# Supplementary Materials: Search strategies improve with practice, but not with time pressure or financial incentives

ADF Clarke

30/06/2020

## Contents



These supplementary materials contain more details of the Bayesian analysis, including power analysis. Please see the source Rmd file for full code.

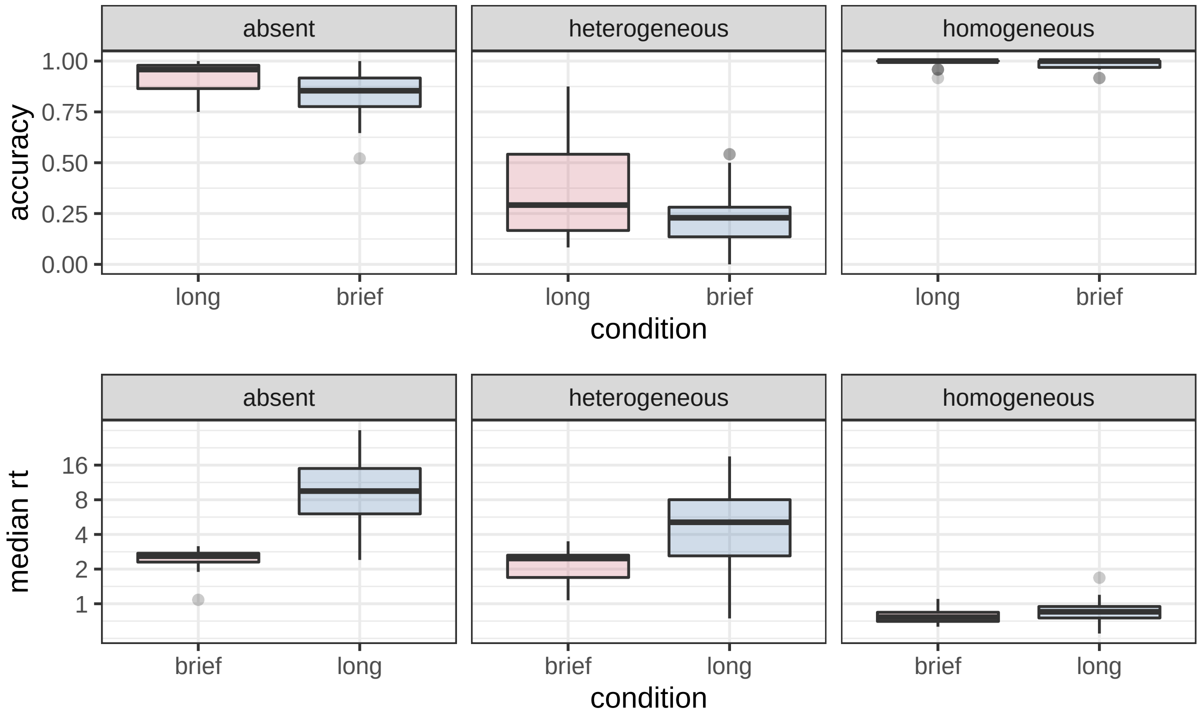
# 1 Experiment 1: Deadline

## 1.1 Descriptive Statistics

We will first look at descriptive statistics for accuracy and reaction time data, to check that it looks sensible and inline with our expectations.

### 1.1.1 Accuracy and Reaction Time

After plotting the accuracy data, incorrect trials are removed from all further analysis.

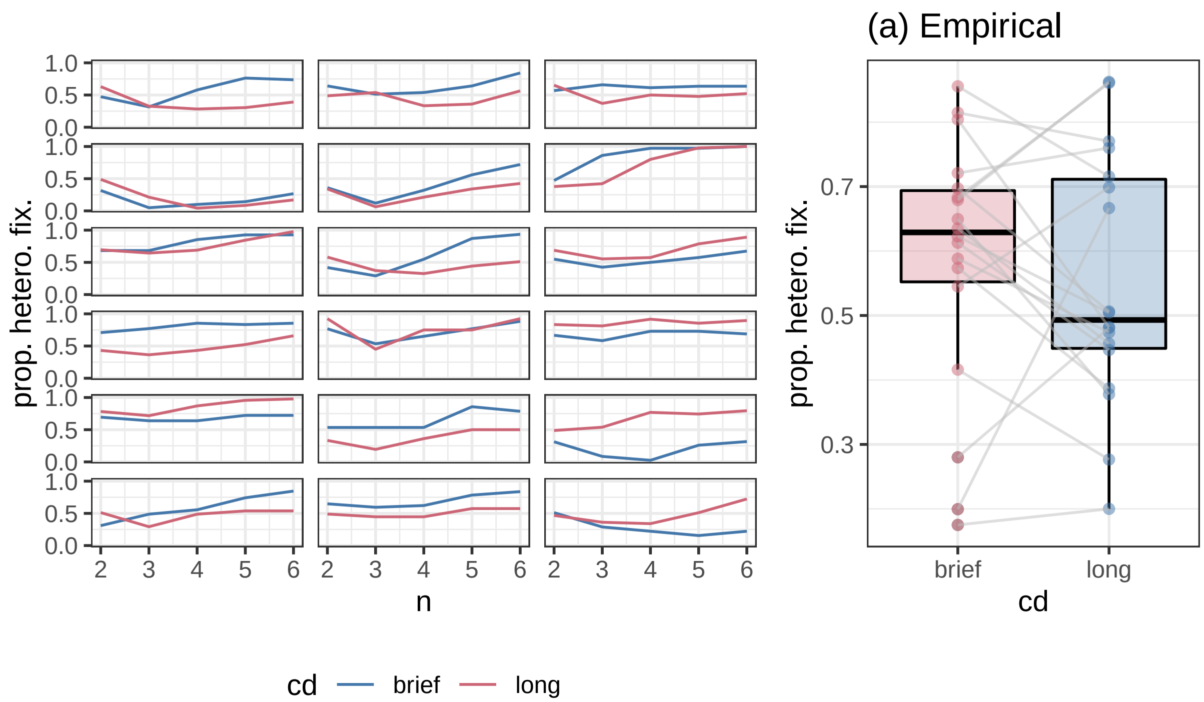



### 1.1.2 Saccadic Strategy

First, we need to merge (join) the fixation and accuracy data, so that we can take only correct target absent trials. We will compute the proportion of fixations to the heterogeneous side of the display for each fixation number, over all trials made by a participant.

Create a facet plot of each individual's strategy.

We can further summarise the data by creating a strategy measure, which is the proportion of all (2 - 6) fixations made by an observer over all trials.




## 1.2 Bayesian Model of Saccadic Strategy

Summarise data so that we have one strategy score per trial per observer.

Note, as beta distributions are only defined over (0, 1), values of 0 and 1 are impossible. To get around this, we will set any such values to 0.001 and 0.999 respectively.

### 1.2.1 Define function for plotting model output

I will want to reuse this plotting code, so I will put it in a function here.

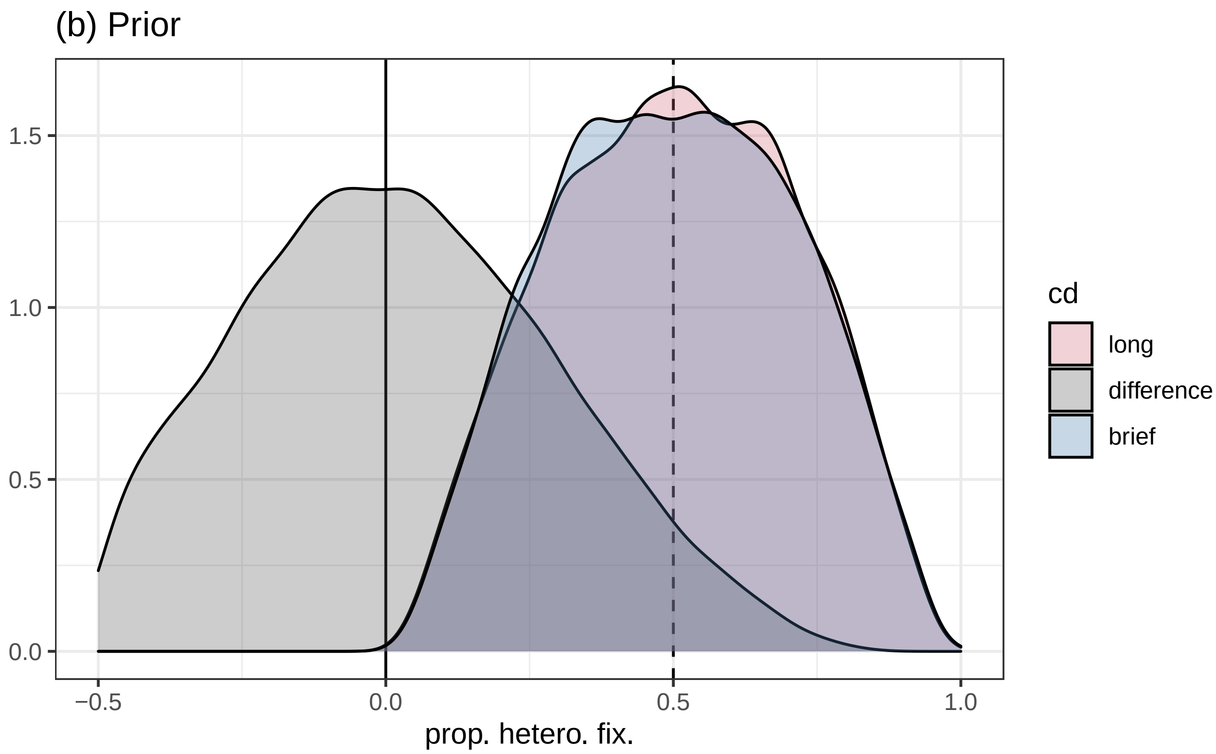
### 1.2.2 Define Priors

We will use $N(0, 1)$, weakly informative priors, illustrated in the plot below.

```
model_priors <- c(
    prior(normal(0, 1), class = "b"))
```

```
prior_model <- brm(
    data = d_strat,
    prop_hetero ~ 0 + cd + (cd | observer),
    family = "beta",
    sample_prior = "only",
    prior = model_priors,
    iter = 5000,
    control = list(adapt_delta = 0.95))
```

```
## Warning: Removed 460 rows containing non-finite values (stat_density).
```




### 1.2.3 Power Analysis

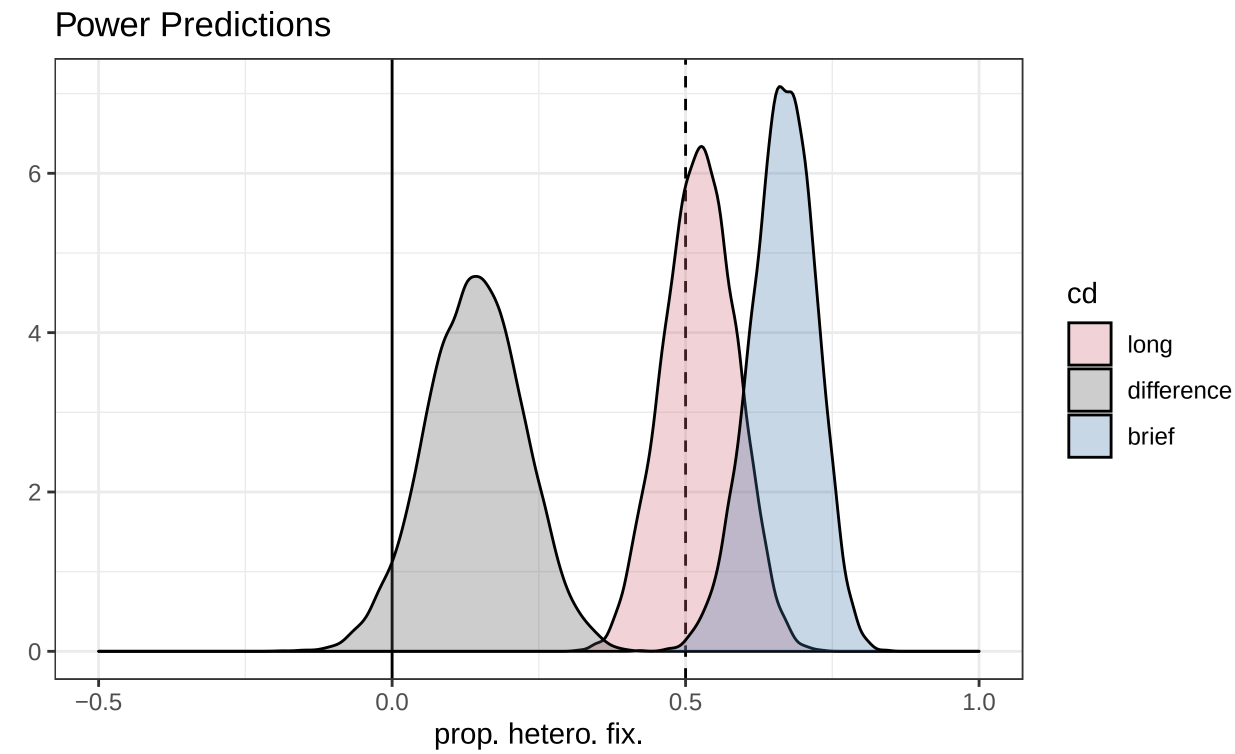
We will carry out our power analysis by simulating our experiment assuming the distributions below.

```
power_prior <- c(
  prior(normal(0.1, 0.25), class = "b", coef = "cdlong"),
    prior(normal(0.7, 0.25), class = "b", coef = "cdbrief"),
  prior(student_t(3, 0, 2), class = "sd"),
  prior(gamma(1, 10), class = "phi")
  )
```

```
## Compiling the C++ model
```

```
## Start sampling
```

We can now plot these distributions to check that they seem reasonable.

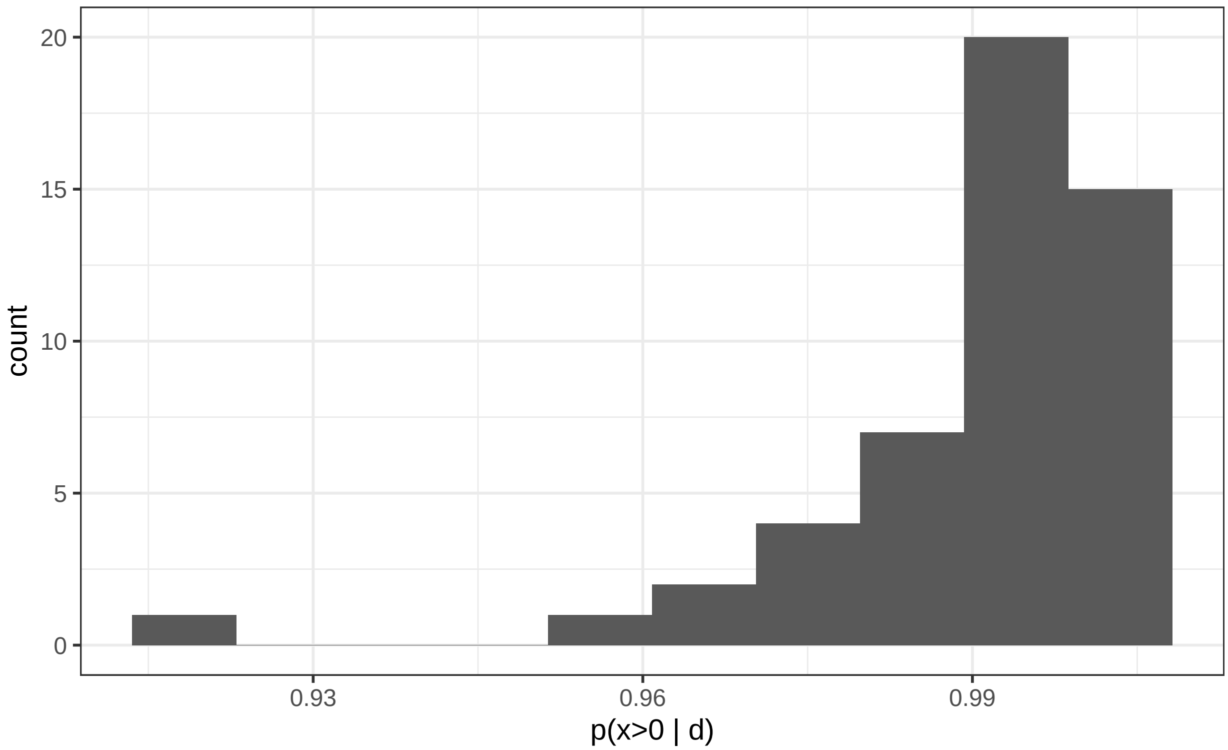



These corresponds to assuming distributions with the means presented below:

| cd | mean prop. fix hetero. |
|---|---|
| brief | 0.6096211 |
| long | 0.4915794 |

We now generate multiple (=50) simulated datasets with 15 observers and 32 correct target absent trials. We then compute $p(\delta > 0|d)$, (the probablity, given the data, of seeing postive difference between the brief and long conditions) for each. The expected distribution of this statistic is shown below. We can see that in (almost?) every iteration, we get a value about 0.95.

And plot!



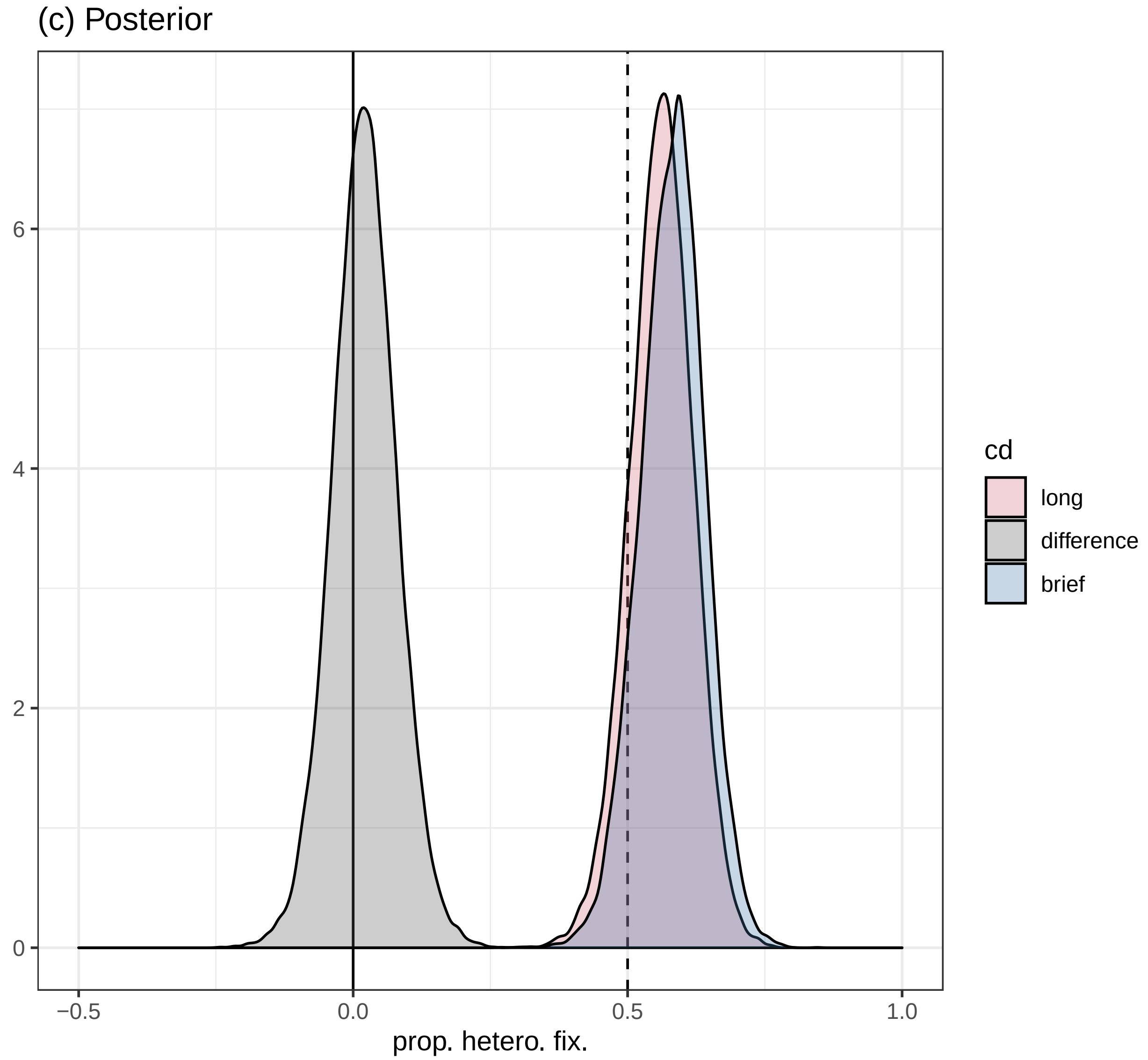

## 1.3 Compute and Plot Posterior

Now that we are confident that we have a sensible prior, and have carried out a power analysis, it is time to fit the model to the data.

```
my_model <- brm(
  data = d_strat,
  prop_hetero ~ 0 + cd + (cd | observer),
  family = "beta",
    prior = model_priors,
    iter = 10000,
    control = list(adapt_delta = 0.95))
```

```
## Compiling the C++ model

## recompiling to avoid crashing R session

## Start sampling

##  Family: beta
##   Links: mu = logit; phi = identity
## Formula: prop_hetero ~ 0 + cd + (cd | observer)
##    Data: d_strat (Number of observations: 1520)
## Samples: 4 chains, each with iter = 10000; warmup = 5000; thin = 1;
##          total post-warmup samples = 20000
##
## Group-Level Effects:
## ~observer (Number of levels: 18)
##                          Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)                1.06      0.20     0.74     1.53 1.00     4643     7710
## sd(cdlong)                   1.00      0.20     0.68     1.47 1.00     4901     8020
```

```
## cor(Intercept,cdlong)    -0.49      0.19    -0.79    -0.06 1.00     6121     8289
##
## Population-Level Effects:
##         Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## cdbrief     0.33      0.24    -0.15     0.82 1.00     3729     5880
## cdlong      0.25      0.24    -0.23     0.71 1.00     4868     7706
##
## Family Specific Parameters:
##     Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## phi     1.71      0.06     1.60     1.83 1.00    17215    12716
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```




```
## Warning: Removed 460 rows containing non-finite values (stat_density).
```

| long | long.lower | long.upper | brief | brief.lower | brief.upper | difference | difference.lower | difference.upp |
|---|---|---|---|---|---|---|---|---|
| 0.5618581 | 0.4495512 | 0.675786 | 0.5835599 | 0.4618632 | 0.6914196 | 0.0202839 | -0.0980948 | 0.135078 |

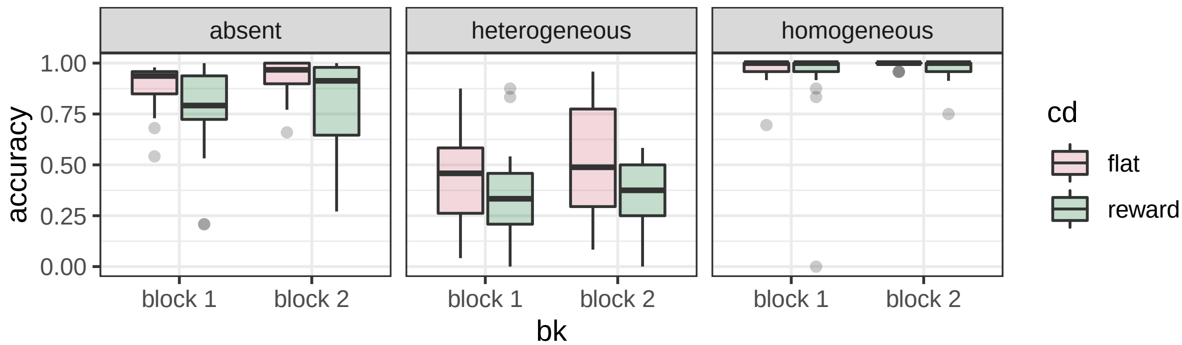
We can conclude that $p(x>0 | d) = $ 0.64055.

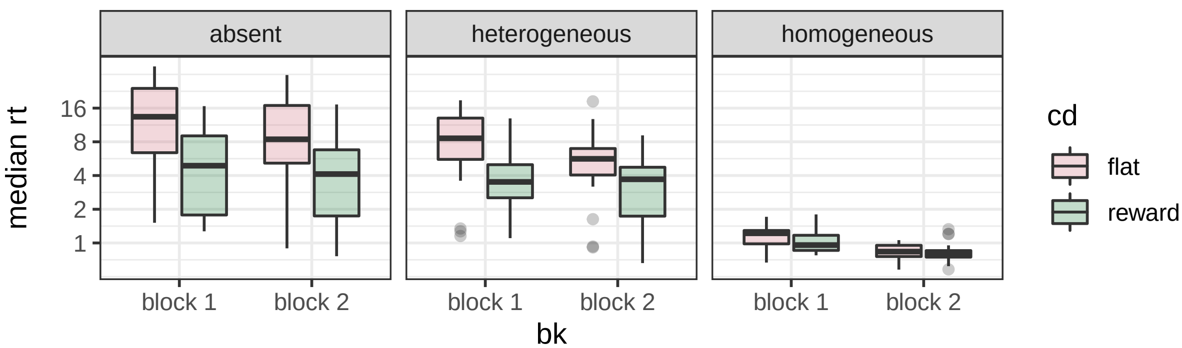
# 2 Experiment 2: Reward

## 2.1 Descriptive Statistics

### 2.1.1 Accuracy and Reaction Time

After plotting the accuracy data, incorrect trials are removed from all further analysis.



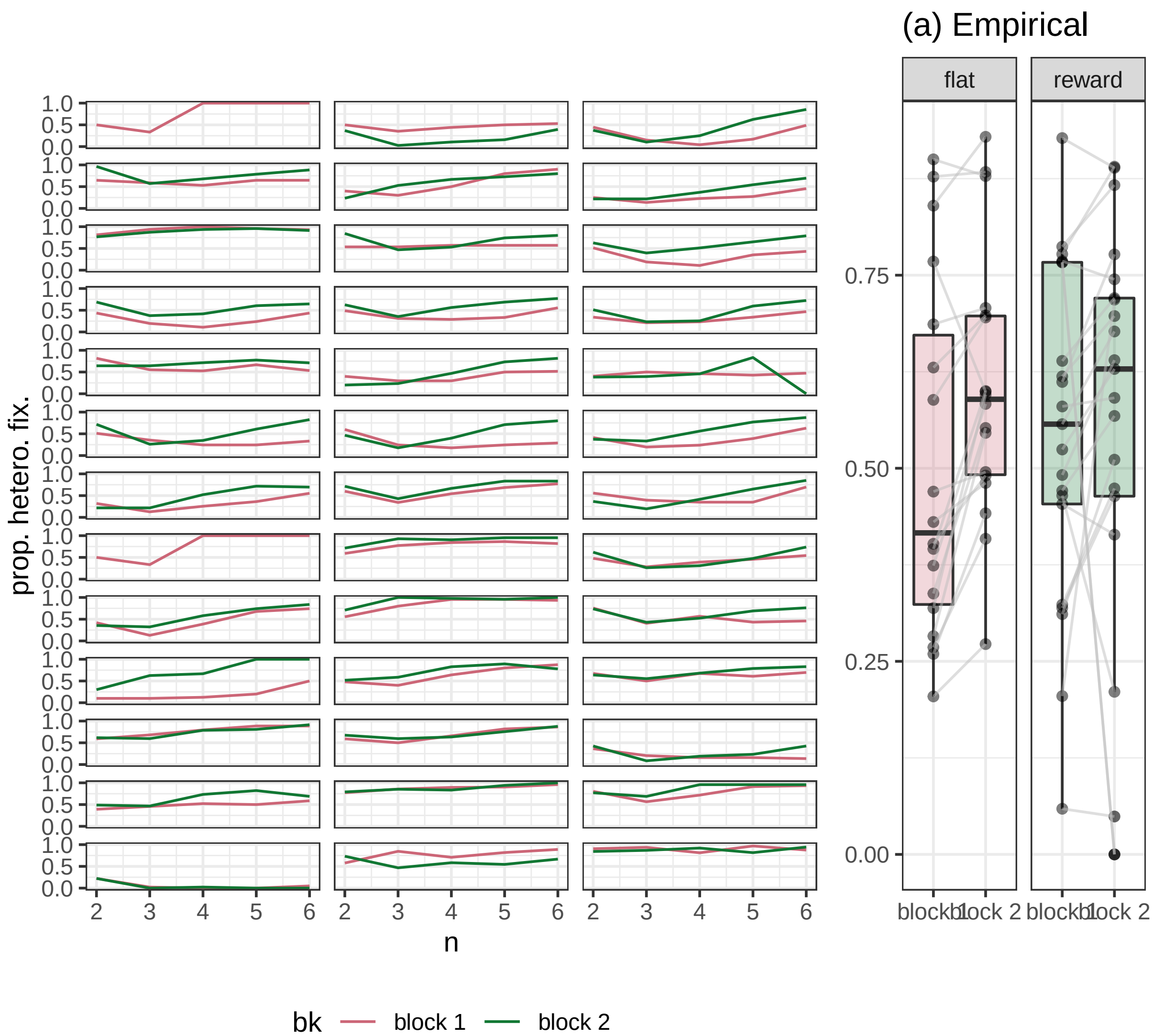

## 2.2 Saccadic Strategy

First, we need to merge (join) the fixation and accuracy data, so that we can take only correct target absent trials. We will compute the proportion of fixations to the heterogeneous side of the display for each fixation number, over all trials made by a participant.

Create a facet plot of each individual's strategy.

We can further summarise the data by creating a strategy measure, which is the proportion of all (2 - 6) fixations made by an observer over all trials.

```
## `summarise()` regrouping output by 'observer', 'bk' (override with `.groups` argument)
```




## 2.3 Bayesian Model of Saccadic Strategy

Summarise data so tha we have one strategy score per trial per observer.

Note, as beta distributions are only defined over (0, 1), values of 0 and 1 are impossible. To get around this, we will set any such values to 0.001 and 0.999 respectively.

### 2.3.1 Define Priors

We will use the same priors, and model structure, as above.

```
model_priors <- c(
    prior(normal(0, 1), class = "b"))
```

```
prior_model <- brm(
    data = d_strat,
    prop_hetero ~ 0 + cd + (cd | observer),
    sample_prior = "only",
    family = "beta",
```

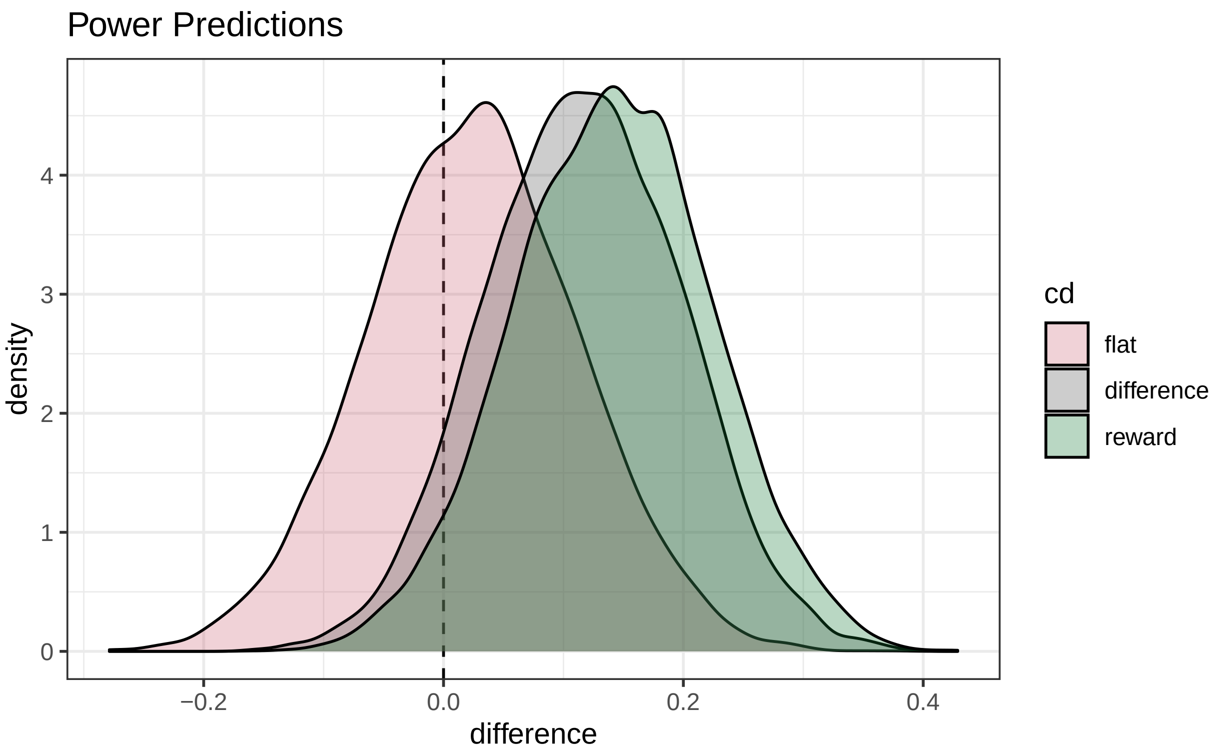
```
    prior = model_priors,
    iter = 10000,
    control = list(adapt_delta = 0.95))
```

And plot, to see if it looks reasonable.

### 2.3.2 Power Analysis

We will carry out our power analysis by simulating our experiment assuming the distributions below.

We can now plot these distributions to check that they seem reasonable.

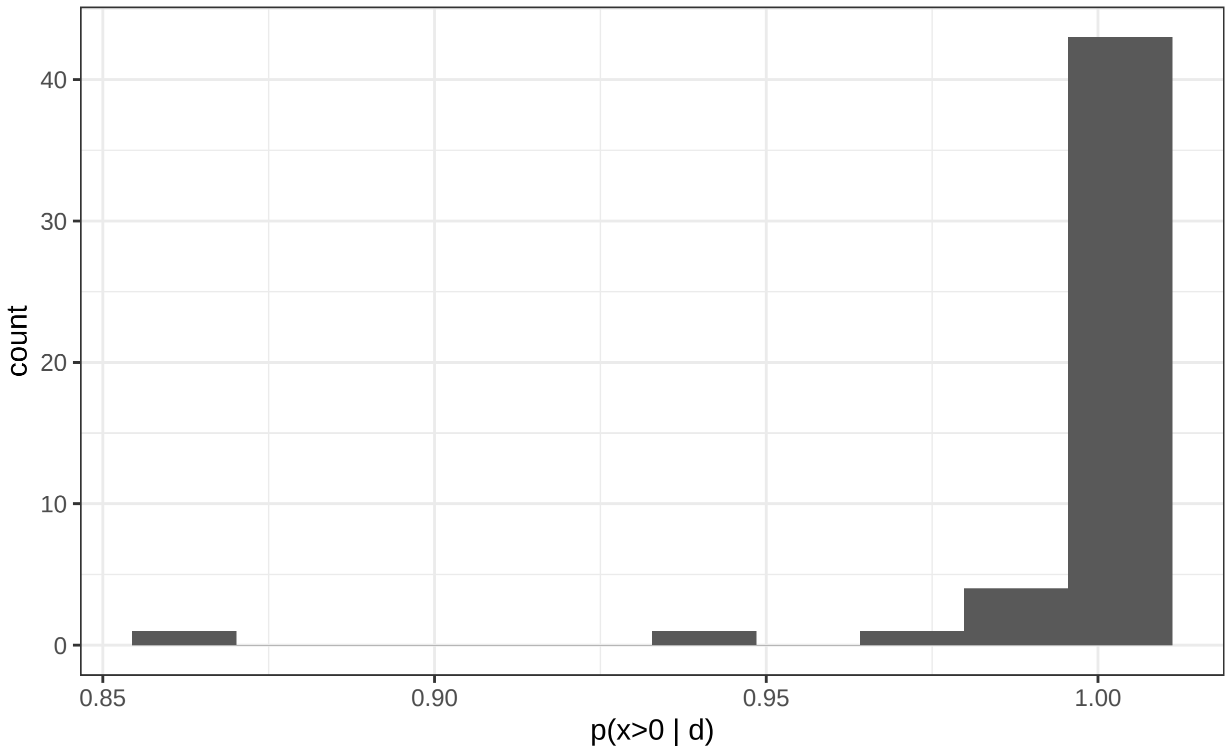



Next, we write a function to generate a simulated dataset.

Now we also need a function that will the key statistic that we are interested in: the probability, given the data, that observers were more strategic in the brief condition than the long.

Finally, we run this a number of times (50) to see the distribution of $p(\delta > 0|d)$ assuming 15 observers and 32 correct target absent trials.

And plot!



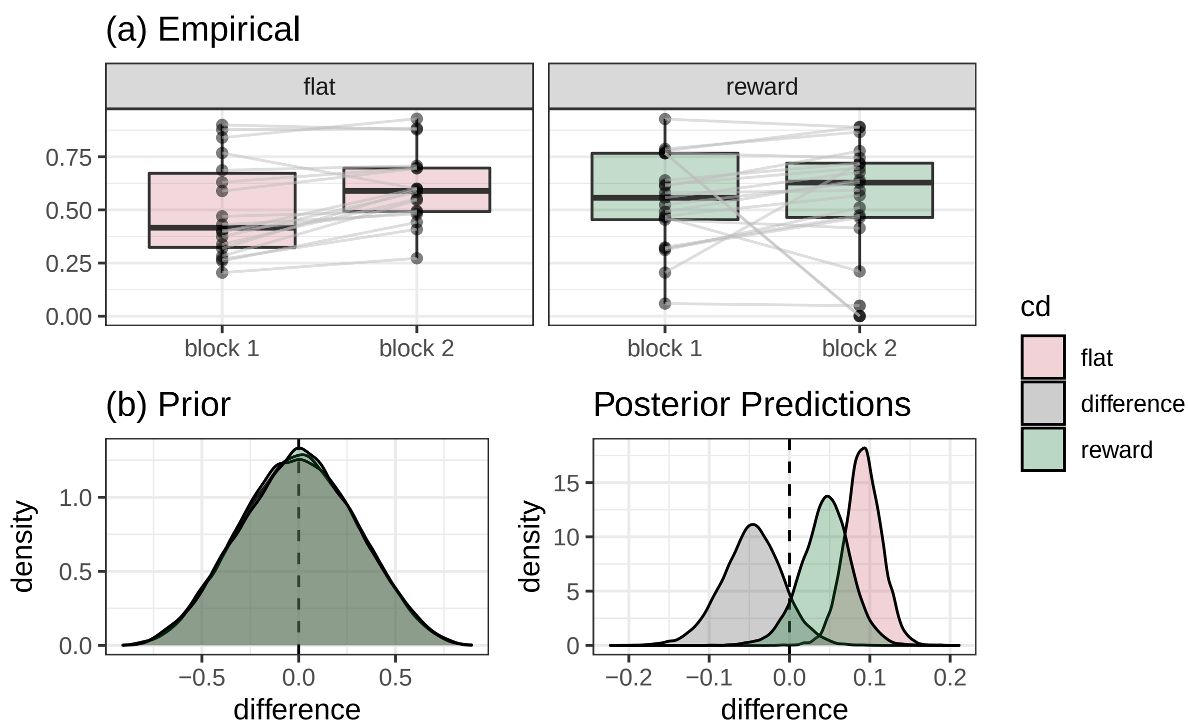

```
## # A tibble: 1 x 1
##   over_90
##     <int>
## 1      49
```

```
## # A tibble: 3 x 2
##   cd      mean_ph
##   <chr>     <dbl>
## 1 flat      0.524
## 2 initial   0.523
## 3 reward    0.595
```

### 2.4 Now refit model using data

```
##  Family: beta
##   Links: mu = logit; phi = identity
## Formula: prop_hetero ~ 0 + cd + (cd | observer)
##    Data: d_strat (Number of observations: 3025)
## Samples: 4 chains, each with iter = 10000; warmup = 5000; thin = 1;
##          total post-warmup samples = 20000
##
## Group-Level Effects:
## ~observer (Number of levels: 39)
##                          Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## sd(Intercept)                0.87      0.11     0.69     1.11 1.00     5481
## sd(cdreward)                 0.48      0.14     0.26     0.80 1.00     8168
## sd(cdflat)                   0.35      0.10     0.18     0.56 1.00    10237
## cor(Intercept,cdreward)      0.01      0.25    -0.46     0.48 1.00    11297
## cor(Intercept,cdflat)       -0.57      0.20    -0.87    -0.11 1.00    15536
## cor(cdreward,cdflat)         0.01      0.47    -0.84     0.84 1.00     2977
```

```
##                          Tail_ESS
## sd(Intercept)                8814
## sd(cdreward)                12277
## sd(cdflat)                  12701
## cor(Intercept,cdreward)     12046
## cor(Intercept,cdflat)       14843
## cor(cdreward,cdflat)         8162
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## cdinitial     0.01      0.14    -0.26     0.29 1.00     3024     5423
## cdreward      0.19      0.18    -0.17     0.55 1.00     4376     8079
## cdflat        0.38      0.13     0.12     0.65 1.00     4315     7665
##
## Family Specific Parameters:
##     Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## phi     1.86      0.04     1.78     1.95 1.00    27513    14492
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```




What is the the probability of a difference $> 0$, given the data?

```
## # A tibble: 1 x 1
##   prob_diff_greater_zero
##                    <dbl>
## 1                  0.102
```

```
## # A tibble: 1 x 12
##     flat flat.lower flat.upper reward reward.lower reward.upper difference
##    <dbl>      <dbl>      <dbl>  <dbl>        <dbl>        <dbl>      <dbl>
```

```
## 1 0.0917     0.0474      0.137 0.0462      -0.0179        0.106    -0.0460
## # ... with 5 more variables: difference.lower <dbl>,
## #   difference.upper <dbl>, .width <dbl>, .point <chr>, .interval <chr>
```

## 2.5 Original Pre-Registered Analysis

We original pre-registered an analysis plan for this experiment using frequentist statistics. The results of this planned analysis are presented here.

### 2.5.1 Accuracy

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: accuracy ~ bk * cd + (1 | observer)
##    Data: d_lmer_acc
##
## REML criterion at convergence: -137.2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.7596 -0.3331  0.0032  0.4835  2.3775
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  observer (Intercept) 0.006457 0.08036
##  Residual             0.003678 0.06065
## Number of obs: 78, groups:  observer, 39
##
## Fixed effects:
##                     Estimate Std. Error        df t value Pr(>|t|)
## (Intercept)         0.796026   0.023729 52.634479  33.547  < 2e-16 ***
## bkblock 2           0.043949   0.020215 37.000000   2.174  0.03617 *
## cdreward           -0.091066   0.032337 52.634479  -2.816  0.00683 **
## bkblock 2:cdreward -0.009082   0.027549 37.000000  -0.330  0.74350
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) bkblc2 cdrwrd
## bkblock 2   -0.426
## cdreward    -0.734  0.313
## bkblck2:cdr  0.313 -0.734 -0.426

## Type III Analysis of Variance Table with Satterthwaite's method
##         Sum Sq  Mean Sq NumDF DenDF F value   Pr(>F)
## bk    0.030104 0.030104     1    37  8.1851 0.006907 **
## cd    0.039275 0.039275     1    37 10.6788 0.002344 **
## bk:cd 0.000400 0.000400     1    37  0.1087 0.743499
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

### 2.5.2 Median Reaction Time

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: median_rt ~ bk * cd + (1 | observer)
##    Data: d_lmer_rt
##
## REML criterion at convergence: 434.4
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.2991 -0.3078 -0.0809  0.3153  3.2528
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  observer (Intercept) 26.257   5.124
##  Residual              5.393   2.322
## Number of obs: 78, groups:  observer, 39
##
## Fixed effects:
##                    Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)         11.4244     1.3260 43.8324   8.616 5.54e-11 ***
## bkblock 2           -3.4926     0.7741 37.0000  -4.512 6.30e-05 ***
## cdreward            -7.1036     1.8071 43.8324  -3.931 0.000297 ***
## bkblock 2:cdreward   2.9688     1.0549 37.0000   2.814 0.007786 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) bkblc2 cdrwrd
## bkblock 2   -0.292
## cdreward    -0.734  0.214
## bkblck2:cdr  0.214 -0.734 -0.292
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##       Sum Sq Mean Sq NumDF DenDF F value    Pr(>F)
## bk    78.174  78.174     1    37 14.4957 0.0005119 ***
## cd    57.004  57.004     1    37 10.5701 0.0024528 **
## bk:cd 42.713  42.713     1    37  7.9202 0.0077858 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: median_rt ~ bk * cd + (1 | observer)
##    Data: d_lmer_rt
##
## REML criterion at convergence: 434.4
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.2991 -0.3078 -0.0809  0.3153  3.2528
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
```

```
##  observer (Intercept) 26.257   5.124
##  Residual              5.393   2.322
## Number of obs: 78, groups:  observer, 39
##
## Fixed effects:
##                     Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)          11.4244     1.3260 43.8324   8.616 5.54e-11 ***
## bkblock 2            -3.4926     0.7741 37.0000  -4.512 6.30e-05 ***
## cdreward             -7.1036     1.8071 43.8324  -3.931 0.000297 ***
## bkblock 2:cdreward   2.9688     1.0549 37.0000   2.814 0.007786 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) bkblc2 cdrwrd
## bkblock 2   -0.292
## cdreward    -0.734  0.214
## bkblck2:cdr  0.214 -0.734 -0.292
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##       Sum Sq Mean Sq NumDF DenDF F value    Pr(>F)
## bk    78.174  78.174     1    37 14.4957 0.0005119 ***
## cd    57.004  57.004     1    37 10.5701 0.0024528 **
## bk:cd 42.713  42.713     1    37  7.9202 0.0077858 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

### 2.5.3 Search Efficiency

```
## fixed-effect model matrix is rank deficient so dropping 3 columns / coefficients
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: median_search_ef ~ bk * cd + (1 | observer)
##    Data: d_lmer_se
##
## REML criterion at convergence: 2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.0444 -0.4745  0.0963  0.4609  1.4285
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  observer (Intercept) 0.05182  0.2276
##  Residual             0.02207  0.1486
## Number of obs: 78, groups:  observer, 39
##
## Fixed effects:
##             Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)  0.50256    0.04353 50.82578  11.546 8.07e-16 ***
## bkblock 2    0.13314    0.04749 43.21037   2.804  0.00754 **
## cdreward    -0.10718    0.06224 49.94386  -1.722  0.09125 .
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##           (Intr) bkblc2
## bkblock 2 -0.274
## cdreward   0.000 -0.706
## fit warnings:
## fixed-effect model matrix is rank deficient so dropping 3 columns / coefficients
```

```
## Missing cells for: bkblock 2:cdinitial, bkblock 1:cdreward, bkblock 1:cdflat.
## Interpret type III hypotheses with care.
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##         Sum Sq  Mean Sq NumDF  DenDF F value   Pr(>F)
## bk    0.173494 0.173494     1 43.210  7.8604 0.007539 **
## cd    0.065451 0.065451     1 49.944  2.9654 0.091255 .
## bk:cd
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

### 2.5.4 Discussion

The results are consistent with the analysis presented in the paper

# 3 Time-course Analysis

I will now look to see what happens to search strategy over time: both on the scale of an individual trial, within and across blocks, and between experimental condition! First, I will fit one model to the data from both experiments.

## 3.1 Prior Predictions

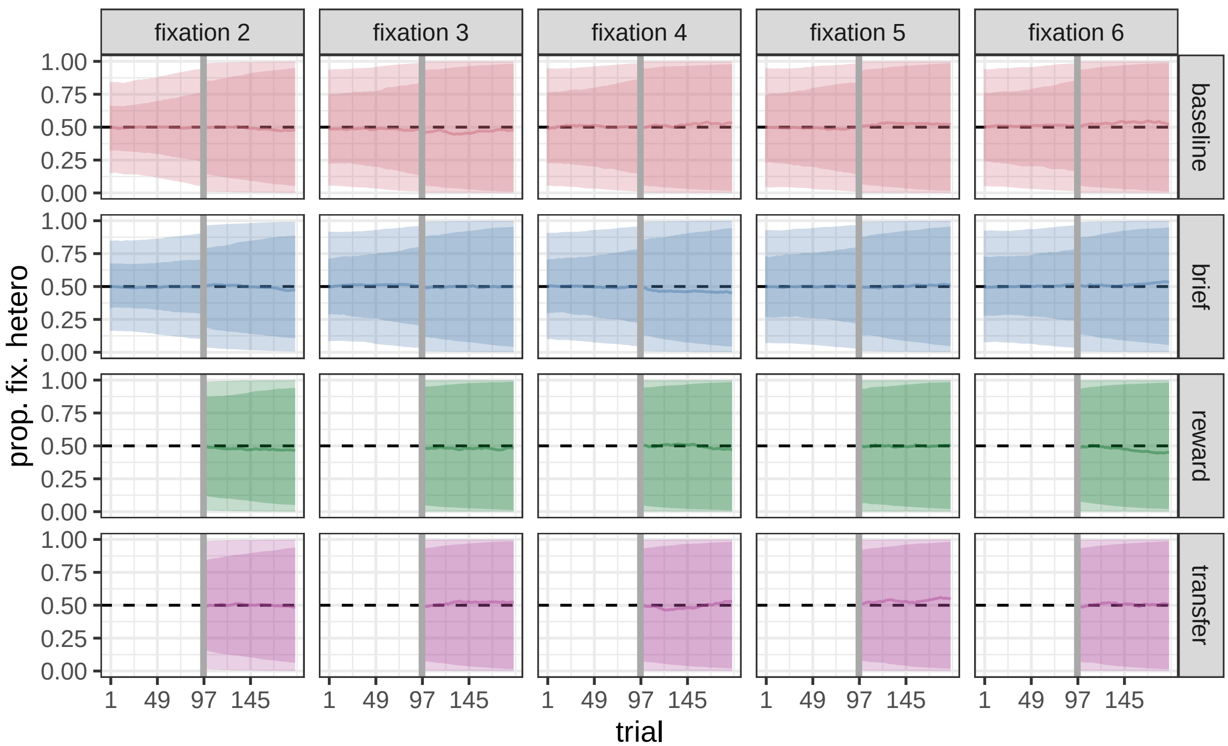
```
model_priors <- c(
    prior(normal(0, 1.0), class = "b"))

m_prior <- brm(
    data = d_strat,
    hetero_fix ~ (0 + cd) * (0 + n) * bk * ts  +
      (ts + n | observer) ,
    family = "bernoulli",
    sample_prior = "only",
    prior = model_priors,
    chains = 1)
```

```
##
## SAMPLING FOR MODEL '3df5a26144c14b1cfd8e30285a5d9b38' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0.015828 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 158.28 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
```

```
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 184.764 seconds (Warm-up)
## Chain 1:                169.48 seconds (Sampling)
## Chain 1:                354.244 seconds (Total)
## Chain 1:
```

```
saveRDS(m_prior, "models/my_prior.model")
m_prior <- readRDS("models/my_prior.model")
```

```
## Warning: Removed 10220 row(s) containing missing values (geom_path).
```




## 3.2 Posterior Predictions

```
m_posterior <- brm(
    data = d_strat,
    hetero_fix ~ (0 + cd) * (0 + n) * bk * ts  +
     (ts + n | observer) ,
```

```r
    family = "bernoulli",
    prior = model_priors,
    chains = 4)

m_posterior <- add_criterion(m_posterior, c("loo", "waic"))

saveRDS(m_posterior, "models/my_posterior.model")
m_posterior <- readRDS("models/my_posterior.model")
```

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: hetero_fix ~ (0 + cd) * (0 + n) * bk * ts + (ts + n | observer)
##    Data: d_strat (Number of observations: 21830)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup samples = 4000
##
## Group-Level Effects:
## ~observer (Number of levels: 57)
##                    Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## sd(Intercept)          0.93      0.11     0.75     1.16 1.00     1685
## sd(ts)                 0.57      0.07     0.45     0.71 1.00     1018
## sd(n3)                 0.69      0.09     0.53     0.87 1.00     2126
## sd(n4)                 0.91      0.10     0.73     1.13 1.00     1796
## sd(n5)                 0.96      0.11     0.78     1.18 1.00     1776
## sd(n6)                 0.95      0.11     0.76     1.18 1.00     2032
## cor(Intercept,ts)     -0.57      0.11    -0.75    -0.33 1.00     1287
## cor(Intercept,n3)      0.13      0.15    -0.16     0.42 1.00     1767
## cor(ts,n3)             0.20      0.15    -0.10     0.48 1.00     1429
## cor(Intercept,n4)      0.06      0.14    -0.22     0.33 1.00     1665
## cor(ts,n4)             0.20      0.13    -0.08     0.45 1.00     1374
## cor(n3,n4)             0.91      0.04     0.81     0.97 1.00     1914
## cor(Intercept,n5)     -0.10      0.14    -0.36     0.17 1.00     1748
## cor(ts,n5)             0.26      0.13    -0.01     0.51 1.00     1402
## cor(n3,n5)             0.79      0.07     0.62     0.90 1.00     1631
## cor(n4,n5)             0.94      0.03     0.87     0.98 1.00     2773
## cor(Intercept,n6)     -0.18      0.14    -0.45     0.09 1.00     2004
## cor(ts,n6)             0.25      0.14    -0.03     0.51 1.00     1567
## cor(n3,n6)             0.64      0.10     0.40     0.81 1.00     1572
## cor(n4,n6)             0.84      0.06     0.71     0.93 1.00     2321
## cor(n5,n6)             0.95      0.03     0.88     0.99 1.00     3058
##                    Tail_ESS
## sd(Intercept)          2433
## sd(ts)                 2030
## sd(n3)                 2779
## sd(n4)                 2620
## sd(n5)                 2664
## sd(n6)                 2452
## cor(Intercept,ts)      2168
## cor(Intercept,n3)      2517
## cor(ts,n3)             2550
## cor(Intercept,n4)      2428
## cor(ts,n4)             2485
## cor(n3,n4)             2658
## cor(Intercept,n5)      2569
```

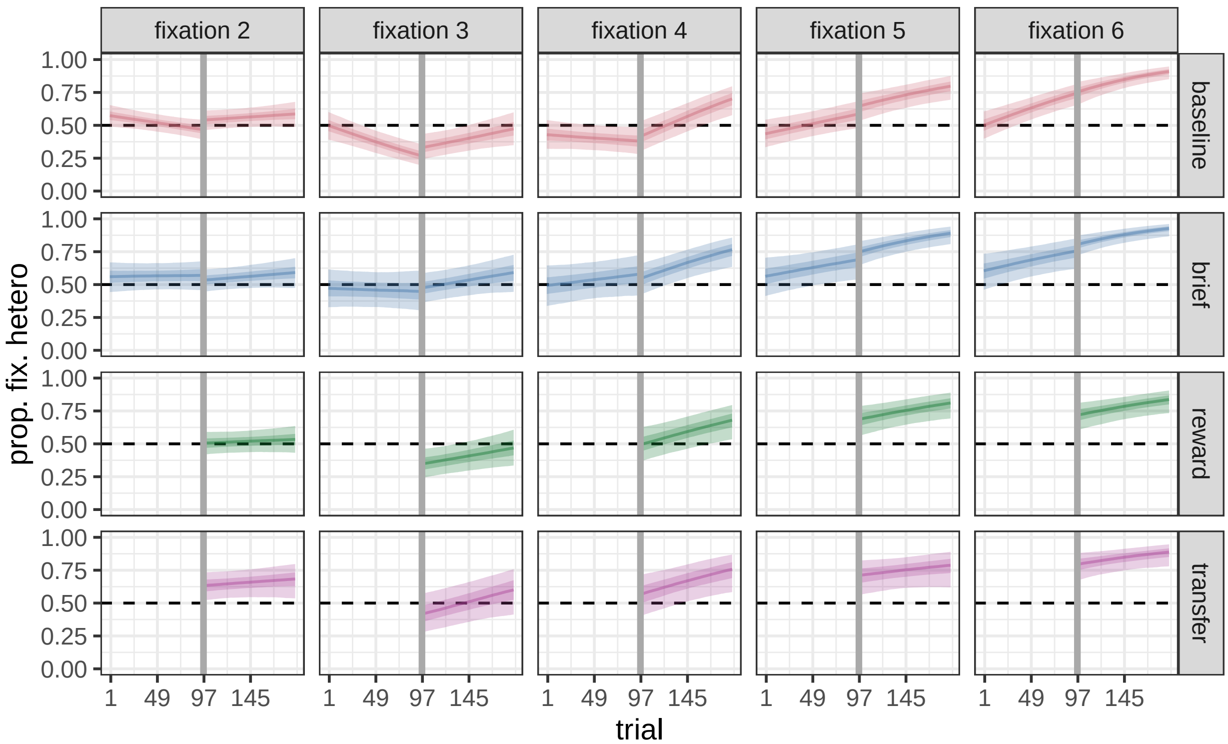
```
## cor(ts,n5)            1936
## cor(n3,n5)            2551
## cor(n4,n5)            3480
## cor(Intercept,n6)     2527
## cor(ts,n6)            2073
## cor(n3,n6)            2083
## cor(n4,n6)            3036
## cor(n5,n6)            3293
##
## Population-Level Effects:
##                         Estimate Est.Error l-95% CI u-95% CI Rhat
## cdbrief                     0.24      0.28    -0.33     0.79 1.00
## cdbaseline                  0.29      0.20    -0.10     0.69 1.00
## cdtransfer                  0.34      0.77    -1.17     1.83 1.00
## cdreward                    0.30      0.24    -0.17     0.78 1.00
## n3                         -0.36      0.26    -0.85     0.15 1.00
## n4                         -0.28      0.27    -0.80     0.26 1.00
## n5                          0.01      0.29    -0.55     0.56 1.00
## n6                          0.18      0.29    -0.38     0.76 1.00
## bkblock2                   -0.35      0.37    -1.09     0.36 1.00
## ts                          0.05      0.26    -0.47     0.54 1.00
## cdbaseline:n3               0.06      0.30    -0.53     0.63 1.00
## cdtransfer:n3              -0.47      0.80    -2.01     1.07 1.00
## cdreward:n3                 0.09      0.32    -0.54     0.73 1.00
## cdbaseline:n4              -0.30      0.32    -0.93     0.34 1.00
## cdtransfer:n4              -0.10      0.81    -1.66     1.50 1.00
## cdreward:n4                 0.44      0.34    -0.25     1.09 1.00
## cdbaseline:n5              -0.56      0.33    -1.20     0.09 1.00
## cdtransfer:n5              -0.02      0.81    -1.62     1.58 1.00
## cdreward:n5                 0.05      0.36    -0.63     0.74 1.00
## cdbaseline:n6              -0.47      0.34    -1.15     0.20 1.00
## cdtransfer:n6              -0.04      0.81    -1.62     1.57 1.00
## cdreward:n6                -0.09      0.36    -0.77     0.64 1.00
## cdbaseline:bkblock2         0.02      0.42    -0.80     0.85 1.00
## cdtransfer:bkblock2         0.34      0.77    -1.16     1.84 1.00
## cdreward:bkblock2          -0.05      0.44    -0.89     0.81 1.00
## n3:bkblock2                -0.12      0.46    -1.03     0.75 1.00
## n4:bkblock2                -0.46      0.46    -1.37     0.44 1.00
## n5:bkblock2                 0.16      0.48    -0.78     1.10 1.00
## n6:bkblock2                 0.23      0.48    -0.68     1.16 1.00
## cdbaseline:ts              -0.47      0.30    -1.07     0.13 1.00
## cdtransfer:ts              -0.01      0.74    -1.49     1.42 1.00
## cdreward:ts                -0.54      0.33    -1.17     0.11 1.00
## n3:ts                      -0.14      0.33    -0.79     0.50 1.00
## n4:ts                       0.31      0.32    -0.32     0.94 1.00
## n5:ts                       0.50      0.34    -0.16     1.18 1.00
## n6:ts                       0.66      0.33     0.02     1.31 1.00
## bkblock2:ts                 0.20      0.30    -0.39     0.78 1.00
## cdbaseline:n3:bkblock2     -0.85      0.57    -1.95     0.24 1.00
## cdtransfer:n3:bkblock2     -0.48      0.80    -2.04     1.09 1.00
## cdreward:n3:bkblock2       -0.67      0.57    -1.77     0.42 1.00
## cdbaseline:n4:bkblock2     -0.47      0.56    -1.56     0.62 1.00
## cdtransfer:n4:bkblock2     -0.10      0.79    -1.60     1.46 1.00
## cdreward:n4:bkblock2       -0.41      0.58    -1.54     0.72 1.00
```

```
## cdbaseline:n5:bkblock2         0.23      0.57    -0.87     1.34 1.00
## cdtransfer:n5:bkblock2         0.00      0.81    -1.58     1.54 1.00
## cdreward:n5:bkblock2          -0.02      0.60    -1.21     1.15 1.00
## cdbaseline:n6:bkblock2         0.01      0.60    -1.18     1.18 1.00
## cdtransfer:n6:bkblock2        -0.06      0.82    -1.62     1.57 1.00
## cdreward:n6:bkblock2           0.02      0.62    -1.22     1.21 1.00
## cdbaseline:n3:ts              -0.43      0.40    -1.19     0.34 1.00
## cdtransfer:n3:ts               0.14      0.75    -1.35     1.62 1.00
## cdreward:n3:ts                -0.36      0.42    -1.18     0.48 1.00
## cdbaseline:n4:ts              -0.09      0.38    -0.84     0.65 1.00
## cdtransfer:n4:ts              -0.06      0.72    -1.47     1.35 1.00
## cdreward:n4:ts                -0.87      0.42    -1.69    -0.04 1.00
## cdbaseline:n5:ts               0.53      0.40    -0.25     1.30 1.00
## cdtransfer:n5:ts              -0.30      0.75    -1.72     1.16 1.00
## cdreward:n5:ts                -0.11      0.43    -0.98     0.74 1.00
## cdbaseline:n6:ts               0.82      0.39     0.05     1.61 1.00
## cdtransfer:n6:ts              -0.20      0.73    -1.66     1.24 1.00
## cdreward:n6:ts                 0.06      0.44    -0.77     0.95 1.00
## cdbaseline:bkblock2:ts         0.43      0.35    -0.24     1.12 1.00
## cdtransfer:bkblock2:ts        -0.01      0.73    -1.44     1.45 1.00
## cdreward:bkblock2:ts           0.42      0.36    -0.28     1.10 1.00
## n3:bkblock2:ts                 0.37      0.38    -0.38     1.11 1.00
## n4:bkblock2:ts                 0.47      0.37    -0.29     1.21 1.00
## n5:bkblock2:ts                 0.28      0.40    -0.52     1.08 1.00
## n6:bkblock2:ts                 0.22      0.40    -0.57     0.99 1.00
## cdbaseline:n3:bkblock2:ts      0.61      0.46    -0.30     1.52 1.00
## cdtransfer:n3:bkblock2:ts      0.16      0.74    -1.27     1.63 1.00
## cdreward:n3:bkblock2:ts        0.52      0.47    -0.39     1.45 1.00
## cdbaseline:n4:bkblock2:ts      0.33      0.45    -0.56     1.19 1.00
## cdtransfer:n4:bkblock2:ts     -0.06      0.73    -1.47     1.33 1.00
## cdreward:n4:bkblock2:ts        0.76      0.48    -0.18     1.69 1.00
## cdbaseline:n5:bkblock2:ts     -0.71      0.47    -1.63     0.20 1.00
## cdtransfer:n5:bkblock2:ts     -0.28      0.74    -1.74     1.20 1.00
## cdreward:n5:bkblock2:ts       -0.10      0.50    -1.10     0.85 1.00
## cdbaseline:n6:bkblock2:ts     -0.69      0.47    -1.61     0.22 1.00
## cdtransfer:n6:bkblock2:ts     -0.18      0.73    -1.65     1.26 1.00
## cdreward:n6:bkblock2:ts       -0.36      0.51    -1.36     0.65 1.00
##                           Bulk_ESS Tail_ESS
## cdbrief                       1529     2235
## cdbaseline                    1136     1685
## cdtransfer                    5938     3218
## cdreward                      1690     2254
## n3                            1985     2881
## n4                            1632     2413
## n5                            1648     2685
## n6                            1858     2658
## bkblock2                      1859     2831
## ts                            1755     2408
## cdbaseline:n3                 1983     2627
## cdtransfer:n3                 5930     2581
## cdreward:n3                   2068     2697
## cdbaseline:n4                 1781     2431
## cdtransfer:n4                 6809     2998
## cdreward:n4                   1979     2744
```

```
## cdbaseline:n5                  1773      2889
## cdtransfer:n5                  7153      2801
## cdreward:n5                    1847      2302
## cdbaseline:n6                  2150      3029
## cdtransfer:n6                  6832      3005
## cdreward:n6                    2295      2693
## cdbaseline:bkblock2            2073      2632
## cdtransfer:bkblock2            5472      3046
## cdreward:bkblock2              2128      2988
## n3:bkblock2                    3382      3253
## n4:bkblock2                    3591      3195
## n5:bkblock2                    3561      3019
## n6:bkblock2                    3860      3265
## cdbaseline:ts                  1819      2700
## cdtransfer:ts                  5504      3063
## cdreward:ts                    1791      2375
## n3:ts                          2586      2962
## n4:ts                          2602      3071
## n5:ts                          2301      2919
## n6:ts                          2856      3117
## bkblock2:ts                    1932      2807
## cdbaseline:n3:bkblock2         4244      3432
## cdtransfer:n3:bkblock2         6971      3221
## cdreward:n3:bkblock2           4304      3144
## cdbaseline:n4:bkblock2         3607      3081
## cdtransfer:n4:bkblock2         6785      3169
## cdreward:n4:bkblock2           3995      3161
## cdbaseline:n5:bkblock2         4121      3413
## cdtransfer:n5:bkblock2         5866      2765
## cdreward:n5:bkblock2           4065      3410
## cdbaseline:n6:bkblock2         4143      3064
## cdtransfer:n6:bkblock2         6683      2859
## cdreward:n6:bkblock2           4566      2972
## cdbaseline:n3:ts               2818      2989
## cdtransfer:n3:ts               6242      2965
## cdreward:n3:ts                 2808      3118
## cdbaseline:n4:ts               2790      2725
## cdtransfer:n4:ts               6367      3228
## cdreward:n4:ts                 3074      3507
## cdbaseline:n5:ts               2639      2857
## cdtransfer:n5:ts               5117      2845
## cdreward:n5:ts                 2652      2461
## cdbaseline:n6:ts               3262      3168
## cdtransfer:n6:ts               6407      3217
## cdreward:n6:ts                 3367      3298
## cdbaseline:bkblock2:ts         2264      2788
## cdtransfer:bkblock2:ts         6190      2884
## cdreward:bkblock2:ts           2189      3141
## n3:bkblock2:ts                 2630      3021
## n4:bkblock2:ts                 2962      2823
## n5:bkblock2:ts                 2591      2886
## n6:bkblock2:ts                 3243      3463
## cdbaseline:n3:bkblock2:ts      3242      3190
## cdtransfer:n3:bkblock2:ts      6197      3244
```

```
## cdreward:n3:bkblock2:ts       3214     3329
## cdbaseline:n4:bkblock2:ts     3428     2979
## cdtransfer:n4:bkblock2:ts     6603     3368
## cdreward:n4:bkblock2:ts       2864     2779
## cdbaseline:n5:bkblock2:ts     3226     3061
## cdtransfer:n5:bkblock2:ts     5368     2545
## cdreward:n5:bkblock2:ts       3028     2713
## cdbaseline:n6:bkblock2:ts     3852     3383
## cdtransfer:n6:bkblock2:ts     6255     2963
## cdreward:n6:bkblock2:ts       3494     3051
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
## Warning: Removed 10220 row(s) containing missing values (geom_path).
```




```
## Warning: Removed 10220 row(s) containing missing values (geom_path).
```

```
##    observer                 cd             bk                    t
##  Length:21830       brief    :3509   Length:21830       Min.   :  1.00
##  Class :character   baseline:9378   Class :character   1st Qu.: 49.00
##  Mode  :character   transfer:2026   Mode  :character   Median : 97.00
##                     reward  :6917                      Mean   : 97.23
##                                                        3rd Qu.:145.00
##                                                        Max.   :192.00
##  n          hetero_fix           ts           targ_side
##  2:4545   Min.   :0.0000   Min.   :0.000   Length:21830
##  3:4486   1st Qu.:0.0000   1st Qu.:0.500   Class :character
##  4:4387   Median :1.0000   Median :1.000   Mode  :character
##  5:4258   Mean   :0.5592   Mean   :1.002
##  6:4154   3rd Qu.:1.0000   3rd Qu.:1.500
```

| x | x | x | x | x | x |
|---|---|---|---|---|---|
| 0.3917148 | 0.0593147 | 7.16e-05 | 0.4776962 | 0.0712027 | 80 |
| | | | | | 40 |
| | | | | | 40 |
| | | | | | 20 |
| | | | | | 16 |

```
##           Max.   :1.0000   Max.   :1.990
##        rt              acc
##  Min.   : 0.271   Min.   :1
##  1st Qu.: 2.841   1st Qu.:1
##  Median : 6.401   Median :1
##  Mean   : 9.645   Mean   :1
##  3rd Qu.:13.027   3rd Qu.:1
##  Max.   :53.535   Max.   :1
```

### 3.3 Can we easily simplify the model?

I now fit simpler models, removing either one of the four variables, or the four-way interaction.

```
## Compiling the C++ model

## Start sampling

## Compiling the C++ model

## recompiling to avoid crashing R session

## Start sampling

## Compiling the C++ model

## recompiling to avoid crashing R session

## Start sampling

## Warning in mw$n = c(sum(str_count(get_variables(m_posterior), "b_")),
## sum(str_count(get_variables(m_posterior_drop_bk), : Coercing LHS to a list
```

# 4 Session Info

```
## R version 3.6.1 (2019-07-05)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS Mojave 10.14.6
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_GB.UTF-8/en_GB.UTF-8/en_GB.UTF-8/C/en_GB.UTF-8/en_GB.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
```

```
## other attached packages:
##  [1] RcppRoll_0.3.0  lmerTest_3.1-2  lme4_1.1-23     Matrix_1.2-17  
##  [5] patchwork_1.0.0 tidybayes_2.0.2 forcats_0.4.0   stringr_1.4.0  
##  [9] dplyr_1.0.0     purrr_0.3.4     readr_1.3.1     tidyr_1.1.0    
## [13] tibble_3.0.1    ggplot2_3.3.2   tidyverse_1.2.1 brms_2.12.0    
## [17] Rcpp_1.0.5     
## 
## loaded via a namespace (and not attached):
##   [1] minqa_1.2.4              colorspace_1.4-1        
##   [3] ellipsis_0.3.1           ggridges_0.5.1          
##   [5] rsconnect_0.8.15         markdown_1.1            
##   [7] base64enc_0.1-3          rstudioapi_0.11         
##   [9] farver_2.0.3             rstan_2.19.3            
##  [11] svUnit_0.7-12            DT_0.14                 
##  [13] fansi_0.4.1              lubridate_1.7.4         
##  [15] xml2_1.2.2               codetools_0.2-16        
##  [17] bridgesampling_0.7-2     splines_3.6.1           
##  [19] knitr_1.25               shinythemes_1.1.2       
##  [21] bayesplot_1.7.0          jsonlite_1.7.0          
##  [23] nloptr_1.2.2.1           broom_0.5.2             
##  [25] shiny_1.3.2              compiler_3.6.1          
##  [27] httr_1.4.1               backports_1.1.8         
##  [29] assertthat_0.2.1         cli_2.0.2               
##  [31] later_0.8.0              htmltools_0.3.6         
##  [33] prettyunits_1.1.1        tools_3.6.1             
##  [35] igraph_1.2.4.1           coda_0.19-3             
##  [37] gtable_0.3.0             glue_1.4.1              
##  [39] reshape2_1.4.3           ggthemes_4.2.0          
##  [41] cellranger_1.1.0         vctrs_0.3.1             
##  [43] nlme_3.1-140             crosstalk_1.0.0         
##  [45] xfun_0.8                 ps_1.3.3                
##  [47] rvest_0.3.4              mime_0.7                
##  [49] miniUI_0.1.1.1           lifecycle_0.2.0         
##  [51] gtools_3.8.1             statmod_1.4.34          
##  [53] MASS_7.3-51.4            zoo_1.8-6               
##  [55] scales_1.1.1             colourpicker_1.0        
##  [57] hms_0.5.0                promises_1.0.1          
##  [59] Brobdingnag_1.2-6        parallel_3.6.1          
##  [61] inline_0.3.15            shinystan_2.5.0         
##  [63] yaml_2.2.0               gridExtra_2.3           
##  [65] loo_2.3.0                StanHeaders_2.21.0-5    
##  [67] stringi_1.4.6            dygraphs_1.1.1.6        
##  [69] checkmate_2.0.0          boot_1.3-22             
##  [71] pkgbuild_1.0.8           rlang_0.4.6             
##  [73] pkgconfig_2.0.3          matrixStats_0.56.0      
##  [75] HDInterval_0.2.0         evaluate_0.14           
##  [77] lattice_0.20-38          labeling_0.3            
##  [79] rstantools_2.0.0         htmlwidgets_1.3         
##  [81] processx_3.4.3           tidyselect_1.1.0        
##  [83] plyr_1.8.5               magrittr_1.5            
##  [85] bookdown_0.18            R6_2.4.1                
##  [87] generics_0.0.2           pillar_1.4.4            
##  [89] haven_2.1.1              withr_2.2.0             
##  [91] xts_0.11-2               abind_1.4-5             
```

```
##  [93] modelr_0.1.5              crayon_1.3.4
##  [95] arrayhelpers_1.0-20160527 utf8_1.1.4
##  [97] rmarkdown_2.1             grid_3.6.1
##  [99] readxl_1.3.1              callr_3.4.3
## [101] threejs_0.3.1             digest_0.6.25
## [103] xtable_1.8-4              numDeriv_2016.8-1.1
## [105] httpuv_1.5.1              RcppParallel_5.0.2
## [107] stats4_3.6.1              munsell_0.5.0
## [109] shinyjs_1.0
```