RLDM_DDMAssignment

Larissa Weyler

2024-05-14

DD Model

```
DDM <- function(threshold, evidenceLeft, evidenceRight, noise, bias=0.5, truth) {
  act <- threshold*bias
  time <- 0
  while ((act < threshold) & (act > 0)) {
    act <- act + (evidenceLeft - evidenceRight) + noise
    time <- time + 1
  }
  decision <- ifelse(act >= threshold, "left", "right")
  accuracy <- ifelse(decision == truth, "correct", "incorrect")
  results <- list("time" = time, "decision" = decision, "accuracy" = accuracy, "noise" = noise)
  return(results)
}
```

Experiment

Setting up matrix

```
ntrials <- 500
nparticipants <- 15

resultsThreshold <- data.frame(matrix(nrow = ntrials*nparticipants, ncol = 9, dimnames= list (NULL, c("

resultsDrift <- data.frame(matrix(nrow = ntrials*nparticipants, ncol = 9, dimnames= list (NULL, c("parti
```

Varying threshold

Simulating data with varying thresholds per condition.

```
for (i in 1:nparticipants) {
  bias <- rnorm(1, mean = 0.5, sd = 0.1) # every person has slightly different bias, but on average the
  for (j in 1:ntrials){
    condition <- sample(c("cheetah", "turtle"), 1) # randomly choose condition
    noise <- rnorm(1, sd = 1)
    evidenceCorrect <- rnorm(1, mean = 0.1, sd = 0.05) # how much evidence is increased for correct dec
    truth <- sample(c("left", "right"), 1) # determine correct decision
```

```
    threshold <- ifelse(condition == "cheetah", rnorm(1, mean = 10, sd = 1), rnorm(1, mean = 20, sd = 1
    if (truth == "left") { # bias correct decision
        evidenceLeft <- 0.2 + evidenceCorrect # base evidence rate + varyingly increase in evidence for
        evidenceRight <- 0.2 - evidenceCorrect
    } else if (truth == "right") {
      evidenceLeft <- 0.2 - evidenceCorrect
      evidenceRight <- 0.2 + evidenceCorrect
    }
    trialResults <- DDM(threshold, # temporarily save results of trial
                        evidenceLeft,
                        evidenceRight,
                        noise,
                        bias,
                        truth)
    resultsThreshold[(i - 1) * ntrials + j, ] <- c(i, condition,
                                          trialResults$time,
                                          trialResults$decision,
                                          trialResults$accuracy,
                                          threshold,
                                          bias,
                                          evidenceLeft - evidenceRight,
                                          noise) #assign results of trial to dataframe

  }
resultsThreshold$time <- as.numeric(resultsThreshold$time)
}
```

Varying drift rate

Simulating data with varying drift rate per condition.

```
for (i in 1:nparticipants) {
  bias <- rnorm(1, mean = 0.5, sd = 0.1)
  threshold <- rnorm(1, mean = 10, sd = 2) # fixed threshold (cautiousness) per person
  for (j in 1:ntrials){
    condition <- sample(c("cheetah", "turtle"), 1)
    noise <- rnorm(1, sd = 1)
    evidenceCorrect <- rnorm(1, mean = 0.1, sd = 0.05)
    truth <- sample(c("left", "right"), 1)
    evidenceCondition <- ifelse(condition=="cheetah", rnorm(1, mean=0.5, sd=0.05), rnorm(1, mean=0.1, s
    if (truth == "left") {
        evidenceLeft <- evidenceCondition + evidenceCorrect # combine evidence rate based on condition
        evidenceRight <- evidenceCondition - evidenceCorrect
    } else if (truth == "right") {
      evidenceLeft <- evidenceCondition - evidenceCorrect
      evidenceRight <- evidenceCondition + evidenceCorrect

    }
    trialResults <- DDM(threshold,
                        evidenceLeft,
                        evidenceRight,
                        noise,
                        bias,
```

```
                          truth)
    resultsDrift[(i - 1) * ntrials + j, ] <- c(i, condition,
                                               trialResults$time,
                                               trialResults$decision,
                                               trialResults$accuracy,
                                               threshold,
                                               bias,
                                               evidenceLeft - evidenceRight,
                                               noise)

  }
resultsDrift$time <- as.numeric(resultsDrift$time)
}
```

Evaluate

```
library(ggplot2)
```

Plots

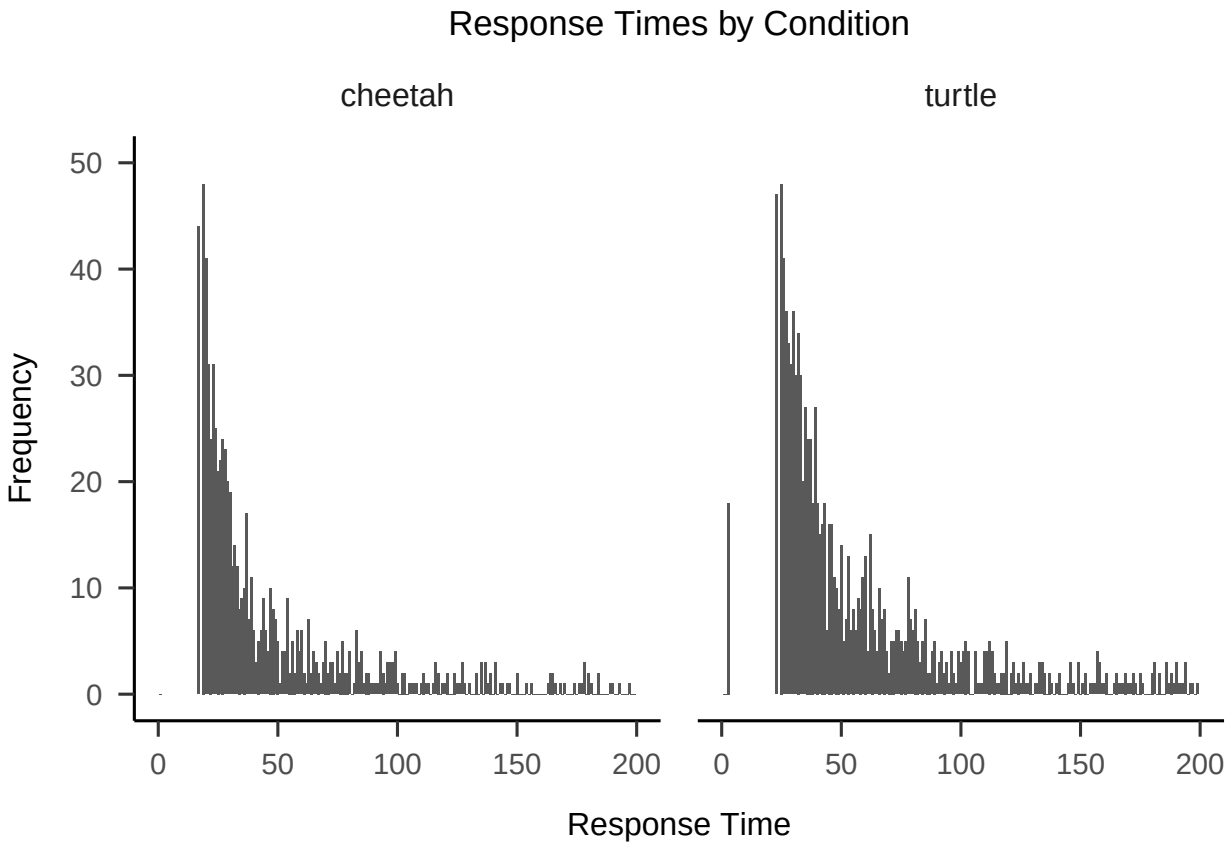
```
## Warning: package 'ggplot2' was built under R version 4.3.3
```

```
# varying threshold
resultsThreshHist <- ggplot(resultsThreshold, aes(x = time)) +
  geom_histogram(binwidth = 1) +
  facet_wrap(~ condition) +
  papaja::theme_apa() +
  labs(title = "Response Times by Condition",
       x = "Response Time",
       y = "Frequency") +
  xlim(c(0,200)) + ylim(0,50)

resultsThreshHist
```

```
## Warning: Removed 227 rows containing non-finite outside the scale range
## (`stat_bin()`).
```

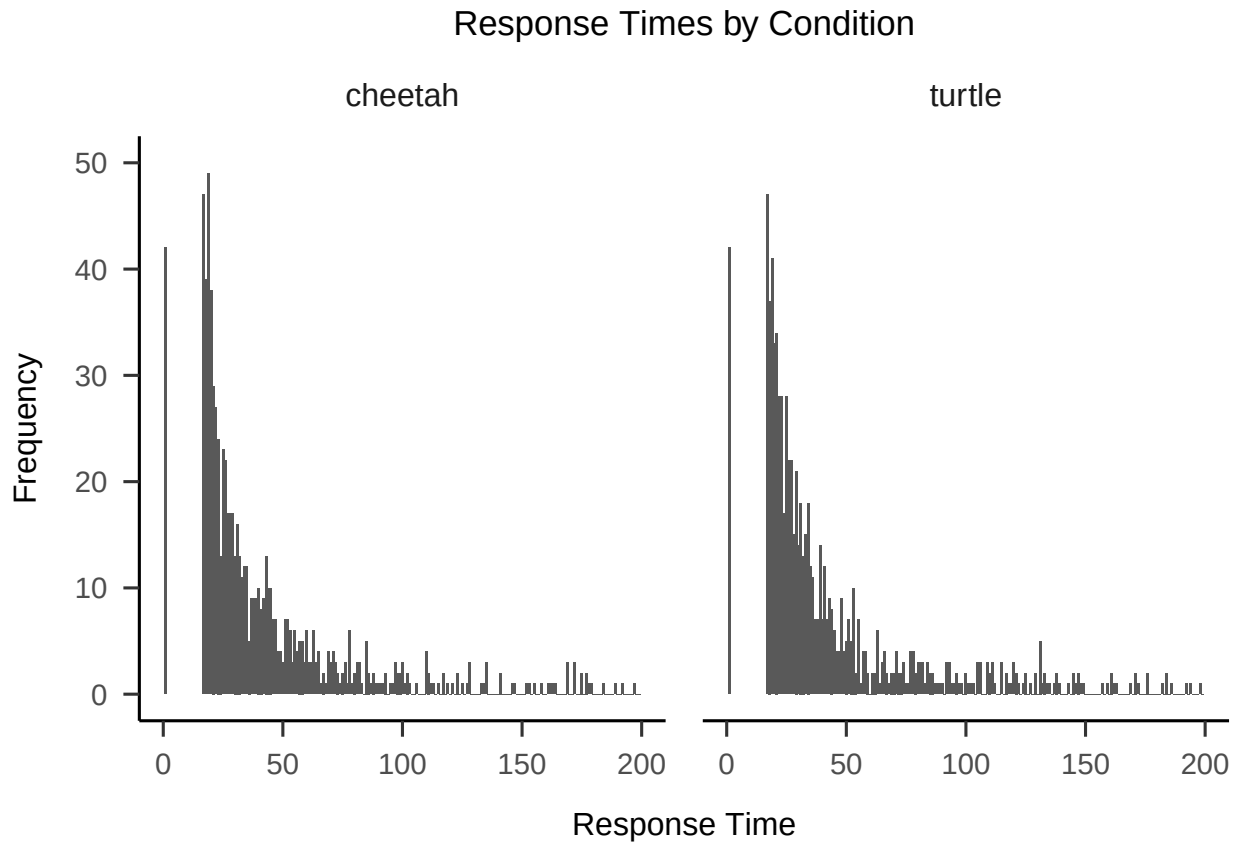
```
## Warning: Removed 40 rows containing missing values or values outside the scale range
## (`geom_bar()`).
```




```
# varying drift rate
resultsDriftHist <- ggplot(resultsDrift, aes(x = time)) +
  geom_histogram(binwidth = 1) +
  facet_wrap(~ condition) +
  papaja::theme_apa() +
  labs(title = "Response Times by Condition",
       x = "Response Time",
       y = "Frequency")  +
  xlim(c(0,200)) + ylim(0,50)
resultsDriftHist
```

```
## Warning: Removed 141 rows containing non-finite outside the scale range
## (‘stat_bin()‘).
```

```
## Warning: Removed 34 rows containing missing values or values outside the scale range
## (‘geom_bar()‘).
```




```
library(dplyr)
```

Summary statistics

```
## Warning: package 'dplyr' was built under R version 4.3.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
summaryStatsThresh <- resultsThreshold %>%
  group_by(condition) %>%
  summarize(
    mean_RT = mean(time),
```

```
    Q1 = quantile(time, 0.25),
    median_RT = median(time),
    Q3 = quantile(time, 0.75)
  )
print(summaryStatsThresh)
```

```
## # A tibble: 2 x 5
##   condition mean_RT    Q1 median_RT    Q3
##   <chr>       <dbl> <dbl>     <dbl> <dbl>
## 1 cheetah      36.7     5         8    16
## 2 turtle      172.      9        15    31
```

```
summaryStatsDrift <- resultsDrift %>%
  group_by(condition) %>%
  summarize(
    mean_RT = mean(time),
    Q1 = quantile(time, 0.25),
    median_RT = median(time),
    Q3 = quantile(time, 0.75)
  )
print(summaryStatsDrift)
```

```
## # A tibble: 2 x 5
##   condition mean_RT    Q1 median_RT    Q3
##   <chr>       <dbl> <dbl>     <dbl> <dbl>
## 1 cheetah      38.7     4         7    15
## 2 turtle       30.6     4         7    15
```

Save data

```
write.csv(resultsThreshold, file = "DDMThresholdResults.csv", row.names = FALSE)
write.csv(resultsDrift, file = "DDMDriftResults.csv", row.names = FALSE)
```