

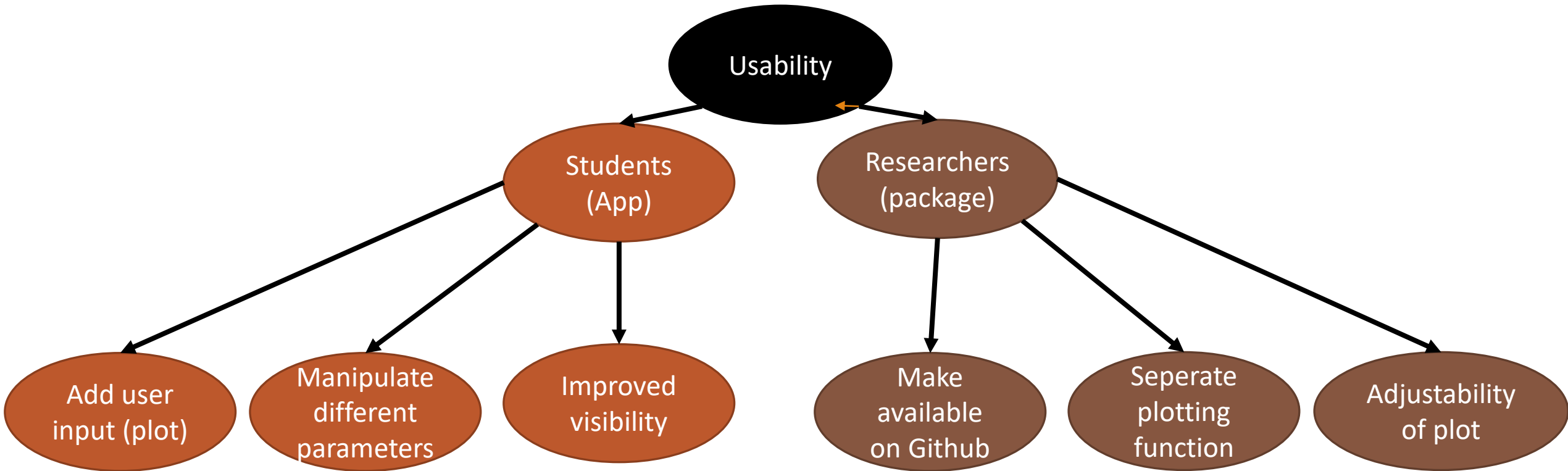
Visualising Psychophysical Models

LISA, KONRAD, LEO, SEAN, THOMAS

SUPERVISED BY ROCIO ALCALA-QUINTANA

A solid orange horizontal bar at the bottom of the slide.

Goal of the Project



Plot Improvements

Make plots more customizable

Allow plotting separately to model fitting

Allow plotting directly from output object

Make subfunctions for individual plots

Previous Plotting

One long plotting function

Internal within model fitting function

Not customisable

```
671 plot_results <- function(pars, Standard, Lev, F1, U1, S1, F2, U2, S2, PSE, Thr_84, logs) {
692     expression(bold(paste("Test ", 1^st, ": ", phantom(paste("F U S   Test ", 2^nd, ": ", "F
693     , cex=0.9)
694     title(main =
695     expression(bold(paste(phantom(paste("Test ", 1^st, ": ")), "F", phantom(paste("U S   Te:
696     , cex=0.9, col.main="blue")
697     title(main =
698     expression(bold(paste(phantom(paste("Test ", 1^st, ": F")), "U", phantom(paste("S   Tes:
699     , cex=0.9, col.main="black")
700     title(main =
701     expression(bold(paste(phantom(paste("Test ", 1^st, ": F U")), "S", phantom(paste("   Te:
702     , cex=0.9, col.main="red")
703     title(main =
704     expression(bold(paste(phantom(paste("Test ", 1^st, ": F U S")), "   Test ", 2^nd, ": ", |
705     , cex=0.9, col.main="black")
706     title(main =
707     expression(bold(paste(phantom(paste("Test ", 1^st, ": F U S   Test ", 2^nd, ": ")), "F", pl
708     , cex=0.9, col.main="pink")
709     title(main =
710     expression(bold(paste(phantom(paste("Test ", 1^st, ": F U S   Test ", 2^nd, ": F")), "U", |
711     , cex=0.9, col.main="gray")
712     title(main =
713     expression(bold(paste(phantom(paste("Test ", 1^st, ": F U S   Test ", 2^nd, ": F U")), "S":
714     , cex=0.9, col.main="cyan")
715 } else if (bin) {
716     title(main =
717     expression(bold(paste("Test ", 1^st, ": ", phantom(paste("F   Test ", 2^nd, ": S")))))
718     , cex=0.9)
719     title(main =
720     expression(bold(paste(phantom(paste("Test ", 1^st, ": ")), "F", phantom(paste("   Test '
831     lines(c(Inr_84, Inr_84), c(U, 1), col="black")
832 }
833 }
```

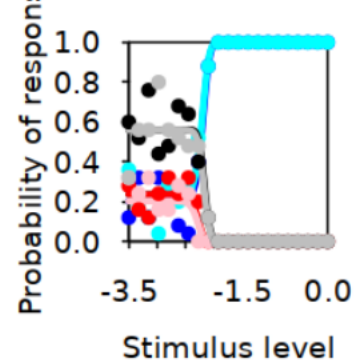
Previous Plotting

Functional, but hard to customize

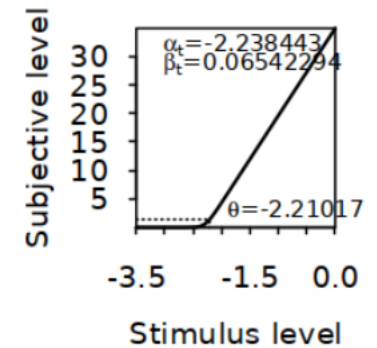
Only provides all plots together – no option to plot individually

Readability issues

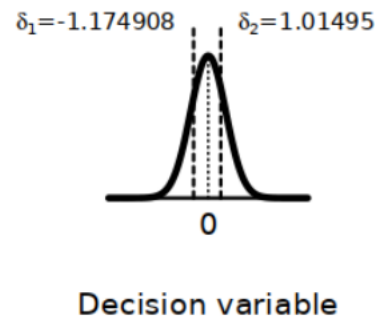
Test 1st: FUS Test 2nd: FUS



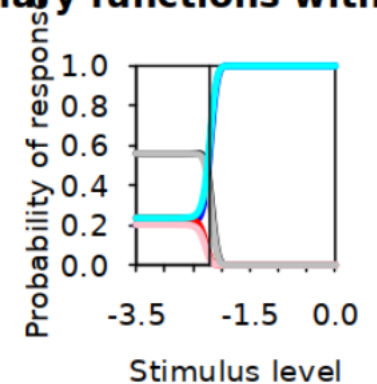
Psychophysical function



Decision boundaries

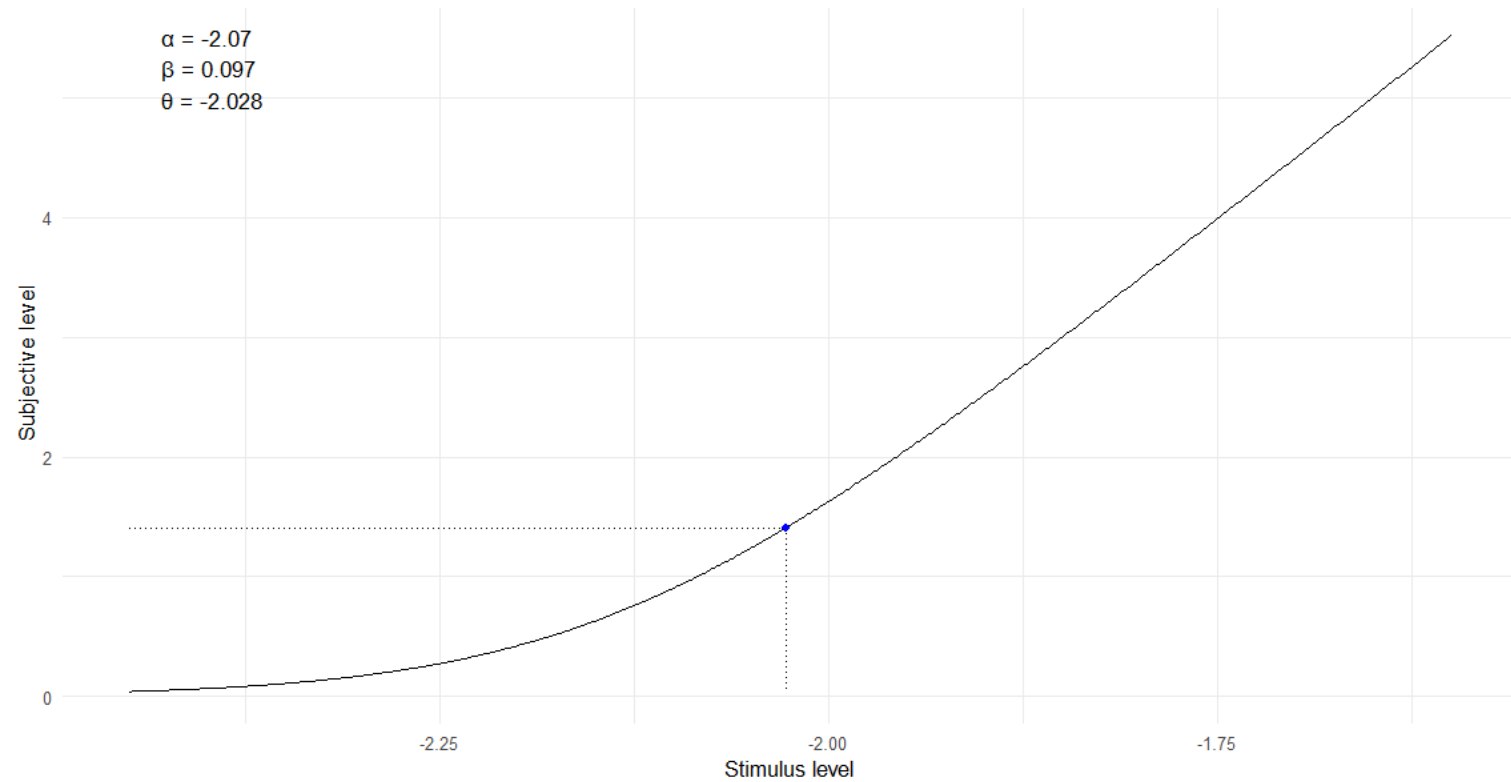
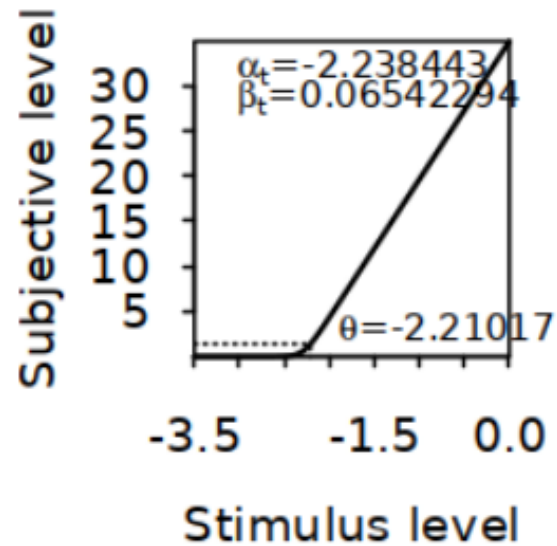


Ternary functions with all $\varepsilon = 0$



Psychophysical Function

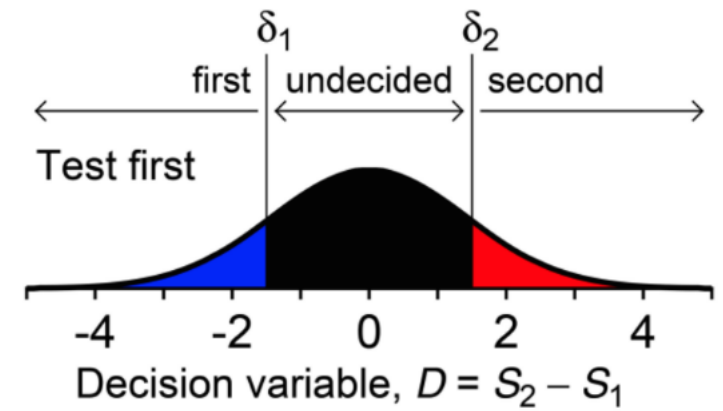
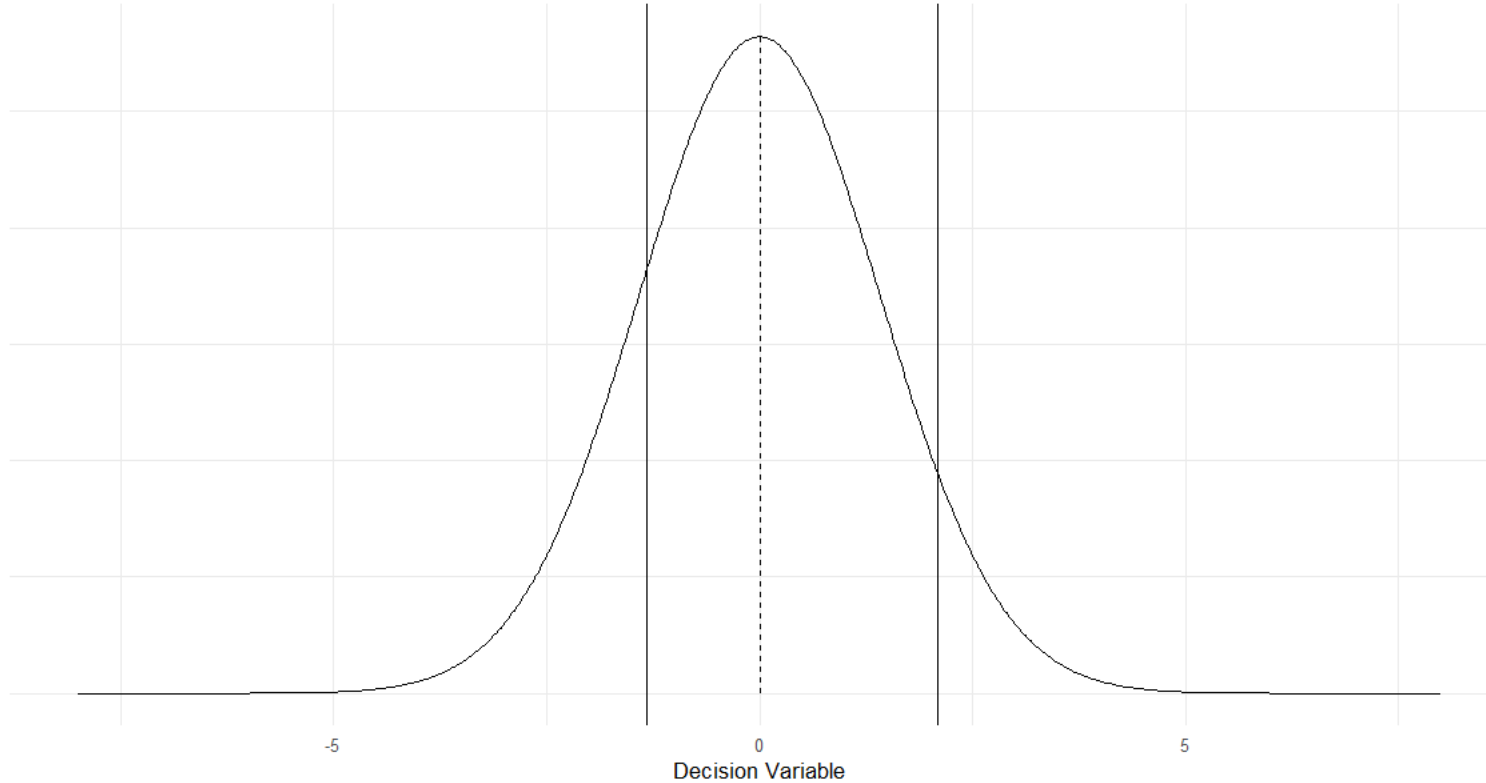
Psychophysical function



Psychophysical Function

The psychophysical function, μ , relates physical magnitude (x-axis) to sensory effect (y-axis). While different forms have been proposed for different sensory modalities and stimuli, here, the functional form of μ is a model assumption, since it can accommodate a wide range of modalities (e.g., pain, passage of time, brightness).

Decision Boundary

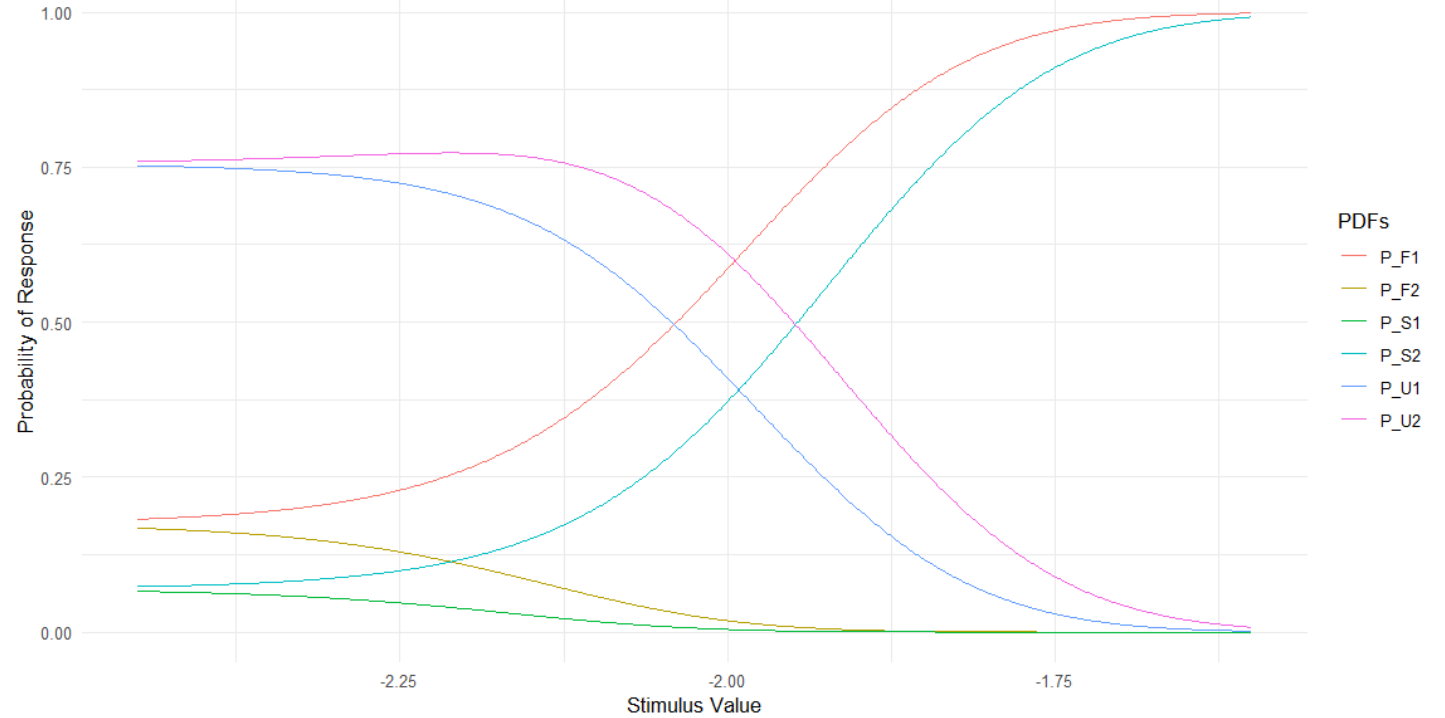
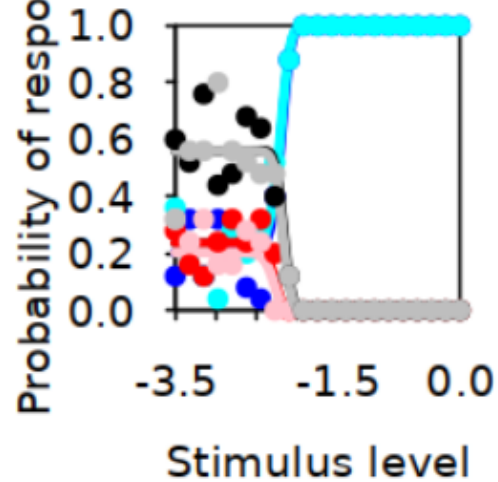


Decision Boundaries

Judgements are performed using: a) a decision variable that represents the perceived difference between two sensory effect (x-axis; $D = S_2 - S_1$); b) two criteria, δ_1 and δ_2 that partition the decision space and determine the responses.

Psychometric Function

Test 1st: FUS Test 2nd: FUS



Psychometric Function

Psychometric functions depict the probability (y-axis) of each type of judgment (color) as a function of the test level (x-axis). Perceptual effects ($\mu_s \neq \mu_t$) shift psychometric functions away from the standard level equally regardless of presentation order, while decisional bias ($\delta_1 \neq -\delta_2$) shifts psychometric functions for each presentation order on opposite directions relative to the common location they would have without such bias.

Overall Structure

Individual function for each plot

Wrapper function to combine plots together

```
# plot decision boundary function
plot_boundary <- function(model){

  d1 = model$delta_1; d2 = model$delta_2
  xsup <- ceiling(max(abs(c(8, d1, d2)))) # take absmx of decision bounds for xlims
  xinf <- -xsup # set negative version for lower bound
  step <- (xsup-xinf)/800 # abs sum of both over 800 to define intervals
  x <- seq(xinf,xsup,step) # set 800 steps between limits
  y <- dnorm(x,0,sqrt(2)) # normal distribution (u=0,sd=2^0.5)
  data <- data.frame(x=x, y=y)

  p <- ggplot(data, aes(x,y)) +
    geom_line() +
    geom_vline(xintercept = d1) +
    geom_vline(xintercept = d2) +
    geom_segment(aes(x=0 , y=0, xend=0, yend=max(y)), linetype=2) +
    xlab('Decision Variable') +
    theme_minimal() +
    theme(axis.title.y = element_blank(),
          axis.text.y = element_blank())

  return(p)
}
```

```
# wrapper function
plot_results <- function(model,to_plot = 1:3){
  p = NULL

  if(1 %in% to_plot){
    p1 <- plot_psychphys(model)
  }
  if(2 %in% to_plot){
    p2 <- plot_boundary(model)
  }
  if(3 %in% to_plot){
    p3 <- plot_psychmet(model)
  }

  cowplot::plot_grid(plotlist=mget(paste0("p", to_plot)))
}
```

Over to the app...

https://levolz.shinyapps.io/psychphys_app

