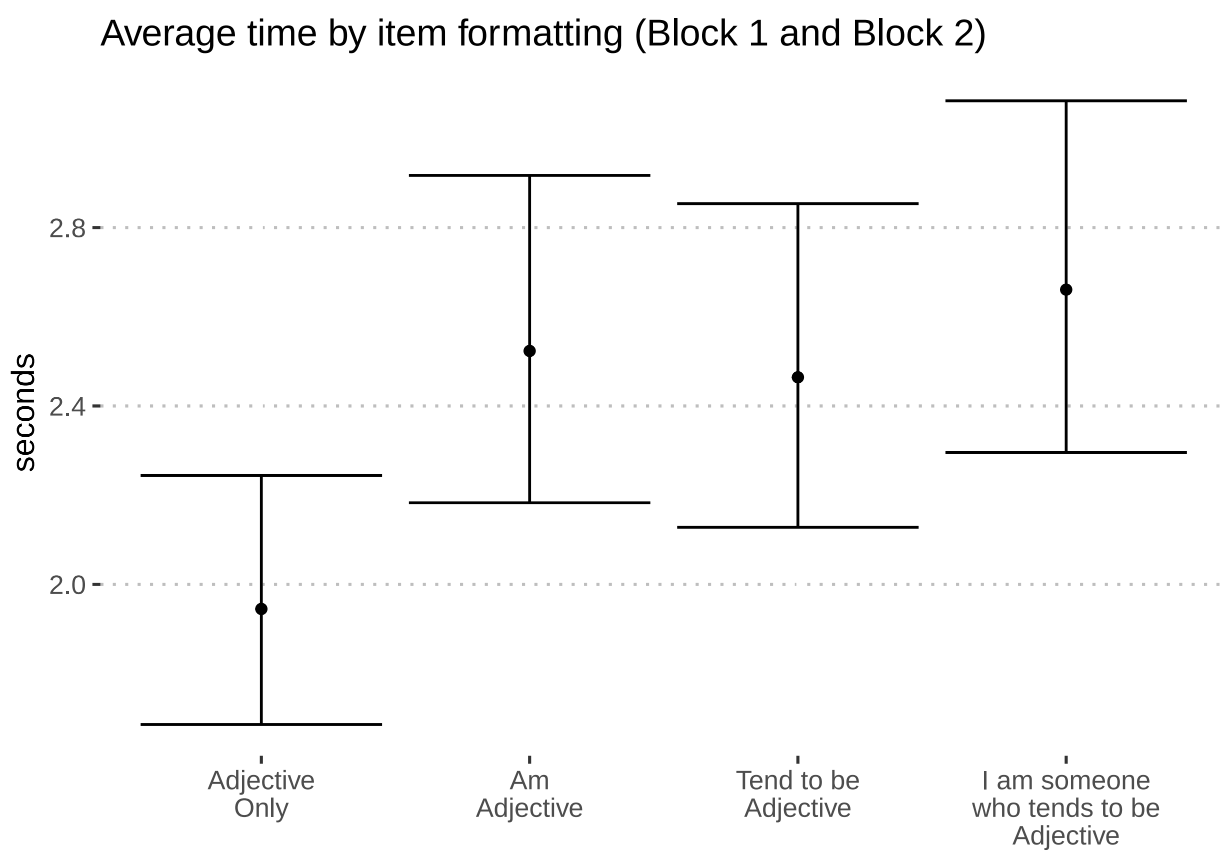


Figure 35: Predicted seconds on personality items by condition, using only Block 1 data.

```
means_by_group = items_12 %>%
  group_by(format) %>%
  summarise(m = mean(seconds_log),
            s = sd(seconds_log))

items_12 %>%
  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()
```

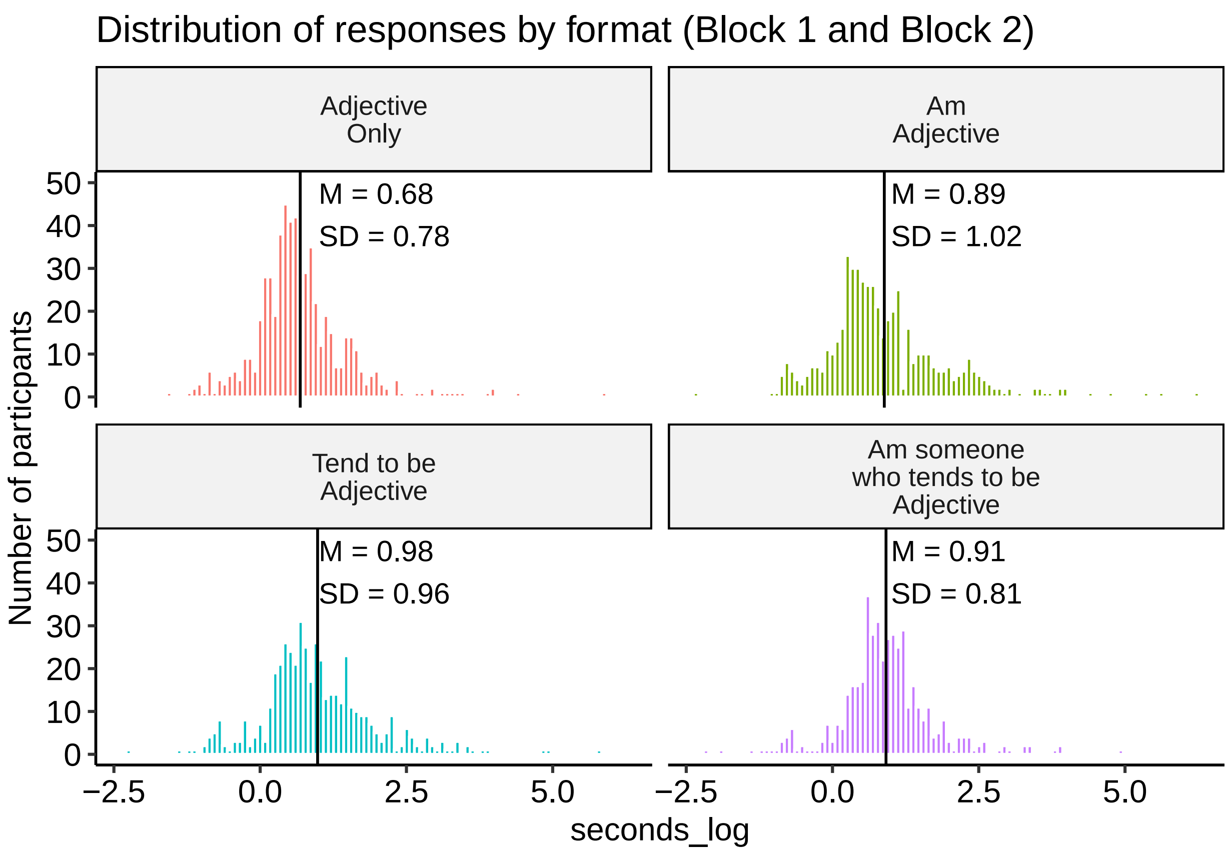


Figure 36: Distribution of log-seconds by category, block 1 and block 2

```
pairs(emmeans(mod.format_b2, "format"), adjust = "holm") %>%
  kable(booktabs = T, digits = c(0, 2,2,1,2,3)) %>%
  kable_styling()
```

5.2.0.1 Pairwise comparisons

contrast	estimate	SE	df	t.ratio	p.value
Adjective Only - Am Adjective	-0.26	0.06	2015.9	-4.60	0.000
Adjective Only - Tend to be Adjective	-0.24	0.06	2009.1	-4.04	0.000
Adjective Only - Am someone who tends to be Adjective	-0.31	0.06	2018.3	-5.35	0.000
Am Adjective - Tend to be Adjective	0.02	0.06	2014.6	0.40	0.752
Am Adjective - Am someone who tends to be Adjective	-0.05	0.06	2018.8	-0.89	0.752
Tend to be Adjective - Am someone who tends to be Adjective	-0.08	0.06	2024.4	-1.28	0.601

5.2.1 One model for each adjective

We can also repeat this analysis separately for each trait.

```
mod_by_item_b2 = items_12 %>%
  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 %>%
  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()
```

5.2.2 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!

Table 22: 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

Table 23: 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 - 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 - Am someone who tends to be Adjective	-0.21	0.18	65.08	-1.20	0.702
Tend to be Adjective - Am someone who tends to be Adjective	-0.18	0.19	65.85	-0.94	0.702

5.2.3 Helfpul

```
helpful_model_b2 = items_12 %>%
  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"))
```

5.2.4 Caring

```
caring_model_b2 = items_12 %>%
  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"))
```

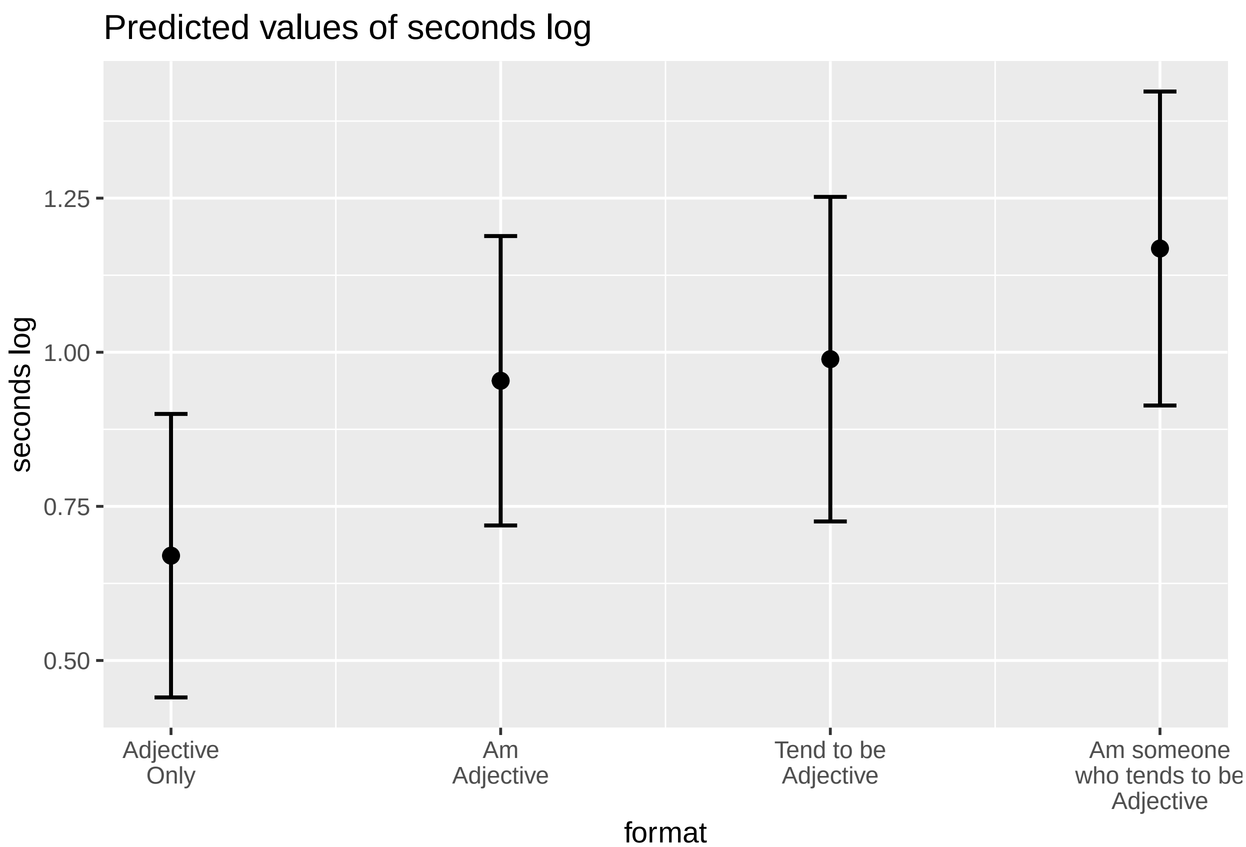


Figure 37: Average log-seconds to “helpful” by format (Block 1 and Block 2)

Table 24: 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 - 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 - Am someone who tends to be Adjective	0.36	0.26	65.29	1.35	0.723
Tend to be Adjective - Am someone who tends to be Adjective	0.31	0.27	64.65	1.16	0.757

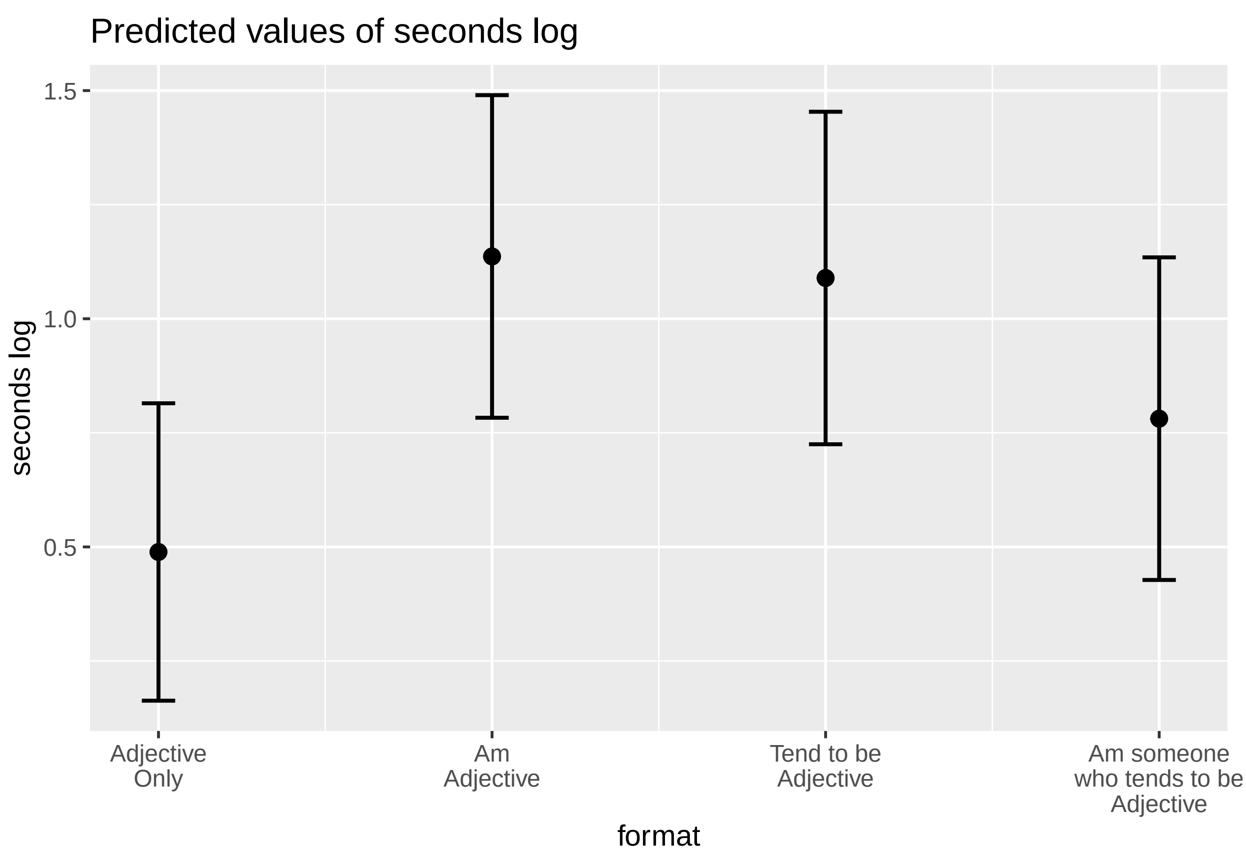


Figure 38: Average log-seconds to "caring" by format (Block 1 and Block 2)

Table 25: 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 - 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 - Am someone who tends to be Adjective	-0.26	0.28	62.04	-0.93	0.724
Tend to be Adjective - Am someone who tends to be Adjective	0.10	0.28	63.89	0.37	0.724

5.2.5 Adventurous

```
adventurous_model_b2 = items_12 %>%
  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"))
```

5.3 Analysis: Account for memory effects

```
mod.format_mem = lmer(seconds_log~format*delayed_memory + (1|proid),
                  data = items_12)
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
```

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

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

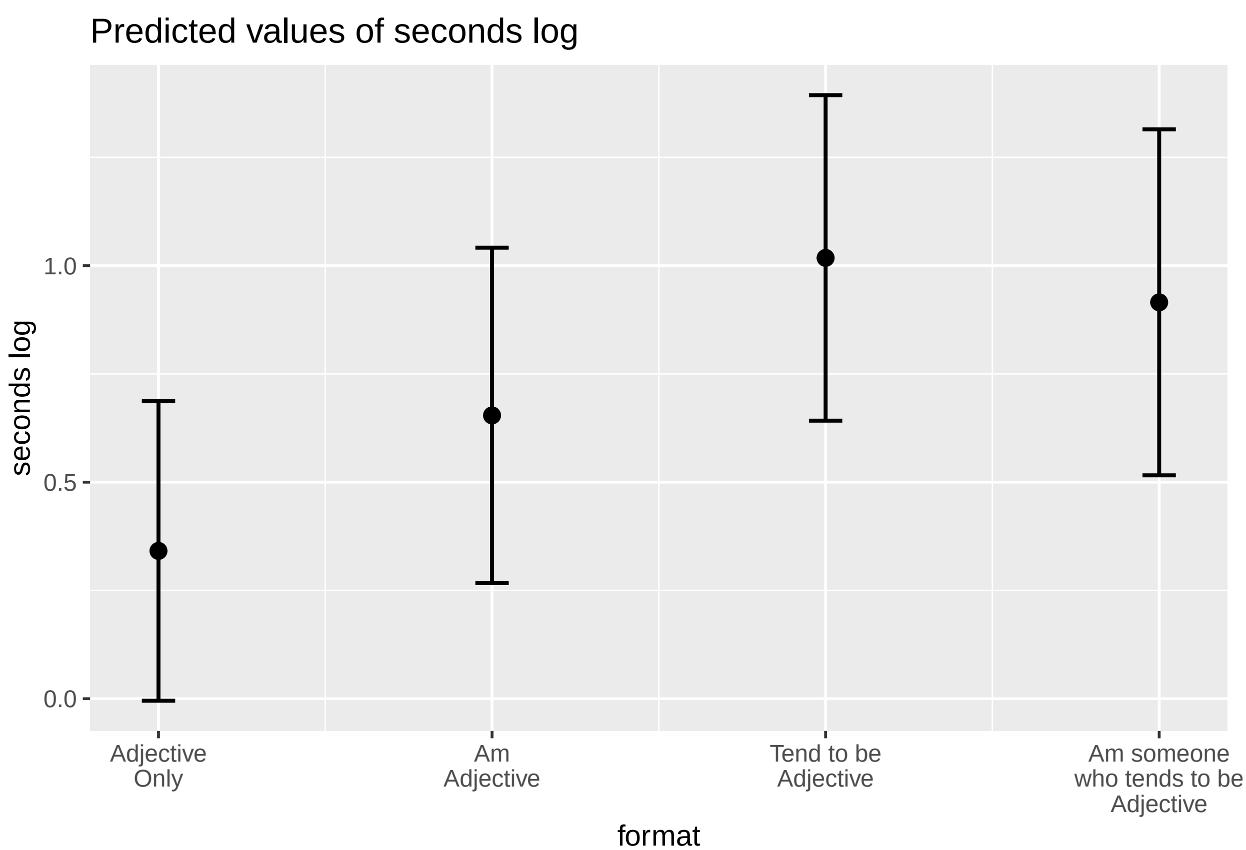


Figure 39: Average log-seconds to “adventurous” by format (Block 1 and Block 2)

```
## Formula: seconds_log ~ format * delayed_memory + (1 | proid)
##    Data: items_12
##
## 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
## formatAm 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
## formatAm 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
## formatAm 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
## formatAm someone\nwho tends to be\nAdjective:delayed_memory    0.02036
##                                                               df t value
## (Intercept)                                             51.00813   4.882
## formatAm\nAdjective                                   2020.78149  -0.977
## formatTend to be\nAdjective                           2012.77870   1.185
## formatAm someone\nwho tends to be\nAdjective          2021.63962   3.234
## delayed_memory                                          51.97799  -0.494
## formatAm\nAdjective:delayed_memory                    2017.90928   3.854
## formatTend to be\nAdjective:delayed_memory            2008.14522   1.014
## formatAm someone\nwho tends to be\nAdjective:delayed_memory 2021.99447  -0.609
##                                                       Pr(>|t|)
## (Intercept)                                           1.08e-05 ***
## formatAm\nAdjective                                    0.32877
## formatTend to be\nAdjective                            0.23612
## formatAm someone\nwho tends to be\nAdjective           0.00124 **
## delayed_memory                                         0.62371
## formatAm\nAdjective:delayed_memory                     0.00012 ***
## formatTend to be\nAdjective:delayed_memory             0.31077
## formatAm someone\nwho tends to be\nAdjective:delayed_memory  0.54247
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Correlation of Fixed Effects:
##              (Intr) frmtAA frTtbA frAswttbA dlyd_m frAA:_ fTtbA:
## frmtAmAdjct -0.370
## frmtTndtbAd -0.366  0.474
## frmtAswttbA -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
## frAswttbA:_  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
```

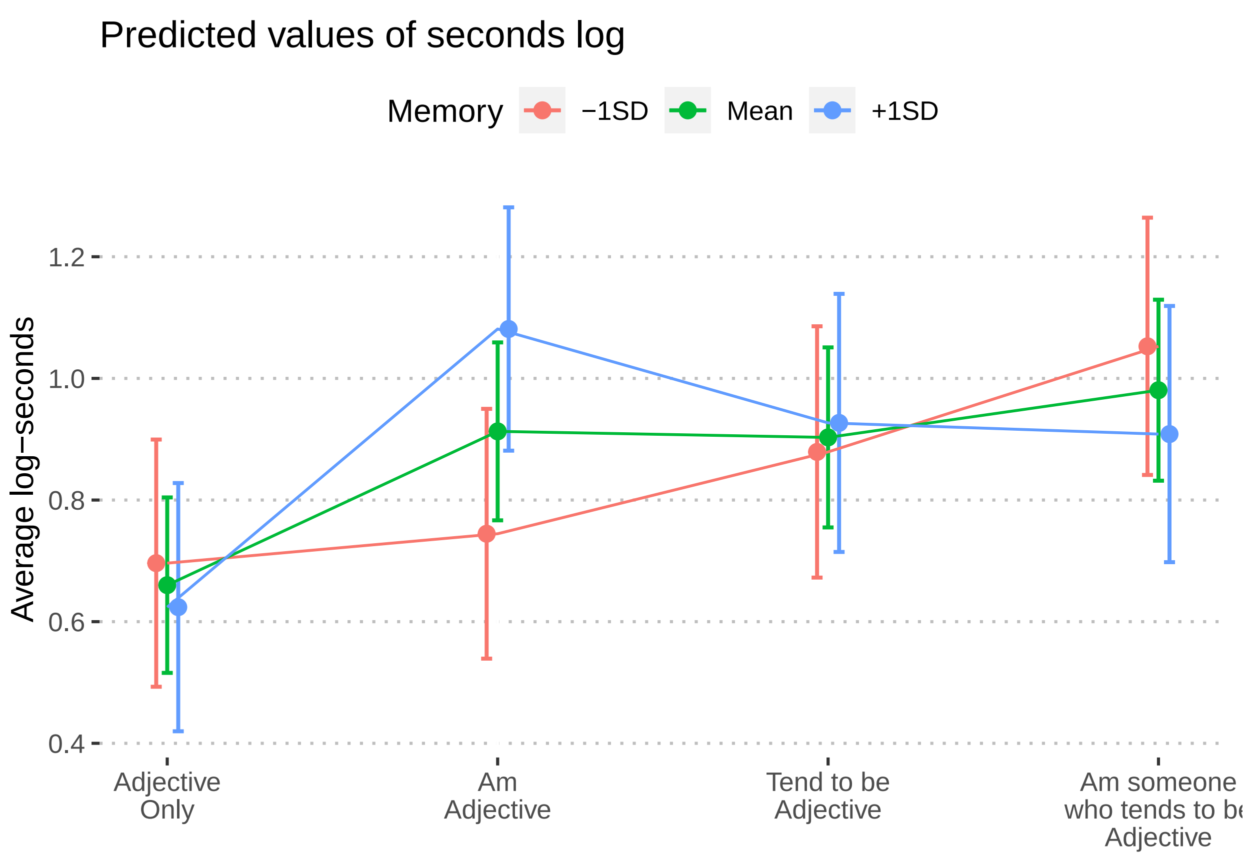


Figure 40: Predicted log-seconds on personality items by condition after controlling for delayed_memory.

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

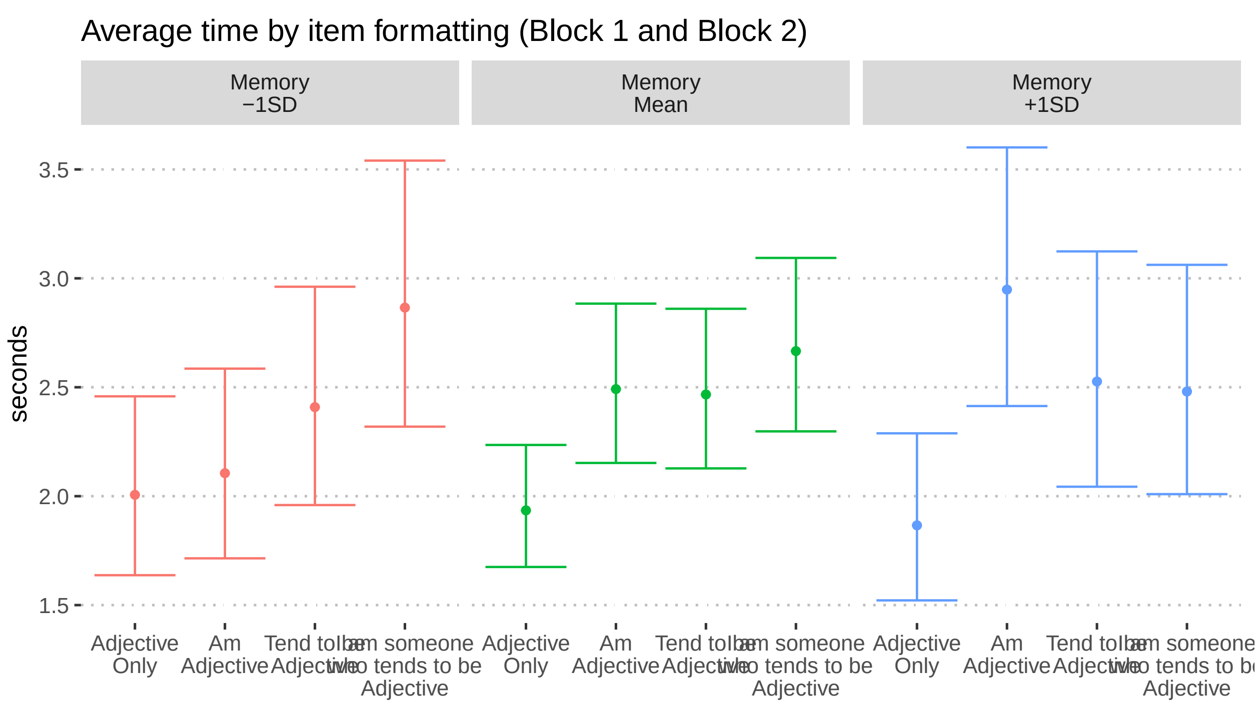


Figure 41: Predicted seconds on personality items by condition after controlling for delayed_memory.

5.3.1 One model for each adjective

```
mod_by_item_mem = items_12 %>%
  group_by(item) %>%
  nest() %>%
  mutate(mod = map(data, ~lmer(seconds_log~format*delayed_memory + (1|proid), data = .))) %>%
  mutate(aov = map(mod, anova))
```

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

```r
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) %>%
  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()
```

5.3.2 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"
```

5.3.3 Outgoing

```
outgoing_model_mem = items_12 %>%
  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]"))
```

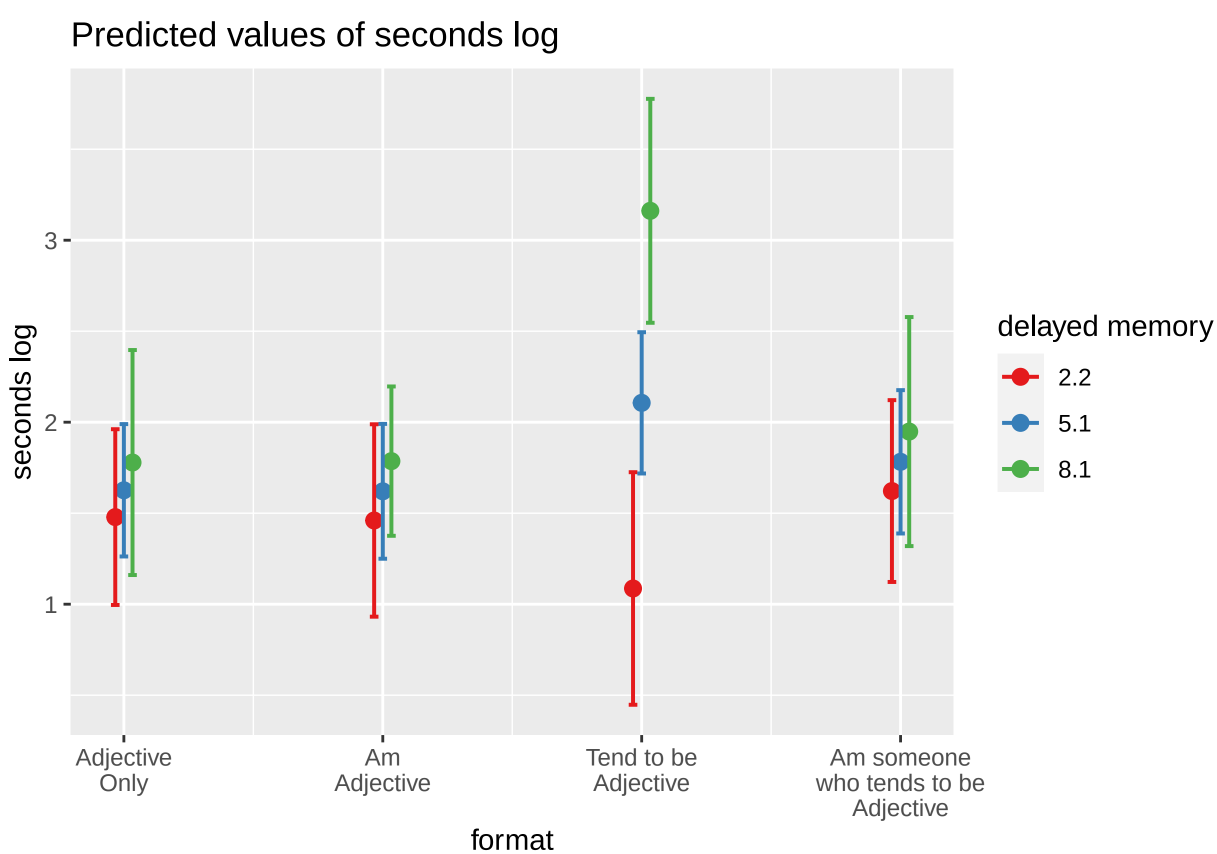


Figure 42: Average log-seconds to "outgoing" by format (Block 1 and Block 2)

5.3.4 Warm

```
warm_model_mem = items_12 %>%
  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]"))
```

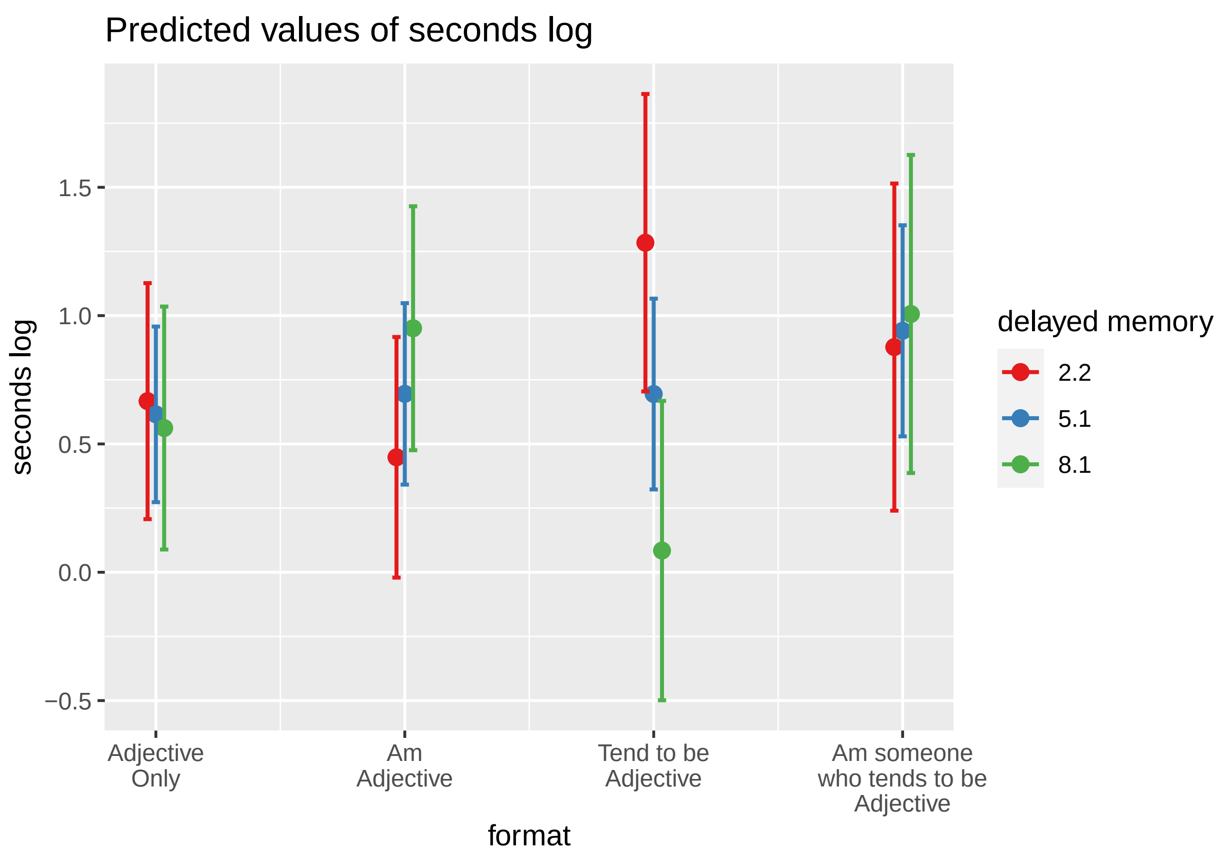


Figure 43: Average log-seconds to "warm" by format (Block 1 and Block 2)

5.3.5 Responsible

```
responsible_model_mem = items_12 %>%
  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]"))
```

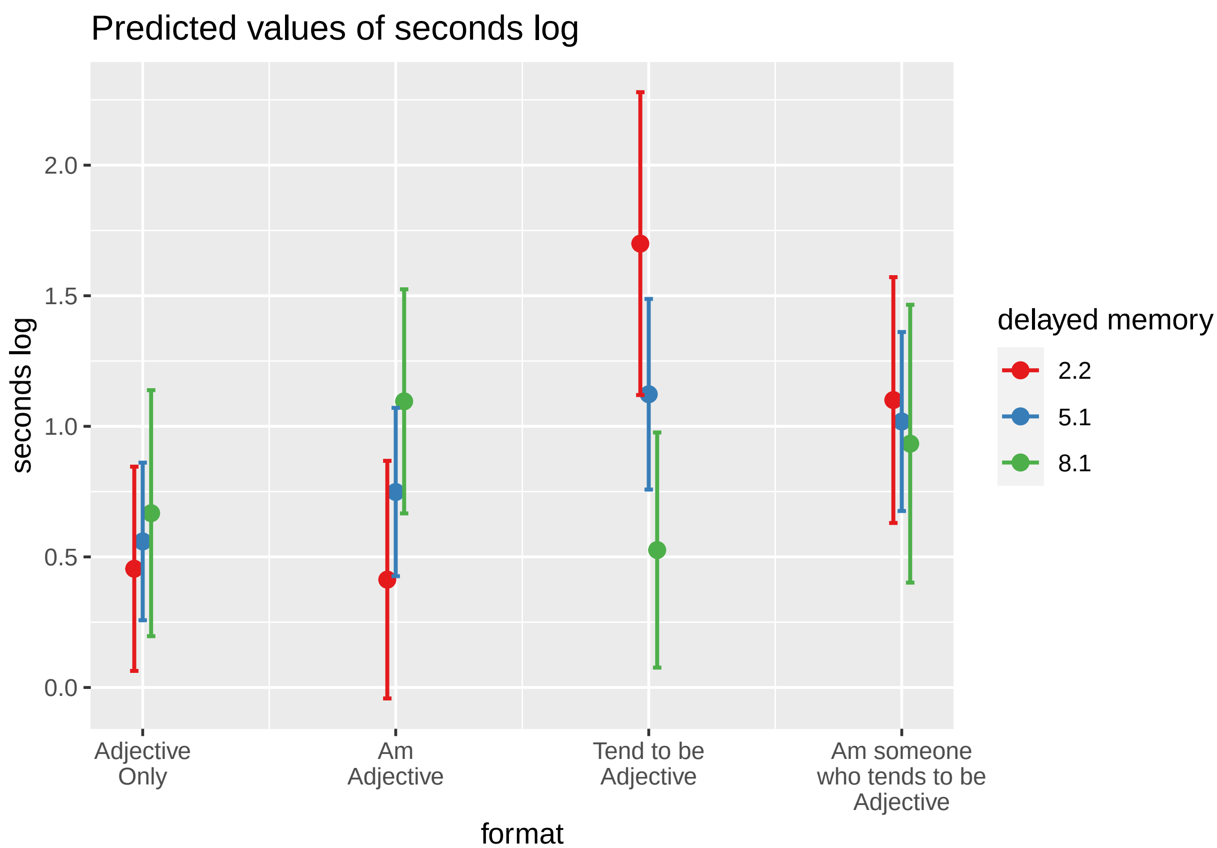


Figure 44: Average log-seconds to “responsible” by format (Block 1 and Block 2)

5.3.6 Imaginative

```
imaginative_model_mem = items_12 %>%
  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]"))
```

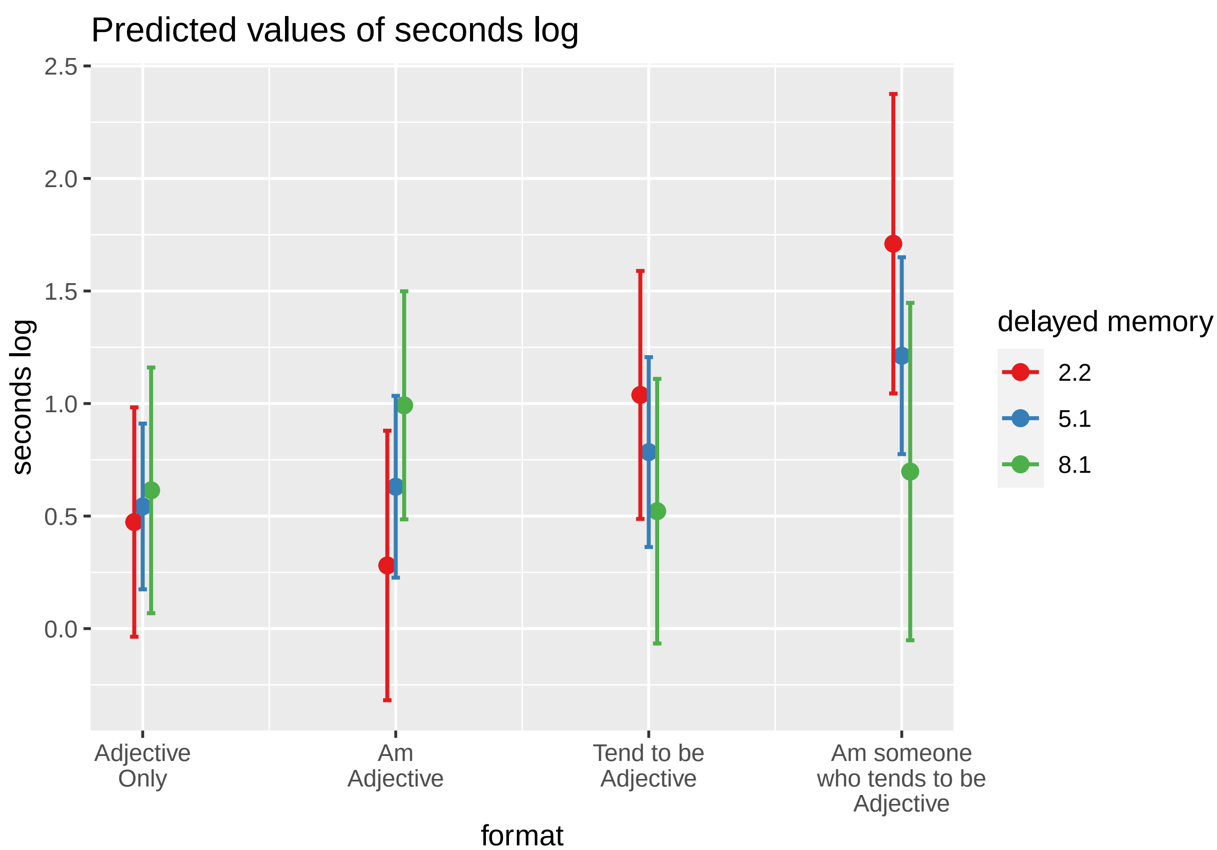


Figure 45: Average log-seconds to "imaginative" by format (Block 1 and Block 2)

5.3.7 Talkative

```
talkative_model_mem = items_12 %>%
  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]"))
```

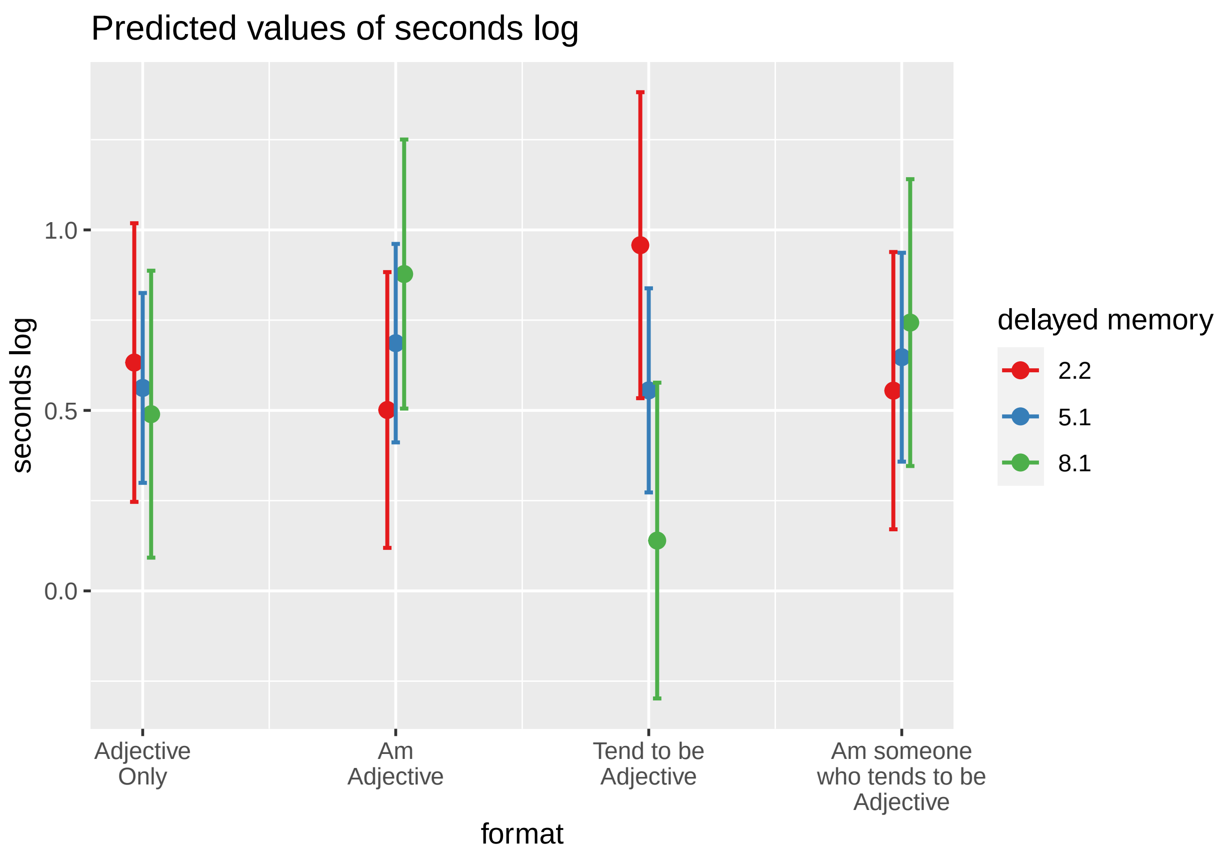


Figure 46: Average log-seconds to "talkative" by format (Block 1 and Block 2)

5.3.8 Sophisticated

```
sophisticated_model_mem = items_12 %>%
  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]"))
```

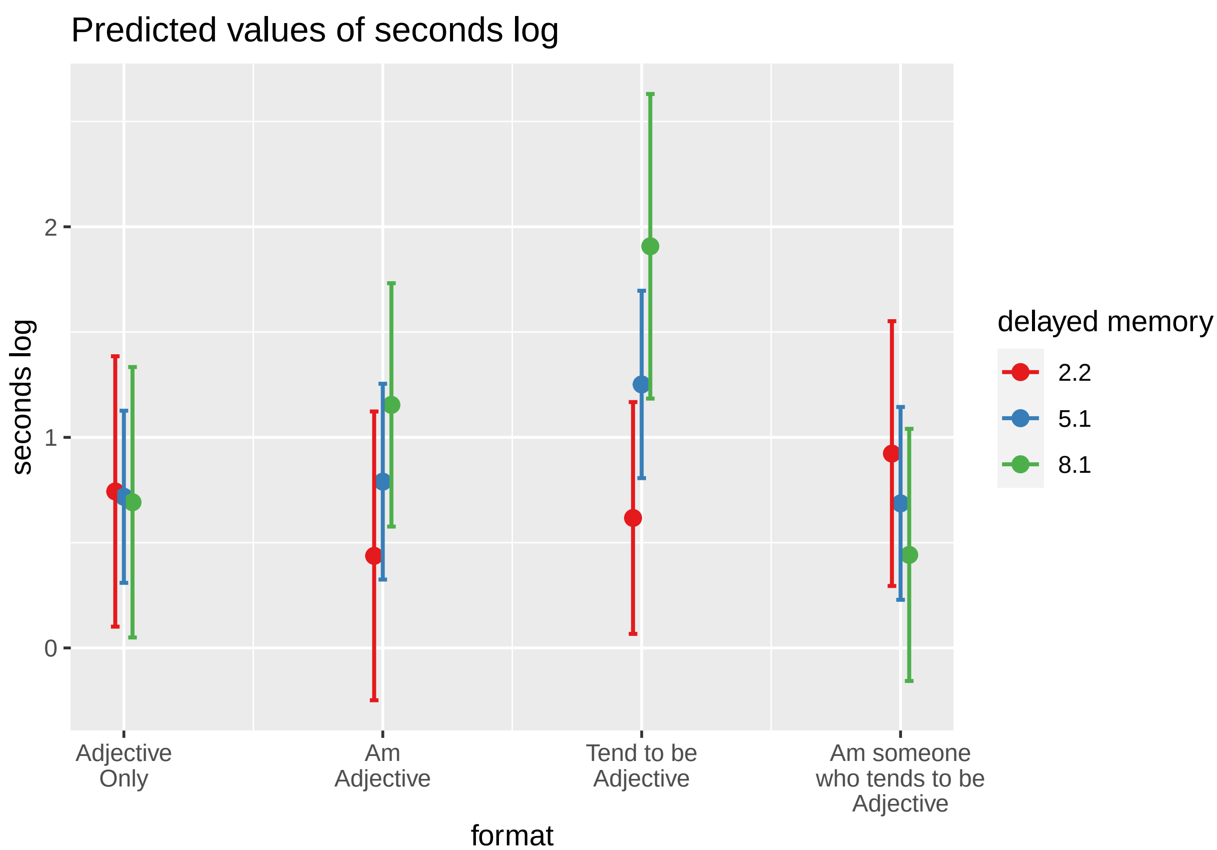


Figure 47: Average log-seconds to "sophisticated" by format (Block 1 and Block 2)

6 How does device type affect means and timing of responses?

6.1 Responses

6.1.1 Response by device

We used a multilevel model, nesting response within participant to account for dependence. Our primary predictor was device format. Here, we use data from blocks 1 and 2.

```
items_12 = items_df %>% filter(block %in% c("1","2"))
```

```
mod.response = lmer(response~devicetype + (1|proid),
                    data = items_12)
anova(mod.response)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##            Sum Sq Mean Sq NumDF DenDF F value Pr(>F)
## devicetype 3.1474  1.5737     2    32  1.0194 0.3722
```

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

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

```
means_by_group = items_12 %>%
  group_by(devicetype) %>%
  summarise(m = mean(response),
            s = sd(response))

items_12 %>%
  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()
```

6.1.2 Device by format

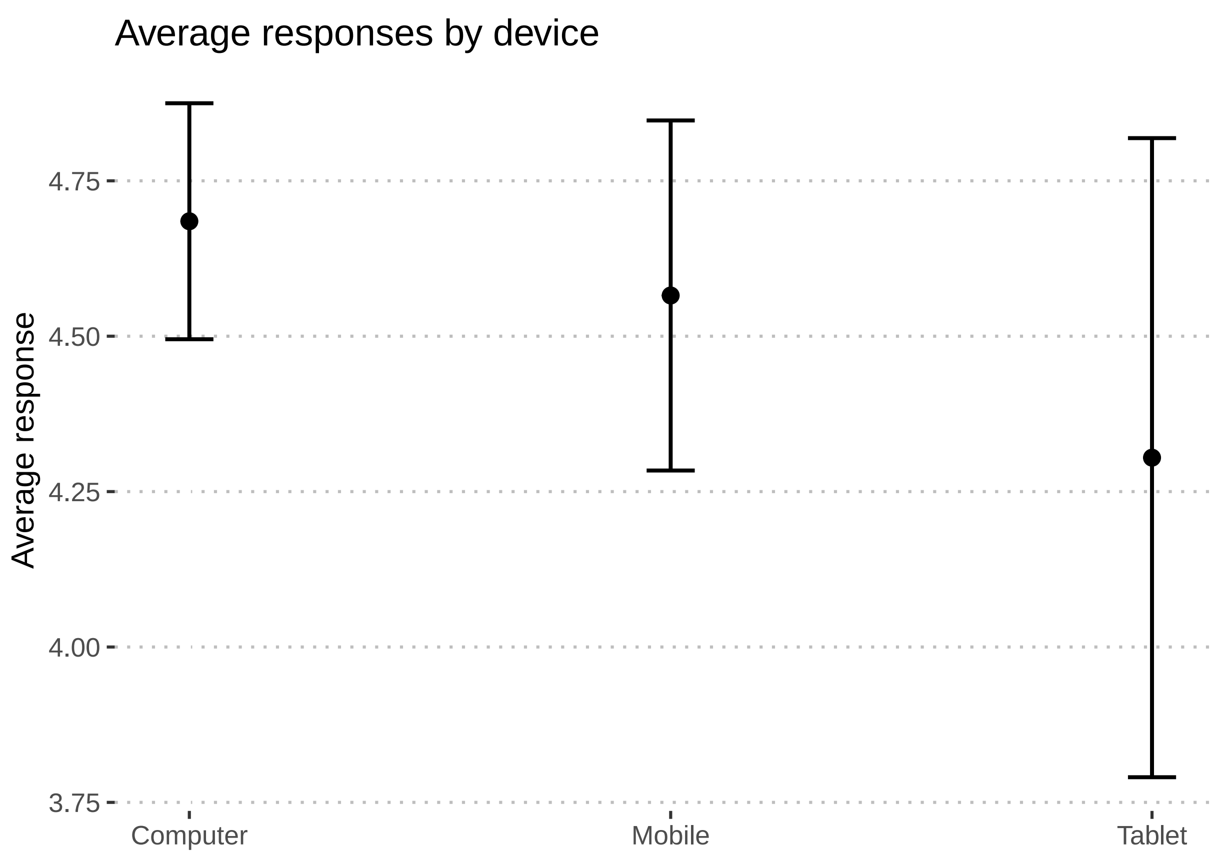


Figure 48: Predicted response on personality items by condition.

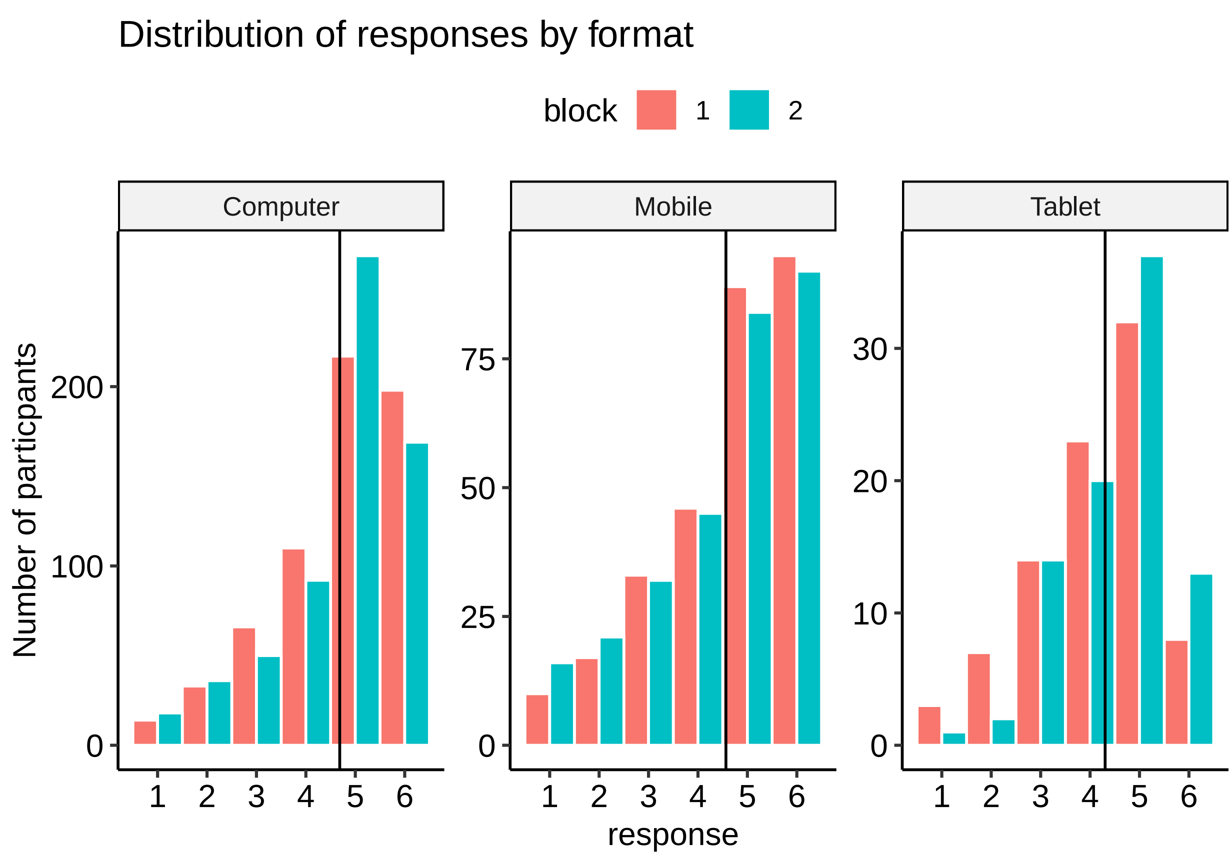


Figure 49: Distribution of responses by category

```r
mod.response2 = lmer(response~devicetype*format + (1|proid),
                     data = items_12)
anova(mod.response2)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##                    Sum Sq Mean Sq NumDF   DenDF F value Pr(>F)
## devicetype          1.8811 0.94055     2   32.87  0.6102 0.5493
## format              5.9073 1.96910     3 1960.37  1.2774 0.2805
## devicetype:format 14.9750 2.49584     6 1962.03  1.6191 0.1378
```

```r
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()
```

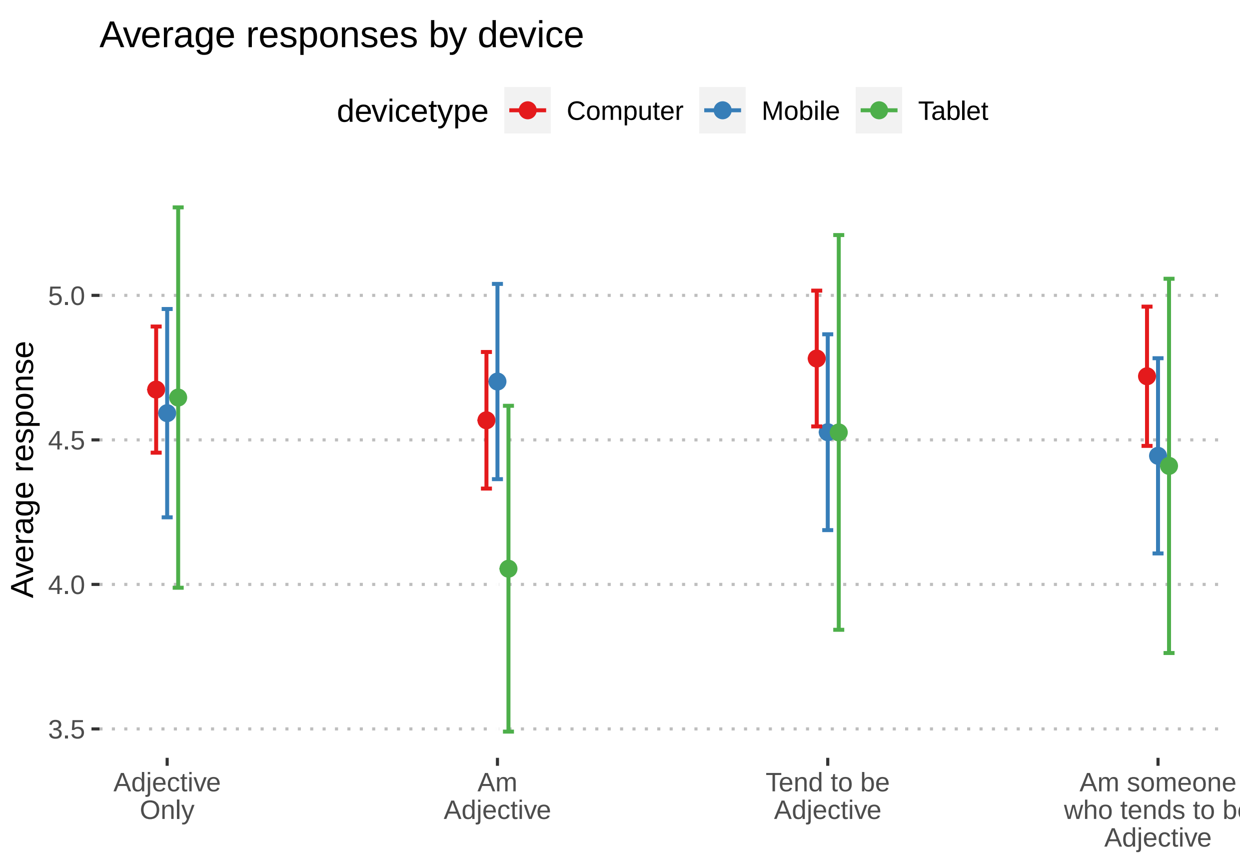


Figure 50: Predicted response on personality items by condition.

6.2 Timing

6.2.1 Timing by device

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

```
mod.timing = lmer(seconds_log~devicetype + (1|proid),
                  data = items_12)
anova(mod.timing)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##            Sum Sq Mean Sq NumDF DenDF F value Pr(>F)
## devicetype 2.7595  1.3798     2    32  2.0181 0.1495
```

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

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

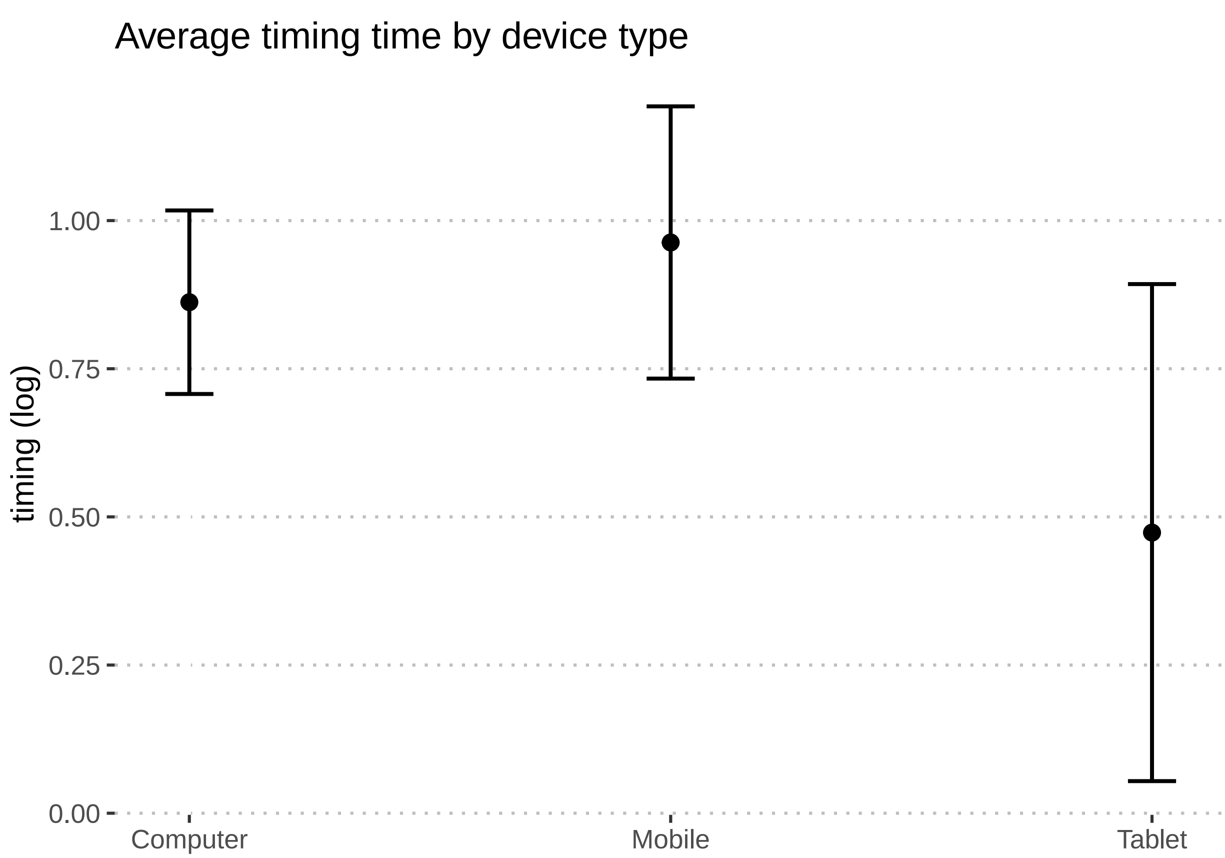


Figure 51: Predicted timing on personality items by condition.

```
means_by_group = items_12 %>%
  group_by(devicetype) %>%
  summarise(m = mean(timing),
            s = sd(timing))

items_12 %>%
  ggplot(aes(x = timing, 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 timing by format",
       x = "timing (logrithmic scale)") +
  theme_pubr()
```

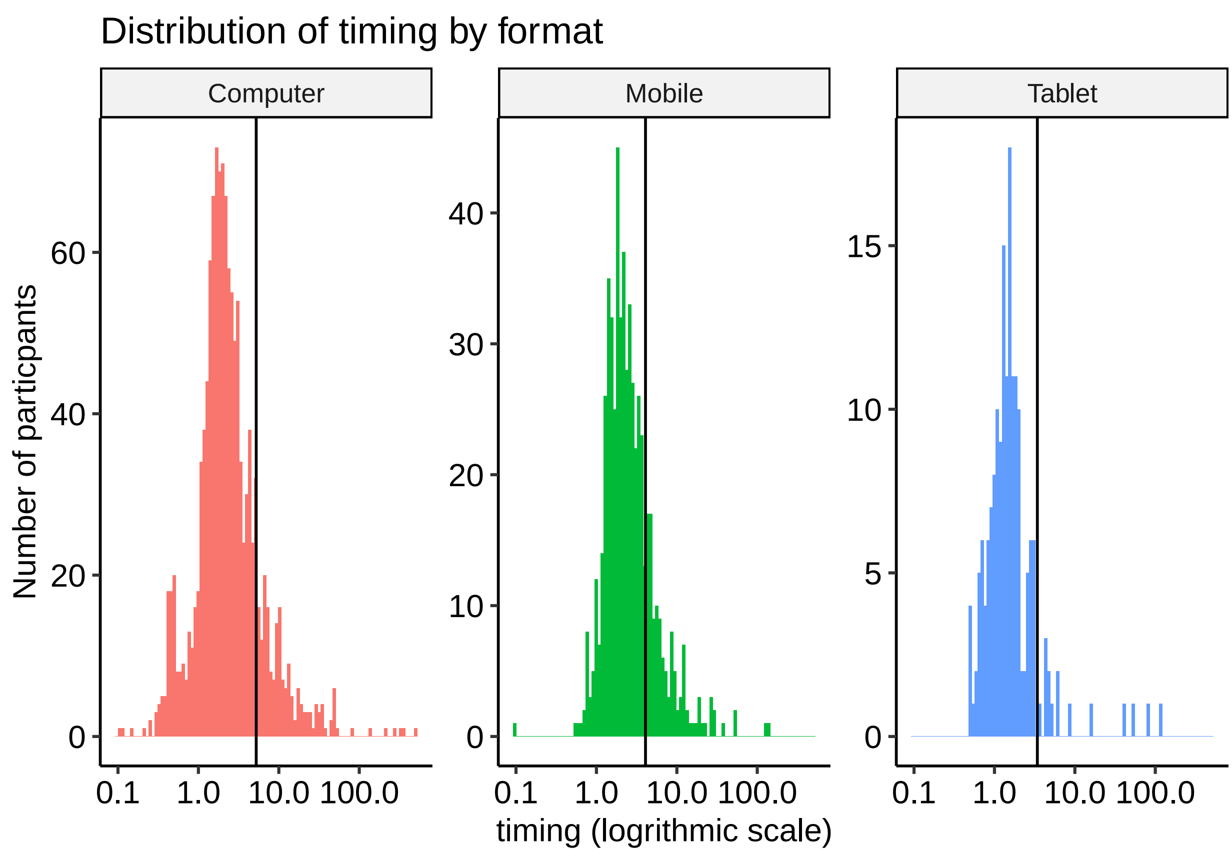


Figure 52: Distribution of secondss by category

6.2.2 Device by format

```
mod.timing2 = lmer(seconds_log~devicetype*format + (1|proid),
                   data = items_12)
anova(mod.timing2)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##                    Sum Sq Mean Sq NumDF  DenDF F value   Pr(>F)
## devicetype          3.2291  1.6146     2   32.3  2.4193 0.104943
## format             10.3707  3.4569     3 2004.2  5.1798 0.001447 **
## devicetype:format 13.3817  2.2303     6 2007.5  3.3418 0.002802 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

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

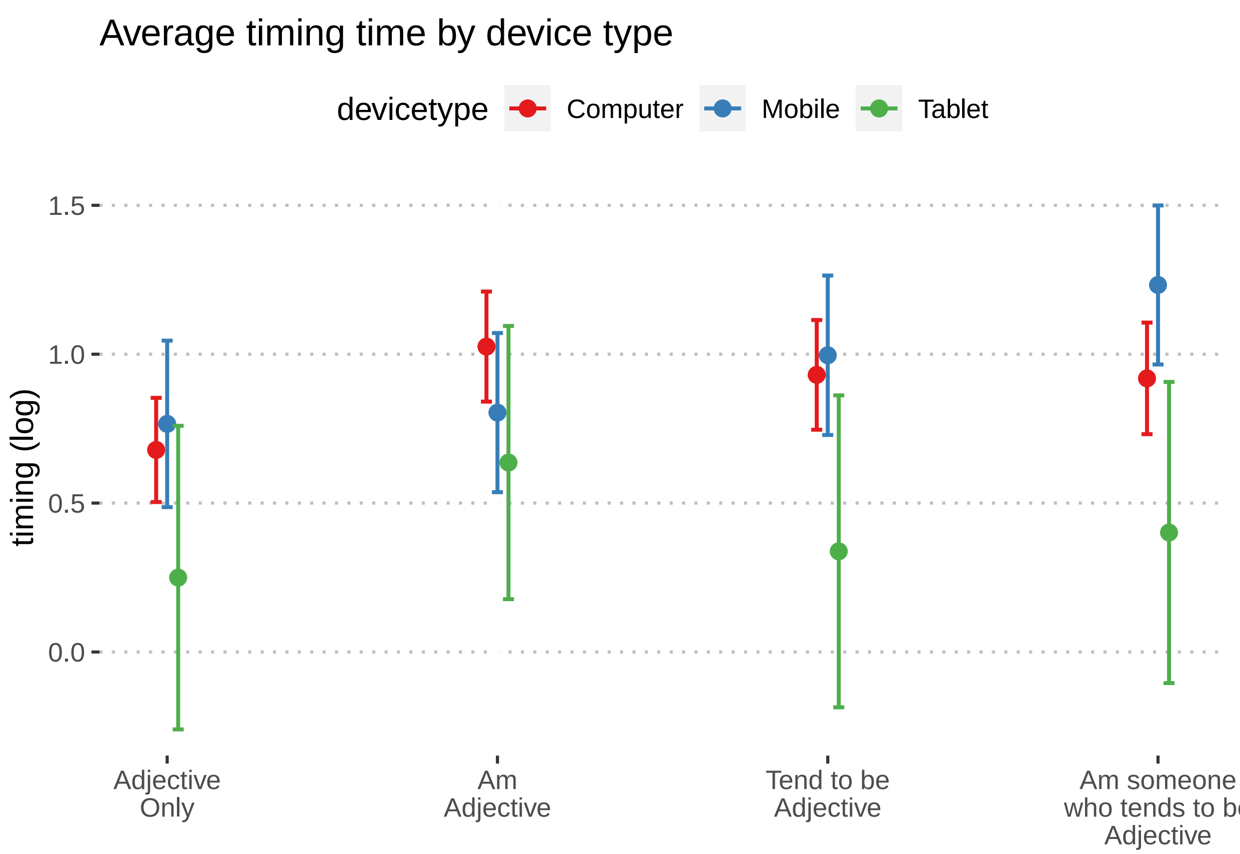


Figure 53: Predicted timing on personality items by condition.

7 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.

7.1 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.

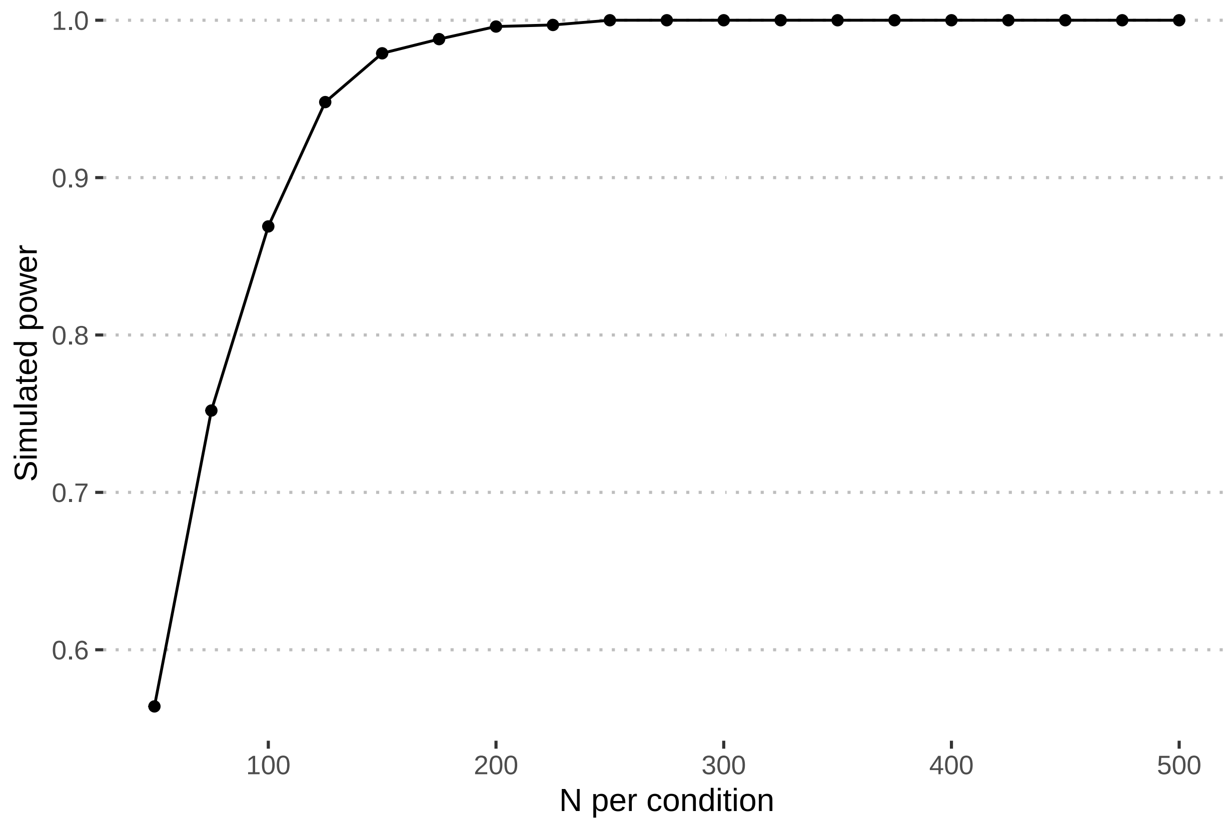
```r
# simulate at various sample sizes
# n = number per condition

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.

```r
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.

```r
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 150 participants per condition.

7.2 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,
```

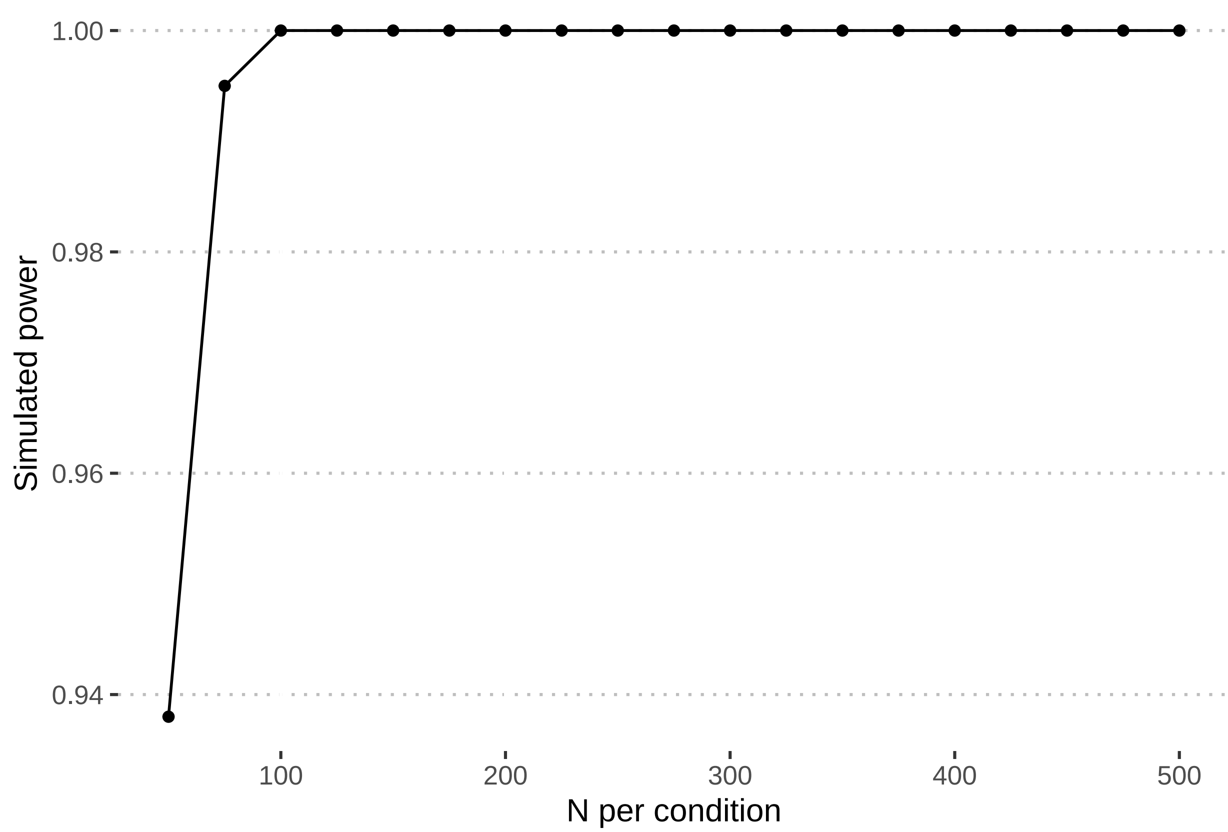
```
  power = 0
)
```

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 75 participants per condition.