Experiment Counterfactual Power Analysis

ZhaoBin Li, Scott Cheng-Hsin Yang, Tomas Folke, Patrick Shafto

Aug 3rd, 2022

```
# conda create -n r_chemicals -c conda-forge r-base r-jsonlite r-tidyverse r-ggplot2 r-lme4 r-afex r-fu
library(tictoc)
library(testthat)
library(binom)
library(viridis)
```

```
## Loading required package: viridisLite
```

```
library(sjPlot)
library(furrr)
```

```
## Loading required package: future
```

```
library(afex)
```

```
## Loading required package: lme4
```

```
## Loading required package: Matrix
```

```
## ************
## Welcome to afex. For support visit: http://afex.singmann.science/
```

```
## - Functions for ANOVAs: aov_car(), aov_ez(), and aov_4()
## - Methods for calculating p-values with mixed(): 'S', 'KR', 'LRT', and 'PB'
## - 'afex_aov' and 'mixed' objects can be passed to emmeans() for follow-up tests
## - NEWS: emmeans() for ANOVA models now uses model = 'multivariate' as default.
## - Get and set global package options with: afex_options()
## - Set orthogonal sum-to-zero contrasts globally: set_sum_contrasts()
## - For example analyses see: browseVignettes("afex")
## ************
```

```
##
## Attaching package: 'afex'
```

```
## The following object is masked from 'package:lme4':
##
##     lmer
```

```
library(lme4)
library(jsonlite)
library(ggplot2)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
```

```
## v tibble  3.1.7     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

```
## v purrr   0.3.4
## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x readr::edition_get()   masks testthat::edition_get()
## x tidyr::expand()        masks Matrix::expand()
## x dplyr::filter()        masks stats::filter()
## x purrr::flatten()       masks jsonlite::flatten()
## x purrr::is_null()       masks testthat::is_null()
## x dplyr::lag()           masks stats::lag()
## x readr::local_edition() masks testthat::local_edition()
## x dplyr::matches()       masks tidyr::matches(), testthat::matches()
## x tidyr::pack()          masks Matrix::pack()
## x tidyr::unpack()        masks Matrix::unpack()
```

```
MIN_AVG_RT <- 0 # ms

# Wrangle --------------------------------------------------------------


# column names in https://psiturk.readthedocs.io/en/stable/command_line.html?highlight=trialdata.csv#do
df <-
  # read_csv("experiments/chemicals/invariance/psiturk/trialdata.csv",
  read_csv("invariance/psiturk/trialdata.csv",
    col_names = c("id", "trialNum", "time", "trialData")
  ) |>
  arrange(time) |> # arrange by time
  glimpse()
```

```
## Rows: 57 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): id, trialData
## dbl (2): trialNum, time
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
## Rows: 57
## Columns: 4
## $ id        <chr> "debugqUNLv:debugVeQp1", "debug8pmDF:debugGHcXK", "ABYGLITD9~
## $ trialNum  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ time      <dbl> 1.657292e+12, 1.657292e+12, 1.657293e+12, 1.657293e+12, 1.65~
## $ trialData <chr> "{\"trials\": [{\"rt\": 837, \"stimulus\": \"<p class=\\\"im~
```

```
# analyse 1 participant
# df <- df |> filter(id %in% c("debugQfDOW:debug1rA1P")) |> glimpse()

# exclude by time
df <- df |>
  filter(time >= 1657293679472) |>
  glimpse()
```

```
## Rows: 46
## Columns: 4
## $ id        <chr> "A1XT5X6QHAVTTK:3LOKT67Y8FLKM9JS52D62DGPWHQYS8", "A3J8UC84NM~
## $ trialNum  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ time      <dbl> 1.657294e+12, 1.657294e+12, 1.657294e+12, 1.657294e+12, 1.65~
```

```
## $ trialData <chr> "{\"trials\": [{\"width\": 1364, \"height\": 625, \"webaudio~
```

```
# remove debug attempts
# df <- df |>
#   filter(!str_detect(id, "debug")) |>
#   glimpse()

# check no duplicate ids
expect_equal(nrow(df |>
  group_by(id) |>
  filter(n() > 1)), 0)

# anonymize id
df <- df |>
  mutate(id = fct_anon(factor(id))) |>
  glimpse()
```

```
## Rows: 46
## Columns: 4
## $ id        <fct> 29, 43, 20, 41, 17, 11, 05, 32, 39, 14, 21, 34, 02, 16, 18, ~
## $ trialNum  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ time      <dbl> 1.657294e+12, 1.657294e+12, 1.657294e+12, 1.657294e+12, 1.65~
## $ trialData <chr> "{\"trials\": [{\"width\": 1364, \"height\": 625, \"webaudio~
```

```
# get nested JSON trial data
df <- df |>
  rowwise() |>
  mutate(trialData = fromJSON(trialData)) |>
  unnest(trialData) |>
  glimpse()
```

```
## Rows: 2,687
## Columns: 39
## $ id               <fct> 29, 29, 29, 29, 29, 29, 29, 29, 29, 29, 29, 29, 29, 2~
## $ trialNum         <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ time             <dbl> 1.657294e+12, 1.657294e+12, 1.657294e+12, 1.657294e+1~
## $ width            <int> 1364, 1364, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ height           <int> 625, 625, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ webaudio         <lgl> TRUE, TRUE, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ browser          <chr> "chrome", "chrome", NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ browser_version  <chr> "103.0.5060", "103.0.5060", NA, NA, NA, NA, NA, NA, N~
## $ mobile           <lgl> FALSE, FALSE, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ os               <chr> "Windows 10", "Windows 10", NA, NA, NA, NA, NA, NA, N~
## $ fullscreen       <lgl> TRUE, TRUE, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ vsync_rate       <dbl> 60.18, 59.94, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ webcam           <lgl> FALSE, FALSE, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ microphone       <lgl> FALSE, FALSE, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ trial_type       <chr> "browser-check", "browser-check", "preload", "survey-~
## $ trial_index      <int> 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,~
## $ time_elapsed     <int> 1015, 2021, 8125, 13191, 16000, 76653, 85653, 91645, ~
## $ internal_node_id <chr> "0.0-0.0", "0.0-1.0", "0.0-2.0", "0.0-3.0", "0.0-4.0"~
## $ condition        <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
## $ robotCondition   <chr> "opp", "opp", "opp", "opp", "opp", "opp", "opp", "opp~
## $ success          <lgl> NA, NA, TRUE, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ timeout          <lgl> NA, NA, FALSE, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ failed_images    <list> <NULL>, <NULL>, [], <NULL>, <NULL>, <NULL>, <NULL>, ~
```

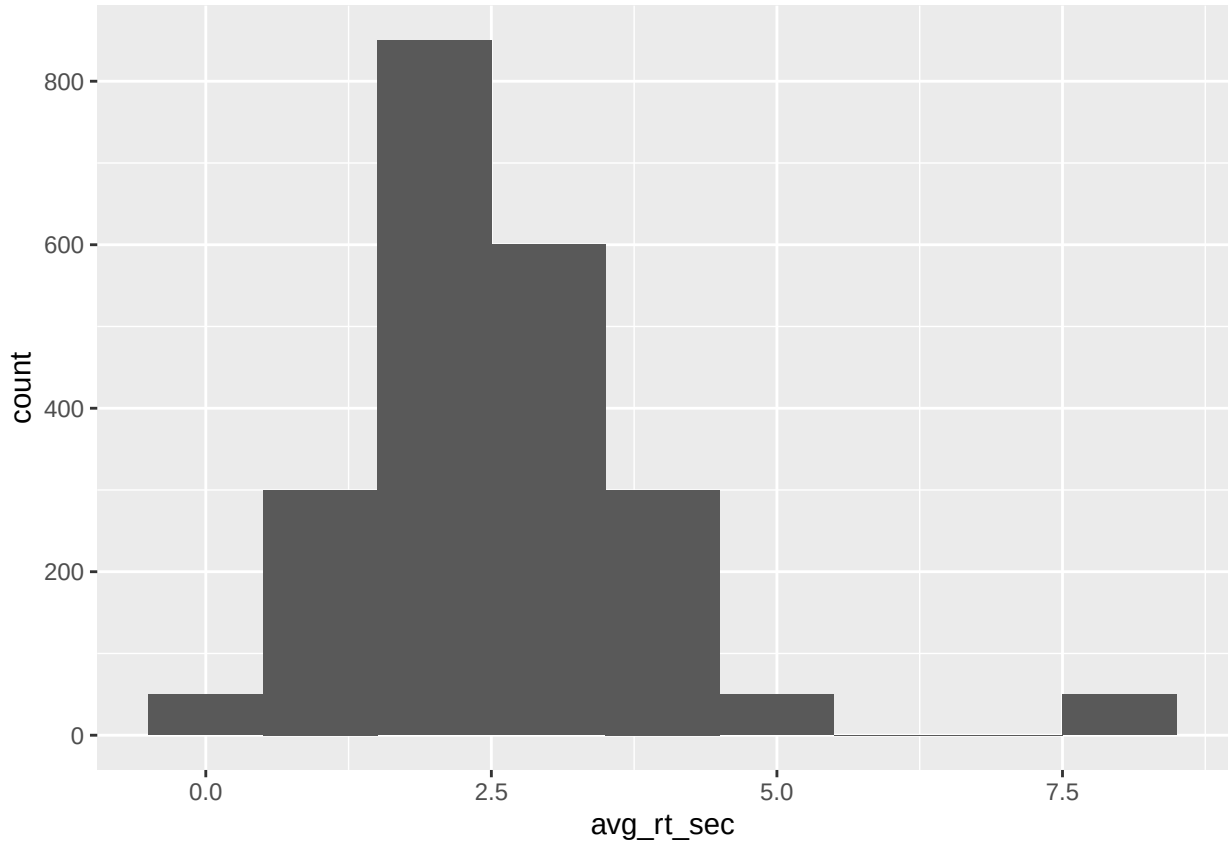
```
## $ failed_audio     <list> <NULL>, <NULL>, [], <NULL>, <NULL>, <NULL>, <NULL>, ~
## $ failed_video     <list> <NULL>, <NULL>, [], <NULL>, <NULL>, <NULL>, <NULL>, ~
## $ rt               <int> NA, NA, NA, 5064, 2803, 60653, 8997, 5991, 1966, 1599~
## $ response         <list> <NULL>, <NULL>, <NULL>, ["Male"], ["29"], <NULL>, 1,~
## $ question_order   <list> <NULL>, <NULL>, <NULL>, 0, <NULL>, <NULL>, <NULL>, <~
## $ view_history     <list> <NULL>, <NULL>, <NULL>, <NULL>, <NULL>, [<data.frame~
## $ stimulus         <chr> NA, NA, NA, NA, NA, NA, "<p><span>Aike <i class=\"fa-~
## $ trial            <chr> NA, NA, NA, NA, NA, NA, "practice", "practice", "prac~
## $ salLabel         <chr> NA, NA, NA, NA, NA, NA, "safe", "toxic", "safe", NA, ~
## $ promptLabel      <chr> NA, NA, NA, NA, NA, NA, "toxic", "safe", "toxic", NA,~
## $ imgNum           <int> NA, NA, NA, NA, NA, NA, 2, 3, 4, NA, 49, 13, 28, 21, ~
## $ buttonOrder      <list> <NULL>, <NULL>, <NULL>, <NULL>, <NULL>, <NULL>, <"op~
## $ sal              <chr> NA, NA, NA, NA, NA, NA, "/static/stimuli/ax_sal_2.svg~
## $ buttonPaths      <list> <NULL>, <NULL>, <NULL>, <NULL>, <NULL>, <NULL>, <"/s~
## $ responseCompare  <chr> NA, NA, NA, NA, NA, NA, "same", "same", "same", NA, "~
## $ labelCompare     <chr> NA, NA, NA, NA, NA, NA, "opp", "opp", "opp", NA, "opp~
```

```
# save data
# df |> write_csv("experiments/chemicals/invariance/wrangled_data.csv")
df |> write_csv("invariance/wrangled_data.csv")

# restrict to main trials
df <- df |>
  filter(trial == "results") |>
  glimpse()
```

```
## Rows: 2,200
## Columns: 39
## $ id               <fct> 29, 29, 29, 29, 29, 29, 29, 29, 29, 29, 29, 29, 29, 2~
## $ trialNum         <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ time             <dbl> 1.657294e+12, 1.657294e+12, 1.657294e+12, 1.657294e+1~
## $ width            <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ height           <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ webaudio         <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ browser          <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ browser_version  <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ mobile           <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ os               <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ fullscreen       <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ vsync_rate       <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ webcam           <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ microphone       <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ trial_type       <chr> "html-button-response", "html-button-response", "html~
## $ trial_index      <int> 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 2~
## $ time_elapsed     <int> 96763, 99117, 100716, 104213, 106828, 109356, 111373,~
## $ internal_node_id <chr> "0.0-7.0-0.0", "0.0-7.0-0.1", "0.0-7.0-0.2", "0.0-7.0~
## $ condition        <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
## $ robotCondition   <chr> "opp", "opp", "opp", "opp", "opp", "opp", "opp", "opp~
## $ success          <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ timeout          <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ failed_images    <list> <NULL>, <NULL>, <NULL>, <NULL>, <NULL>, <NULL>, <NUL~
## $ failed_audio     <list> <NULL>, <NULL>, <NULL>, <NULL>, <NULL>, <NULL>, <NUL~
## $ failed_video     <list> <NULL>, <NULL>, <NULL>, <NULL>, <NULL>, <NULL>, <NUL~
## $ rt               <int> 1550, 2351, 1598, 3496, 2612, 2528, 2015, 1502, 1495,~
## $ response         <list> 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0~
```

```
## $ question_order   <list> <NULL>, <NULL>, <NULL>, <NULL>, <NULL>, <NULL>, <NUL~
## $ view_history     <list> <NULL>, <NULL>, <NULL>, <NULL>, <NULL>, <NULL>, <NUL~
## $ stimulus         <chr> "<p><span>Aike <i class=\"fa-solid fa-robot robot1  i~
## $ trial            <chr> "results", "results", "results", "results", "results"~
## $ salLabel         <chr> "safe", "safe", "toxic", "toxic", "toxic", "toxic", "~
## $ promptLabel      <chr> "toxic", "toxic", "safe", "safe", "safe", "safe", "sa~
## $ imgNum           <int> 49, 13, 28, 21, 18, 24, 37, 30, 36, 35, 45, 25, 16, 5~
## $ buttonOrder      <list> <"same", "opp">, <"same", "opp">, <"opp", "same">, <~
## $ sal              <chr> "/static/stimuli/ax_sal_49.svg", "/static/stimuli/ax_~
## $ buttonPaths      <list> <"/static/stimuli/ax_sal_49.svg", "/static/stimuli/a~
## $ responseCompare  <chr> "same", "opp", "opp", "same", "same", "same", "same",~
## $ labelCompare     <chr> "opp", "opp", "opp", "opp", "opp", "opp", "opp", "opp~
```

```r
# restrict to practice trials
# df <- df |>
#   filter(trial == "practice") |>
#   glimpse()

# get mean rt
df |>
  group_by(id) |>
  mutate(avg_rt_sec = mean(rt) / 1000) |>
  ggplot(aes(avg_rt_sec)) +
  geom_histogram(binwidth = 1)
```




```r
# exclude participants with mean rt < MIN_AVG_RT
df <- df |>
  group_by(id) |>
```

```
  filter(mean(rt) > MIN_AVG_RT) |>
  ungroup()

# Visualize ------------------------------------------------------------------
df <- df |>
  rename(img = imgNum) |>
  select(id, img, responseCompare, robotCondition, labelCompare) |>
  glimpse()
```

```
## Rows: 2,200
## Columns: 5
## $ id              <fct> 29, 29, 29, 29, 29, 29, 29, 29, 29, 29, 29, 29, 29, 29~
## $ img             <int> 49, 13, 28, 21, 18, 24, 37, 30, 36, 35, 45, 25, 16, 51~
## $ responseCompare <chr> "same", "opp", "opp", "same", "same", "same", "same", ~
## $ robotCondition  <chr> "opp", "opp", "opp", "opp", "opp", "opp", "opp", "opp"~
## $ labelCompare    <chr> "opp", "opp", "opp", "opp", "opp", "opp", "opp", "opp"~
```

```
df <- df |>
  mutate(
    responseCompare = fct_relevel(factor(responseCompare), ref = "same"),
    robotCondition = fct_relevel(factor(robotCondition), ref = "same"),
    labelCompare = fct_relevel(factor(labelCompare), ref = "same"),
  ) |>
  glimpse()
```

```
## Rows: 2,200
## Columns: 5
## $ id              <fct> 29, 29, 29, 29, 29, 29, 29, 29, 29, 29, 29, 29, 29, 29~
## $ img             <int> 49, 13, 28, 21, 18, 24, 37, 30, 36, 35, 45, 25, 16, 51~
## $ responseCompare <fct> same, opp, opp, same, same, same, same, same, same, sa~
## $ robotCondition  <fct> opp, opp, opp, opp, opp, opp, opp, opp, opp, opp, opp,~
## $ labelCompare    <fct> opp, opp, opp, opp, opp, opp, opp, opp, opp, opp, opp,~
```

```
df |>
  group_by(robotCondition, labelCompare, responseCompare) |>
  summarise(n())
```

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

```
## # A tibble: 8 x 4
## # Groups:   robotCondition, labelCompare [4]
##   robotCondition labelCompare responseCompare `n()`
##   <fct>          <fct>        <fct>           <int>
## 1 same           same         same              465
## 2 same           same         opp               235
## 3 opp            opp          same              410
## 4 opp            opp          opp               340
## 5 random         same         same              227
## 6 random         same         opp               132
## 7 random         opp          same              188
## 8 random         opp          opp               203
```

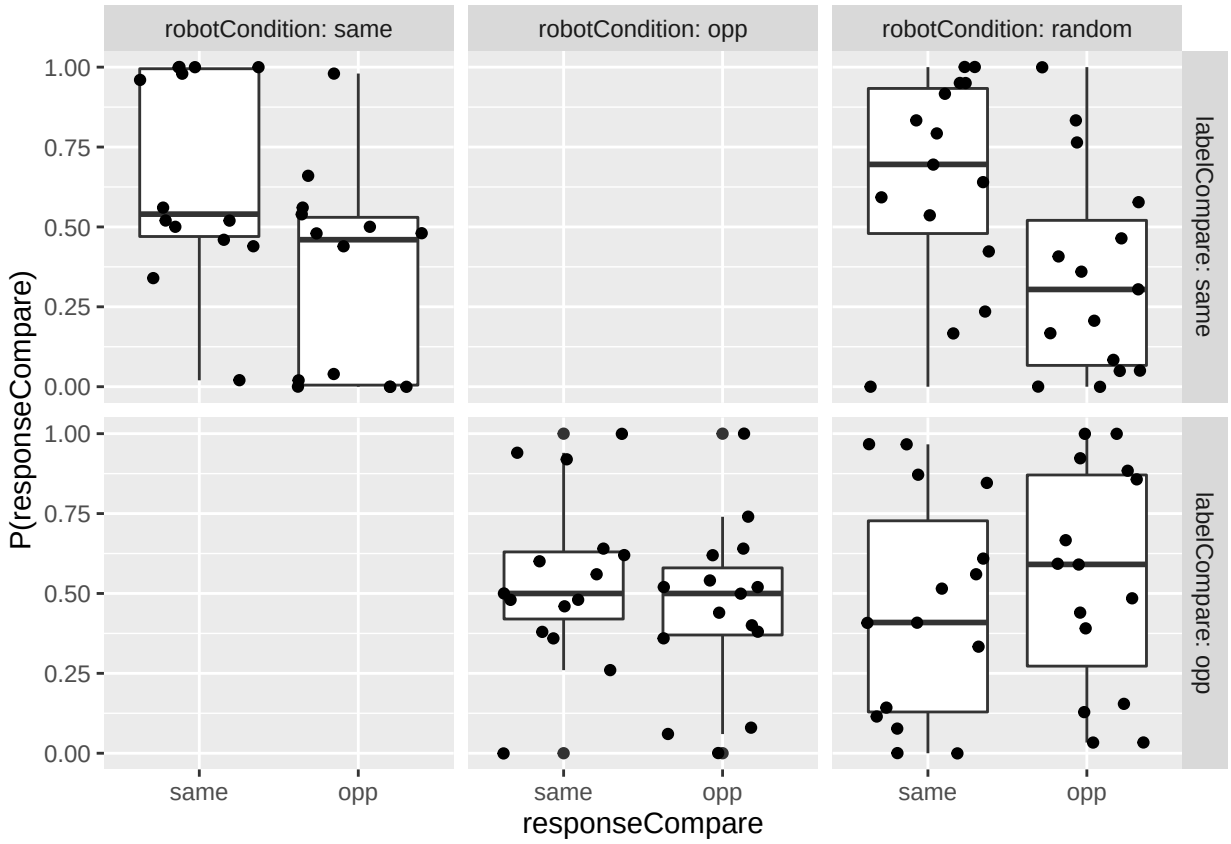
```
df |>
  group_by(robotCondition, labelCompare, id, responseCompare) |>
  summarise(n = n()) |>
```

```
  group_by(robotCondition, labelCompare) |>
  complete(id, responseCompare, fill = list(n = 0)) |>
  group_by(robotCondition, labelCompare, id) |>
  mutate(p = n / sum(n)) |>
  ggplot(aes(responseCompare, p)) +
  geom_boxplot() +
  geom_jitter() +
  facet_grid(vars(labelCompare), vars(robotCondition), labeller = label_both) +
  labs(y = "P(responseCompare)")
```

```
## `summarise()` has grouped output by 'robotCondition', 'labelCompare', 'id'. You
## can override using the `.groups` argument.
```

```
## Warning: Removed 250 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 250 rows containing missing values (geom_point).
```




```
# Analyses ------------------------------------------------------------------
df <- df |>
  mutate(
    choice = ifelse(responseCompare == "same", 0, 1),
    robot = fct_relevel(factor(robotCondition), ref = "same"),
    label = ifelse(labelCompare == "same", 0, 1)
  ) |>
  glimpse()
```

```
## Rows: 2,200
## Columns: 8
```

```
## $ id              <fct> 29, 29, 29, 29, 29, 29, 29, 29, 29, 29, 29, 29, 29, 29~
## $ img             <int> 49, 13, 28, 21, 18, 24, 37, 30, 36, 35, 45, 25, 16, 51~
## $ responseCompare <fct> same, opp, opp, same, same, same, same, same, same, sa~
## $ robotCondition  <fct> opp, opp, opp, opp, opp, opp, opp, opp, opp, opp, opp,~
## $ labelCompare    <fct> opp, opp, opp, opp, opp, opp, opp, opp, opp, opp, opp,~
## $ choice          <dbl> 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, ~
## $ robot           <fct> opp, opp, opp, opp, opp, opp, opp, opp, opp, opp, opp,~
## $ label           <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
```

```
# no need to add interaction with deterministic robots
model_glm <- glm(choice ~ robot + label, data = df, family = binomial)
summary(model_glm)
```

```
##
## Call:
## glm(formula = choice ~ robot + label, family = binomial, data = df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.2102  -1.0990  -0.9045   1.2579   1.4775
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.68245    0.08004  -8.527  < 2e-16 ***
## robotopp    -0.12367    0.18442  -0.671    0.502
## robotrandom  0.14030    0.13560   1.035    0.301
## label        0.61891    0.14908   4.151  3.3e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2983.9  on 2199  degrees of freedom
## Residual deviance: 2940.3  on 2196  degrees of freedom
## AIC: 2948.3
##
## Number of Fisher Scoring iterations: 4
```

```
# use (1 + label | id) since robot between id
# add (* | img) gives singular error because no variance
model_glmer <- glmer(choice ~ robot + label + (label | id), data = df, family = binomial)
summary(model_glmer)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: choice ~ robot + label + (label | id)
##    Data: df
##
##      AIC      BIC   logLik deviance df.resid
##   2162.9   2202.8  -1074.5   2148.9     2193
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -5.2653 -0.7520 -0.0937  0.7358  6.1791
```

```
##
## Random effects:
##  Groups Name        Variance Std.Dev. Corr
##  id     (Intercept) 6.849    2.617
##         label       4.783    2.187    -0.64
## Number of obs: 2200, groups:  id, 44
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.7512     0.7373  -2.375   0.0175 *
## robotopp      0.1334     1.1068   0.121   0.9041
## robotrandom   0.7646     1.0176   0.751   0.4524
## label         1.3110     0.6196   2.116   0.0344 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) robtpp rbtrnd
## robotopp    -0.664
## robotrandom -0.722  0.730
## label       -0.005 -0.557 -0.446
```

```
fixef(model_glmer)
```

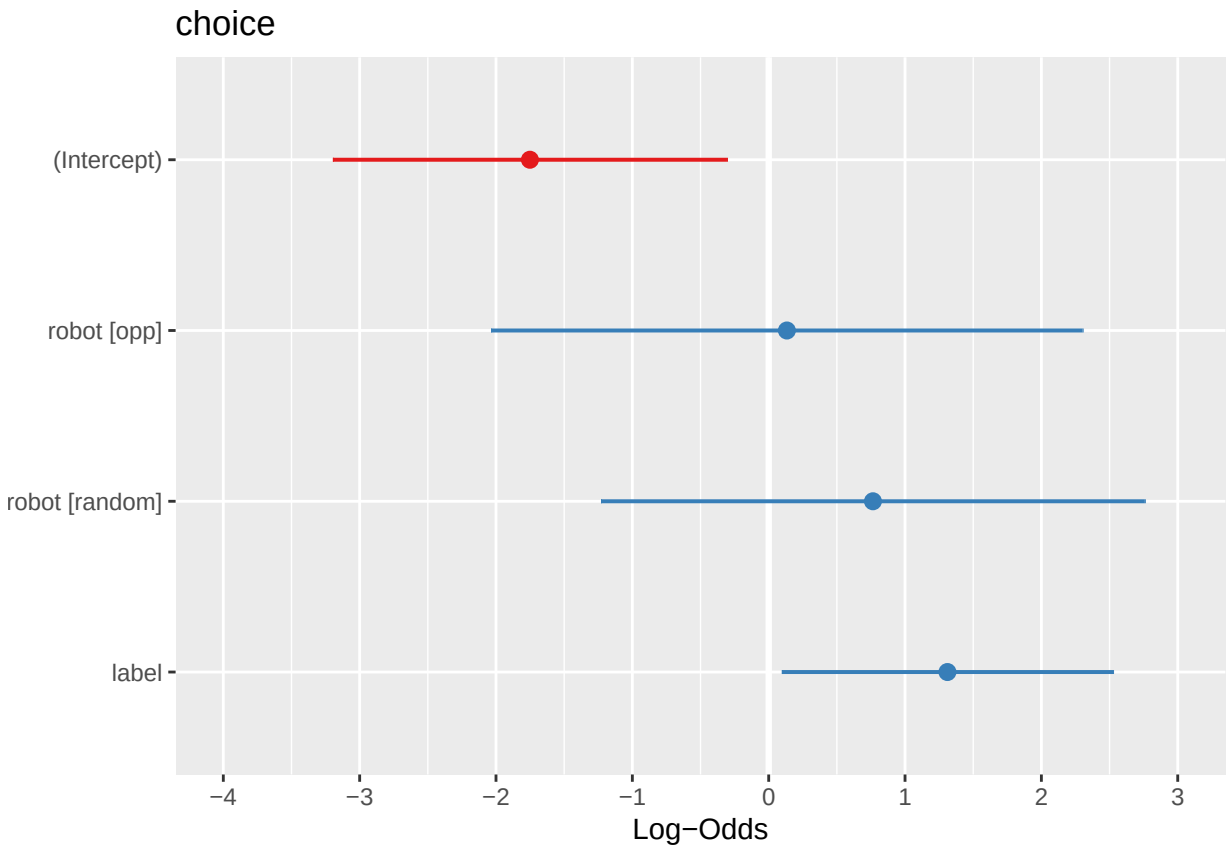
```
## (Intercept)    robotopp robotrandom       label
##  -1.7511770   0.1334172   0.7645967   1.3110203
```

```
ranef(model_glmer)
```

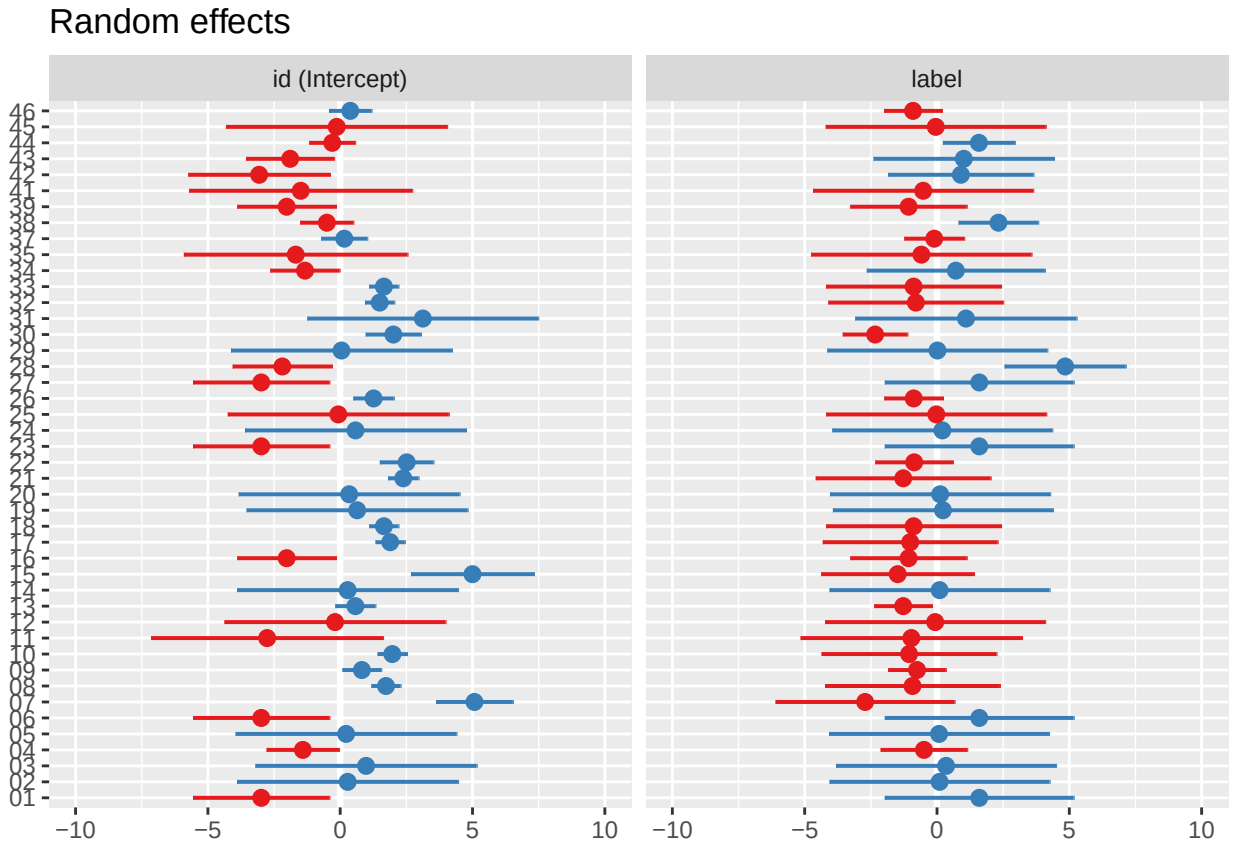
```
## $id
##    (Intercept)       label
## 01 -2.98323882  1.59739353
## 02  0.28112909  0.09857435
## 03  0.97805347  0.34294204
## 04 -1.41113056 -0.49274258
## 05  0.22295674  0.07817695
## 06 -2.98323882  1.59739353
## 07  5.07346433 -2.71661760
## 08  1.73095877 -0.92685249
## 09  0.81881514 -0.75441565
## 10  1.96903633 -1.05433258
## 11 -2.76130623 -0.96821699
## 12 -0.19495239 -0.06835758
## 13  0.57435338 -1.27818822
## 14  0.28112909  0.09857435
## 15  4.99800248 -1.48581415
## 16 -2.02295606 -1.07384844
## 17  1.88932901 -1.01165281
## 18  1.65180230 -0.88446768
## 19  0.64020145  0.22447852
## 20  0.33946560  0.11902931
## 21  2.38360541 -1.27631614
## 22  2.51219122 -0.86269247
## 23 -2.98323882  1.59739353
## 24  0.57816461  0.20272608
```

```
## 25 -0.07170540 -0.02514259
## 26  1.26247712 -0.88016150
## 27 -2.98323882  1.59739353
## 28 -2.18646365  4.84388192
## 29  0.04765283  0.01670886
## 30  2.00783295 -2.33856019
## 31  3.12273485  1.09494735
## 32  1.49231089 -0.79906702
## 33  1.65180230 -0.88446768
## 34 -1.33015175  0.71223791
## 35 -1.68159607 -0.58963032
## 37  0.15457022 -0.10320452
## 38 -0.50403578  2.32316706
## 39 -2.02295606 -1.07384844
## 41 -1.49155140 -0.52299357
## 42 -3.06278734  0.90193925
## 43 -1.89117550  1.01264153
## 44 -0.29967020  1.58309523
## 45 -0.13272926 -0.04653983
## 46  0.38133135 -0.90658253
##
## with conditional variances for "id"
```

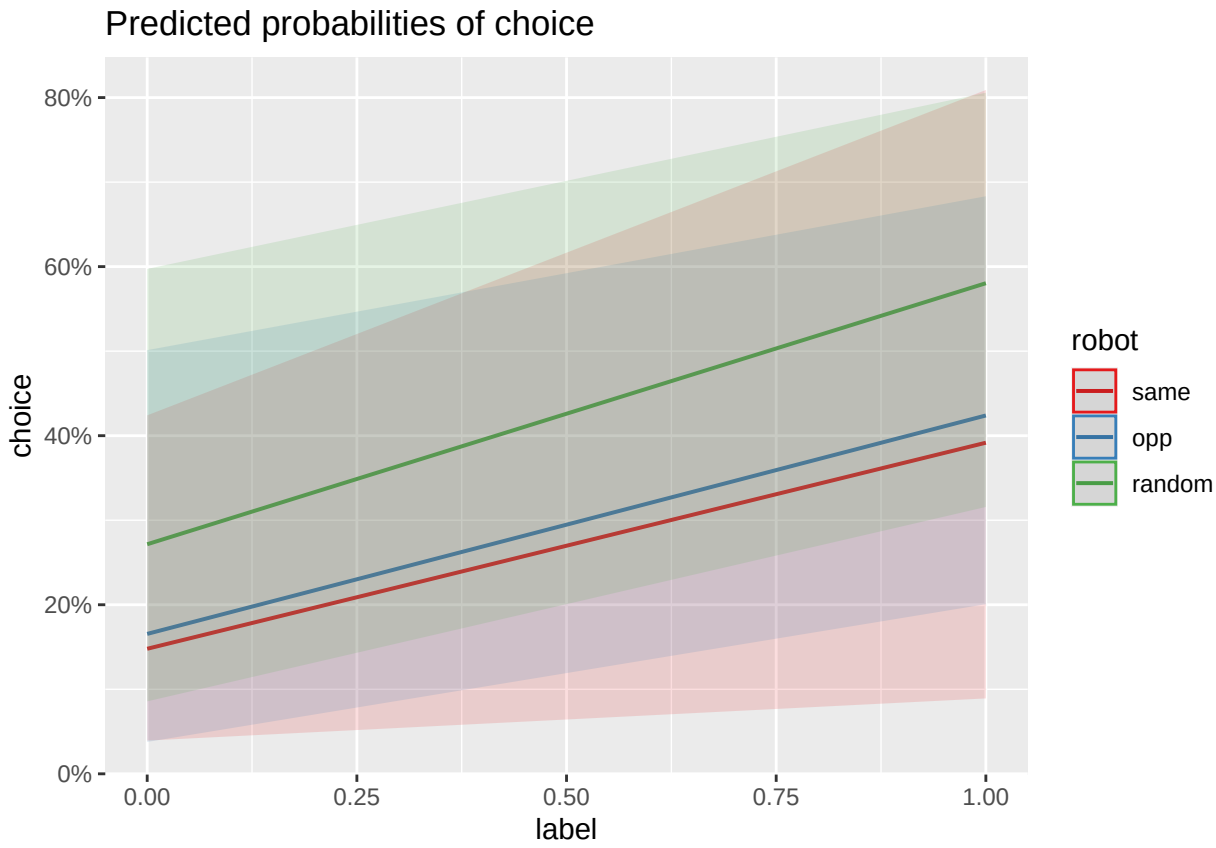
```
plot_model(model_glmer, type = "est", transform = NULL, show.intercept = TRUE)
```




```
plot_model(model_glmer, type = "re", transform = NULL, show.intercept = TRUE)
```

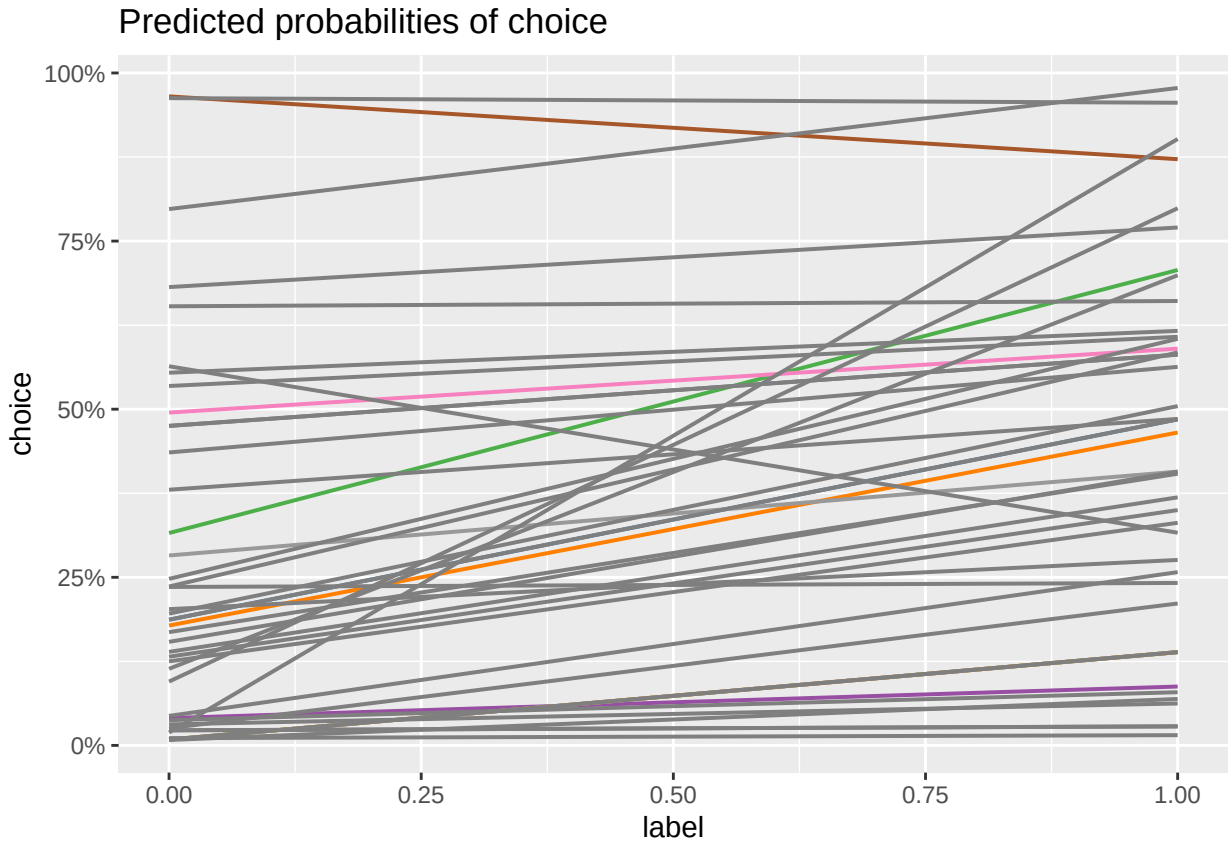



```
plot_model(model_glmer, type = "pred", terms = c("label", "robot"))
```




```
plot_model(model_glmer,
  type = "pred", pred.type = "re", terms = c("label", "id"),
  show.legend = FALSE, ci.lvl = NA
)
```

```
## Warning in RColorBrewer::brewer.pal(n, pal): n too large, allowed maximum for palette Set1 is 9
## Returning the palette you asked for with that many colors
```




```
# type 3 anova comparing one removed
mixed(model_glmer,
  data = df,
  family = binomial,
  method = "LRT", # glmer random levels should be > 50
  test_intercept = TRUE,
  check_contrasts = FALSE # dummy coding to ensure intercept is same bias
)
```

```
## Formula (the first argument) converted to formula.
## Numerical variables NOT centered on 0: label
## If in interactions, interpretation of lower order (e.g., main) effects difficult.
```

```
## Mixed Model Anova Table (Type 3 tests, LRT-method)
##
## Model: choice ~ robot + label + (label | id)
## Data: df
## Df full model: 7
##        Effect df  Chisq p.value
## 1 (Intercept)  1 5.35 *    .021
## 2       robot  2   0.95    .623
## 3       label  1 3.83 +    .050
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '+' 0.1 ' ' 1
```

```
# on amarel
# nc <- detectCores() - 2
# cl <- makeCluster(rep("localhost", nc)) # make cluster
#
```

```r
# tic()
# mixed(model_glmer,
#   data = df, family = binomial, method = "PB",
#   check_contrasts = FALSE, test_intercept = TRUE, cl = cl, # parallelize model solving
#   args_test = list(cl = cl, nsim = 1000) # parallelize pkbr simulation
# )
# toc()


# Power analysis ----------------------------------------------------------
sqglmer <- safely(.f = quietly(.f = glmer))

es <- 0.362
alpha <- 0.05

get_coefs <- function(id_num, img_num) {
  sampled_df <- df |>
    nest(data = -c(robotCondition, id)) |>
    group_by(robotCondition) |>
    slice_sample(n = id_num %/% 3, replace = TRUE) |>
    unnest(data) |>
    group_by(id) |>
    slice_sample(n = img_num, replace = TRUE) |>
    ungroup()

  results_list <- sqglmer(choice ~ robot + label + (1 | id), data = sampled_df, family = binomial)

  if (is.null(results_list$error)) {
    # replace model with coefs to save memory
    results_list$result$result <- coef(summary(results_list$result$result))
  }
  results_list
}

num_reps <- 2

power_df <- expand_grid(id_num = seq(30, 100, length.out = 3), img_num = seq(10, 40, length.out = 3)) |>
  slice(rep(1:n(), times = num_reps)) |>
  glimpse()
```

```
## Rows: 18
## Columns: 2
## $ id_num  <dbl> 30, 30, 30, 65, 65, 65, 100, 100, 100, 30, 30, 30, 65, 65, 65,~
## $ img_num <dbl> 10, 25, 40, 10, 25, 40, 10, 25, 40, 10, 25, 40, 10, 25, 40, 10~
```

```r
plan(multisession, workers = availableCores() - 5)
tic()
multi_power_df <- power_df |>
  mutate(data = future_map2(id_num, img_num, ~ get_coefs(.x, .y),
    .progress = TRUE,
    .options = furrr_options(seed = TRUE)
  ))
toc()
```

```
## 16.301 sec elapsed
```

```
power_df <- multi_power_df |>
  unnest_wider(data) |>
  filter(is.na(error)) |> # exclude error
  unnest_wider(result) |>
  filter(is.na(warnings) || lengths(warnings) == 0) |> # exclude convergence issues
  glimpse()
```

```
## Warning in is.na(warnings) || lengths(warnings) == 0: 'length(x) = 18 > 1' in
## coercion to 'logical(1)'
```

```
## Rows: 18
## Columns: 7
## $ id_num   <dbl> 30, 30, 30, 65, 65, 65, 100, 100, 100, 30, 30, 30, 65, 65, 65~
## $ img_num  <dbl> 10, 25, 40, 10, 25, 40, 10, 25, 40, 10, 25, 40, 10, 25, 40, 1~
## $ result   <list> <<matrix[4 x 4]>>, <<matrix[4 x 4]>>, <<matrix[4 x 4]>>, <<m~
## $ output   <chr> "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "~
## $ warnings <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ messages <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ error    <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
```

```
power_df <- power_df |>
  rowwise() |>
  mutate(
    label_sig = (result["label", "Pr(>|z|)"] < alpha & abs(result["label", "Estimate"]) > es),
    int_sig = (result["(Intercept)", "Pr(>|z|)"] < alpha & abs(result["(Intercept)", "Estimate"]) > es)
  ) |>
  glimpse()
```

```
## Rows: 18
## Columns: 9
## Rowwise:
## $ id_num    <dbl> 30, 30, 30, 65, 65, 65, 100, 100, 100, 30, 30, 30, 65, 65, 6~
## $ img_num   <dbl> 10, 25, 40, 10, 25, 40, 10, 25, 40, 10, 25, 40, 10, 25, 40, ~
## $ result    <list> <<matrix[4 x 4]>>, <<matrix[4 x 4]>>, <<matrix[4 x 4]>>, <<~
## $ output    <chr> "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", ~
## $ warnings  <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ messages  <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ error     <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ label_sig <lgl> TRUE, TRUE, FALSE, FALSE, TRUE, TRUE, TRUE, TRUE, TRUE, FALS~
## $ int_sig   <lgl> TRUE, FALSE, FALSE, TRUE, TRUE, FALSE, TRUE, TRUE, TRUE, TRU~
```

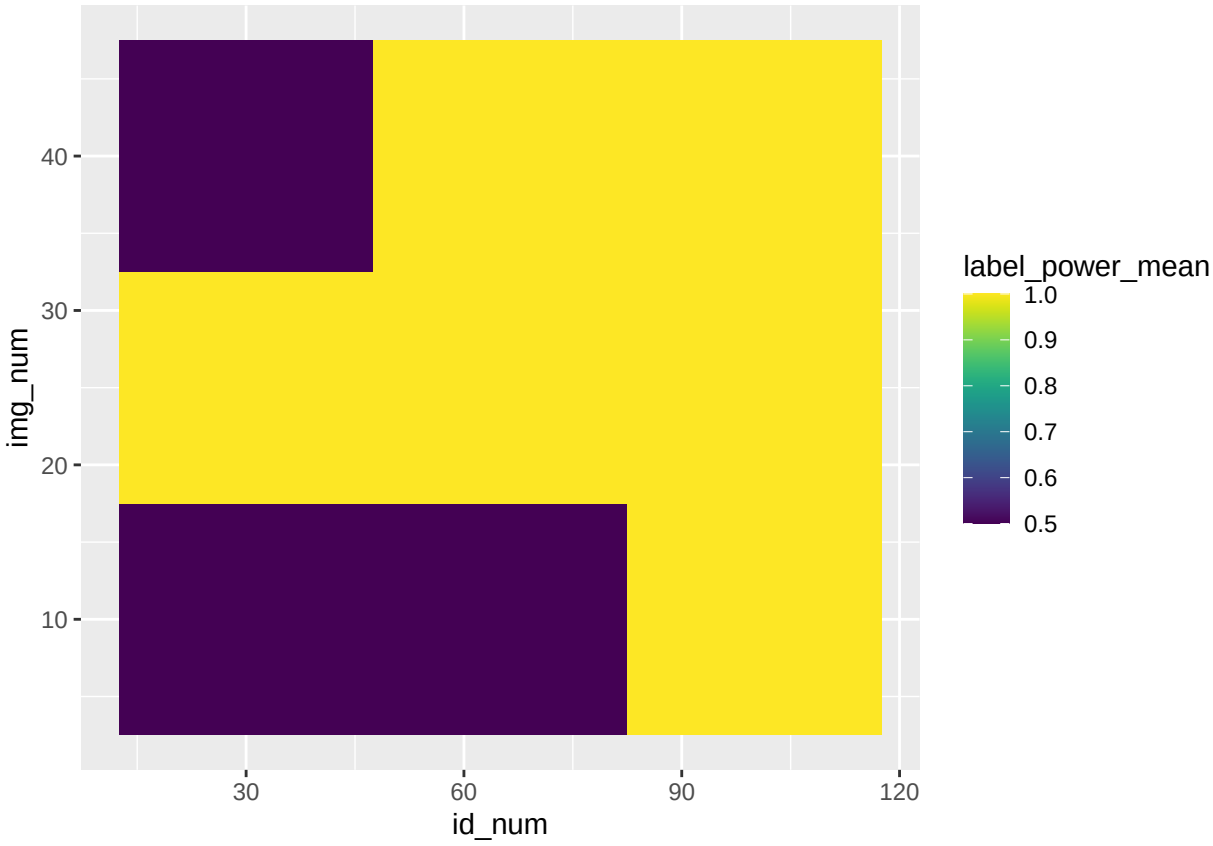
```
power_df <- power_df |>
  group_by(id_num, img_num) |>
  summarise(label_power = binom.confint(sum(label_sig), num_reps, methods = "wilson"),
            int_power = binom.confint(sum(int_sig), num_reps, methods = "wilson")) |>
  unnest_wider(c(label_power, int_power), names_sep = "_") |>
  glimpse()
```

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

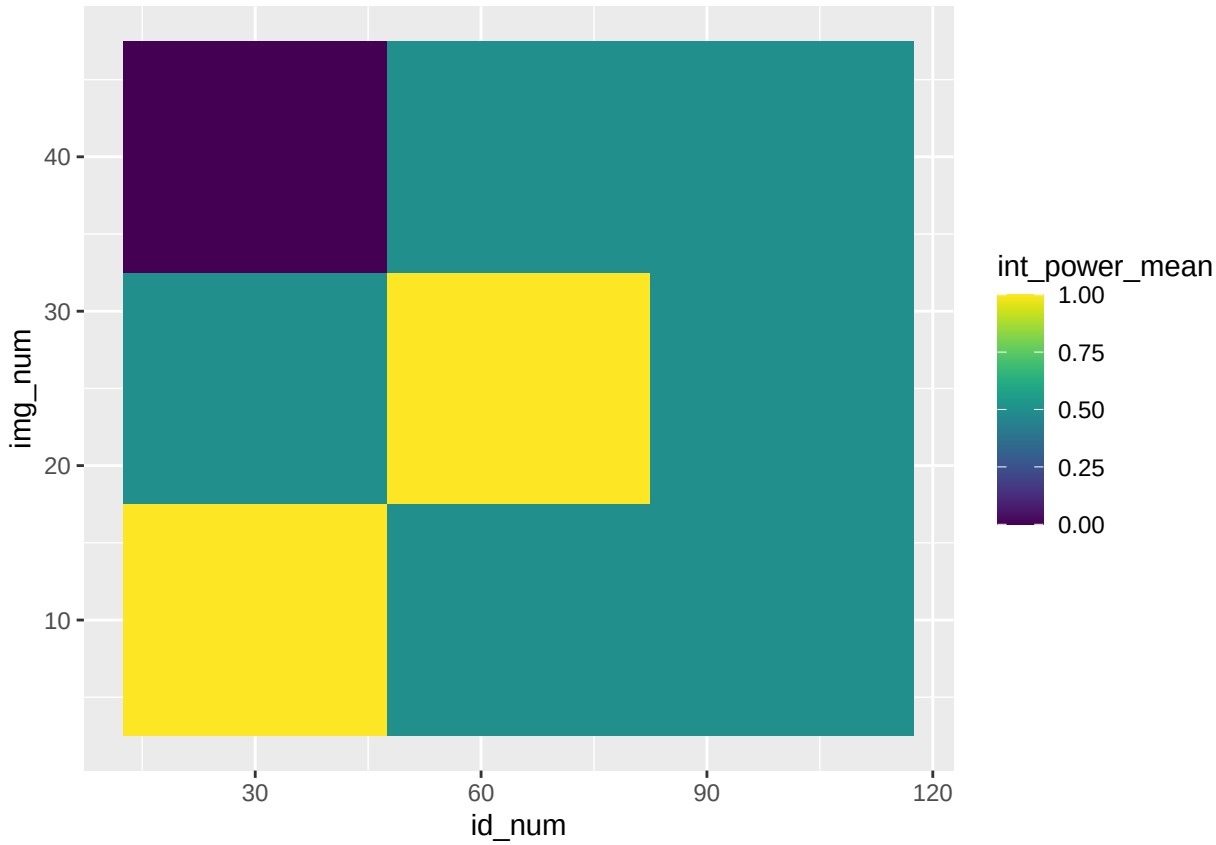
```
## Rows: 9
## Columns: 14
## Groups: id_num [3]
## $ id_num             <dbl> 30, 30, 30, 65, 65, 65, 100, 100, 100
```

```
## $ img_num            <dbl> 10, 25, 40, 10, 25, 40, 10, 25, 40
## $ label_power_method <chr> "wilson", "wilson", "wilson", "wilson", "wilson", "~
## $ label_power_x      <int> 1, 2, 1, 1, 2, 2, 2, 2, 2
## $ label_power_n      <dbl> 2, 2, 2, 2, 2, 2, 2, 2, 2
## $ label_power_mean   <dbl> 0.5, 1.0, 0.5, 0.5, 1.0, 1.0, 1.0, 1.0, 1.0
## $ label_power_lower  <dbl> 0.09453121, 0.34238023, 0.09453121, 0.09453121, 0.3~
## $ label_power_upper  <dbl> 0.9054688, 1.0000000, 0.9054688, 0.9054688, 1.00000~
## $ int_power_method   <chr> "wilson", "wilson", "wilson", "wilson", "wilson", "~
## $ int_power_x        <int> 2, 1, 0, 1, 2, 1, 1, 1, 1
## $ int_power_n        <dbl> 2, 2, 2, 2, 2, 2, 2, 2, 2
## $ int_power_mean     <dbl> 1.0, 0.5, 0.0, 0.5, 1.0, 0.5, 0.5, 0.5, 0.5
## $ int_power_lower    <dbl> 0.34238023, 0.09453121, 0.00000000, 0.09453121, 0.3~
## $ int_power_upper    <dbl> 1.0000000, 0.9054688, 0.6576198, 0.9054688, 1.00000~
```

```
power_df |> ggplot(aes(id_num, img_num)) +
  geom_tile(aes(fill = label_power_mean)) +
  scale_fill_viridis()
```




```
power_df |> ggplot(aes(id_num, img_num)) +
  geom_tile(aes(fill = int_power_mean)) +
  scale_fill_viridis()
```




```
# Side dishes ------------------------------------------------------------

# side <- read_csv("invariance/psiturk/questiondata.csv", col_names = c("id", "question", "response"))
#   glimpse()
#
# side <- side |> pivot_wider(names_from = question, values_from = response)
#
# side <- side |> filter(id %in% df$id)
#
# side |>
#   group_by(condition) |>
#   count()
#
# side |>
#   group_by(robotCondition) |>
#   count()
#
# side |>
#   group_by(gender) |>
#   count()
#
# side |>
#   ggplot(aes(as.numeric(age))) +
#   geom_dotplot()
#
# f <- side |>
#   filter(!is.na(feedback))
```

```
#
# hist(as.numeric(side$meanMainRT) / 1000)
#
# hist(as.numeric(side$timeElapsed) / 1000 / 60)
```