

Divisive Normalization

Comparing Heeger Model with our model Start with Heeger Model: Set some parameters to fit values:

```
c_50 <- 13.1  
n <- 1.5
```

Create contrasts between 1 and 100:

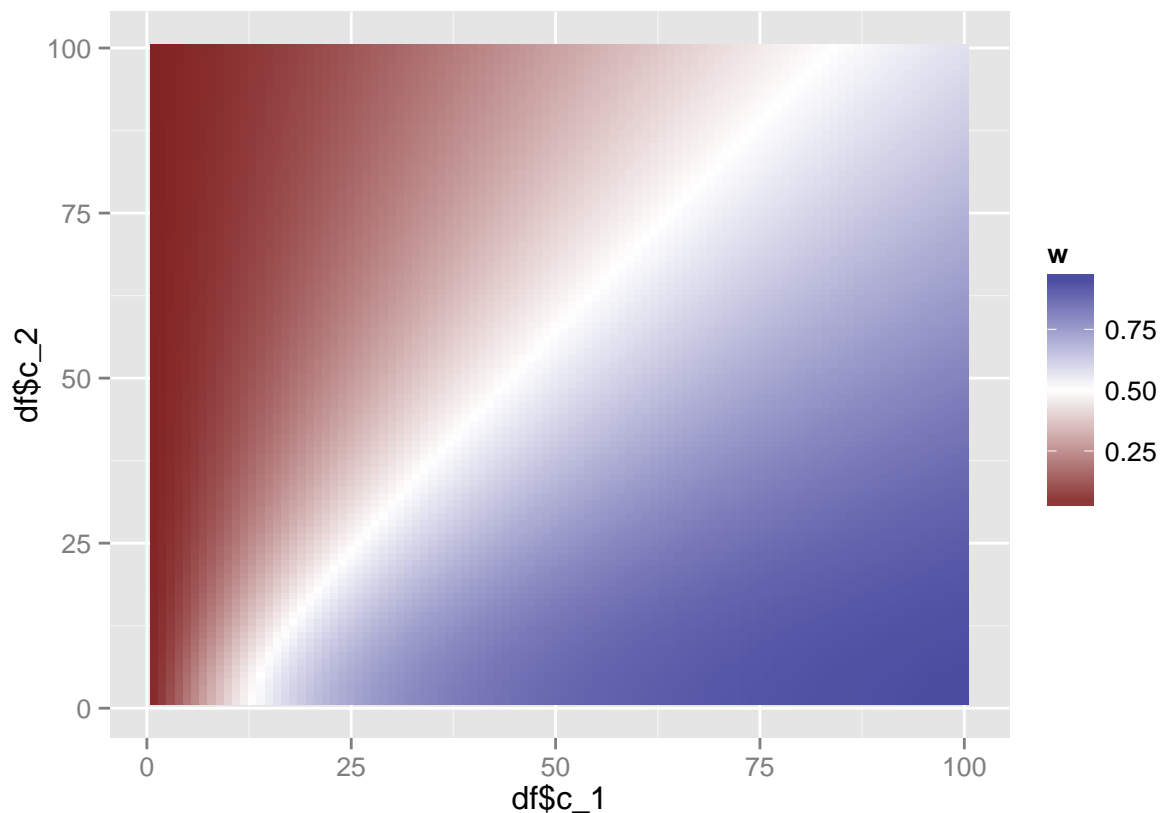
```
c_1 <- 1:100  
c_2 <- 1:100
```

Here's a function that gives us the weights from Heeger's model:

```
heeger <- function(c_1, c_2, c_50, n)  
{  
  w <- (c_1^n)/(c_50^n + (c_1^2 + c_2^2)^(n/2))  
  w  
}
```

Let's plot weights for different values of c_1 and c_2:

```
df <- expand.grid(c_1=c_1, c_2=c_2)  
df$w <- heeger(df$c_1, df$c_2, c_50, n)  
library(ggplot2)  
p <- qplot(df$c_1, df$c_2, data = df, fill = df$w, geom = "raster")  
p + scale_fill_gradient2(name = "w", midpoint = .5)
```



Our model: Initialize some values (SDs):

```
s <- 19
s_c <- 19
```

Computes weight for a given x_1 and x_2 :

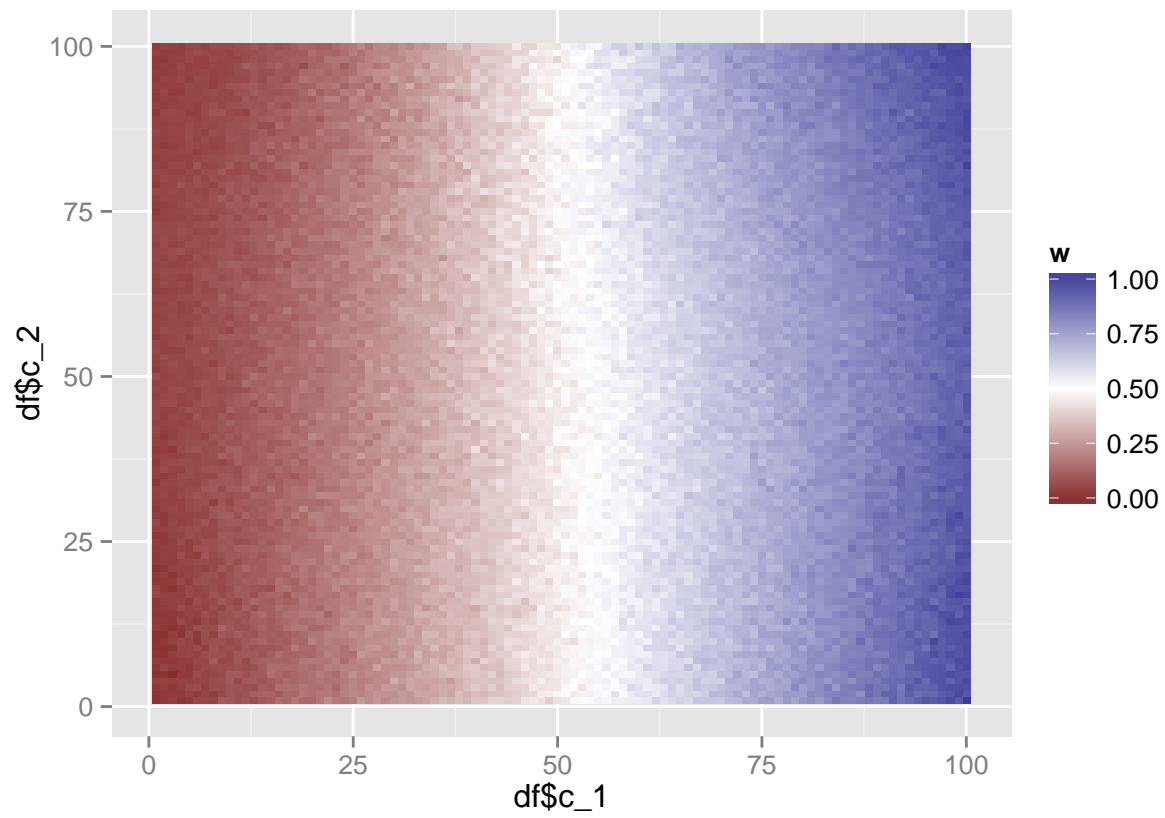
```
ma_halpern <- function(x_1, x_2, s, s_c)
{
  norm = dnorm(x_1, sd = s) * dnorm(x_2, sd=s) + dnorm(x_1, sd=s) + dnorm(x_2, sd=s) + dnorm(x_1, mean=
  w_2 = (x_1/s^2 + x_2/(s^2 + 2*s_c^2))/(1/s^2 + 1/(s^2 + 2*s_c^2))
  num = x_1 * dnorm(x_2, sd=s) + w_2 * dnorm(x_1, mean=x_2, sd=sqrt(2*s^2 + 2*s_c^2))
  w = num/norm
}
```

Now let's estimate the weights for a given c_1 and c_2 by simulating many x values: (Means are normalized so they will be on the same scale as Heeger... maybe this is not the right thing to do though?)

```
estimate_weights <- function(c_1, c_2, s, s_c)
{
  x_1s = rnorm(100, mean=c_1, sd=s)
  x_2s = rnorm(100, mean=c_2, sd=s)
  w = ma_halpern(x_1s, x_2s, s, s_c)
  mean(w/100)
}
vew <- Vectorize(estimate_weights, c("c_1", "c_2"))
```

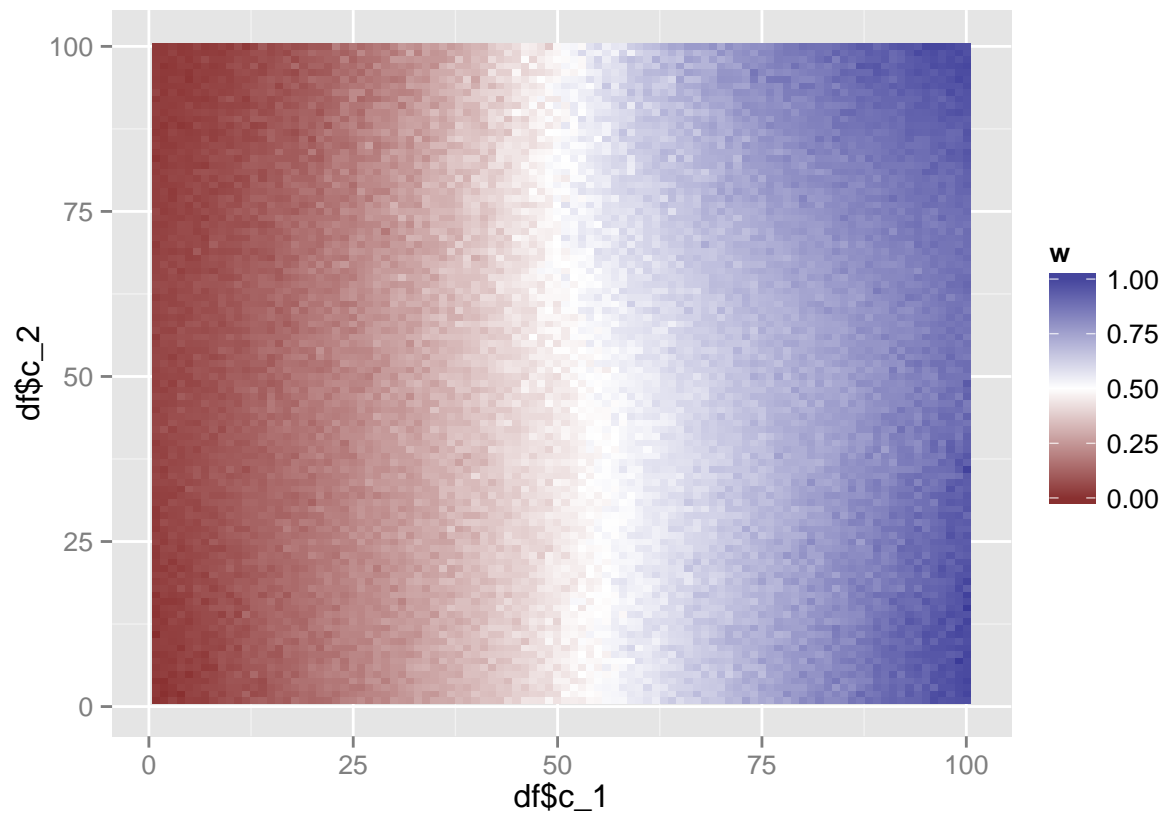
And plot them:

```
df <- expand.grid(c_1=c_1, c_2=c_2)
df$w <- vew(df$c_1, df$c_2, s, s_c)
p <- qplot(df$c_1, df$c_2, data = df, fill = df$w, geom = "raster")
p + scale_fill_gradient2(name = "w", midpoint = .5)
```



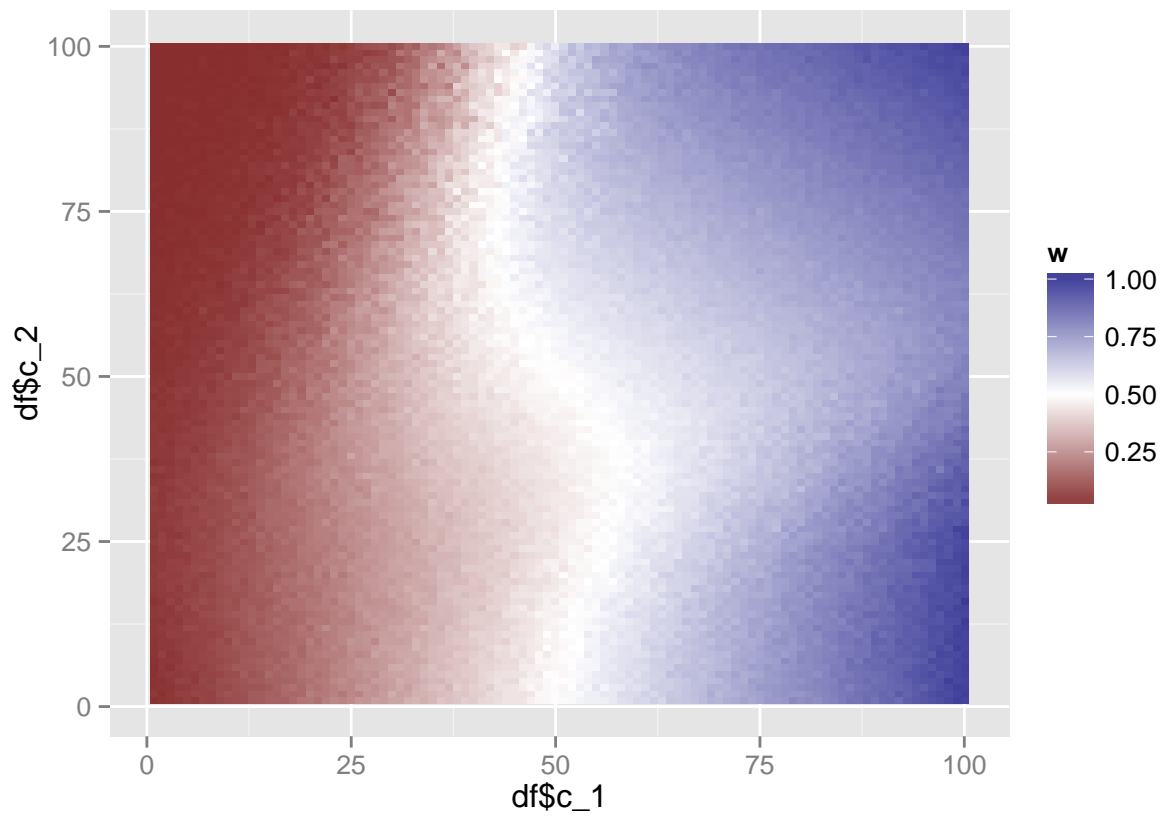
Lets try some less noisy data to get a better view of the function:

```
s_c = 1
```

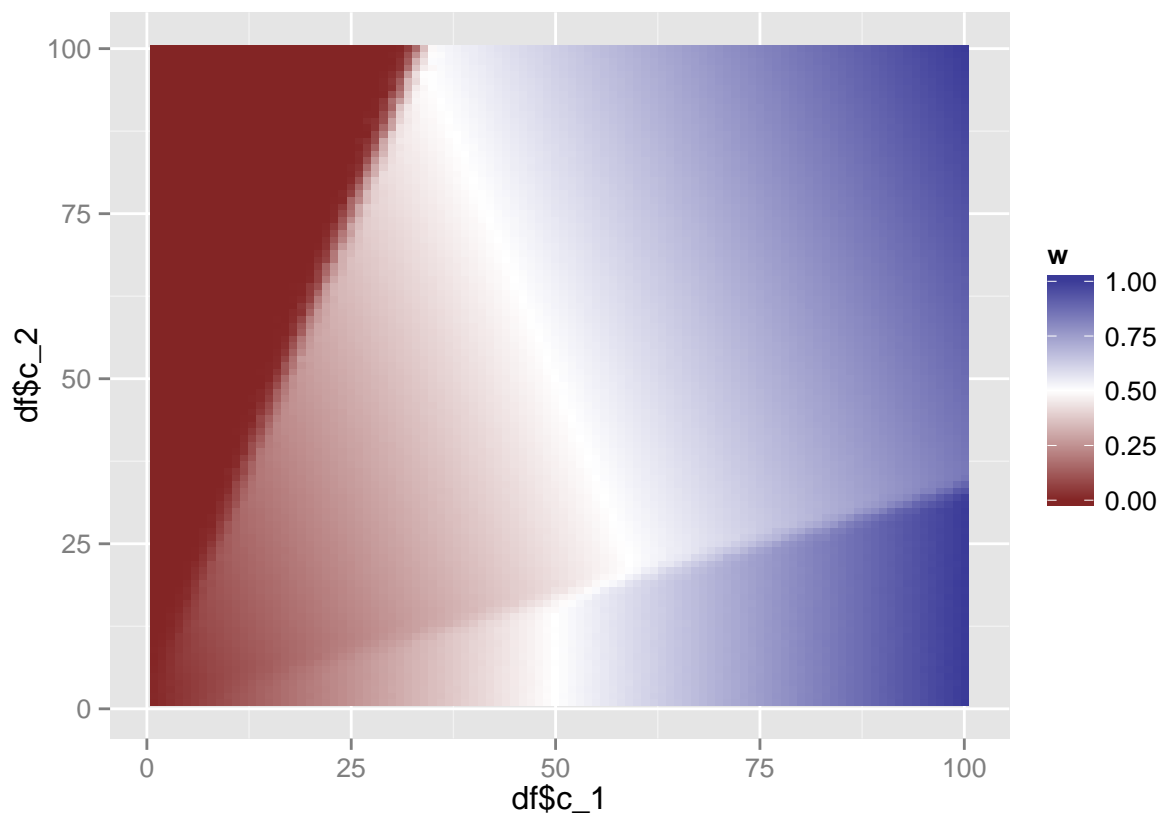


What happens as vision improves?

$s = 10$

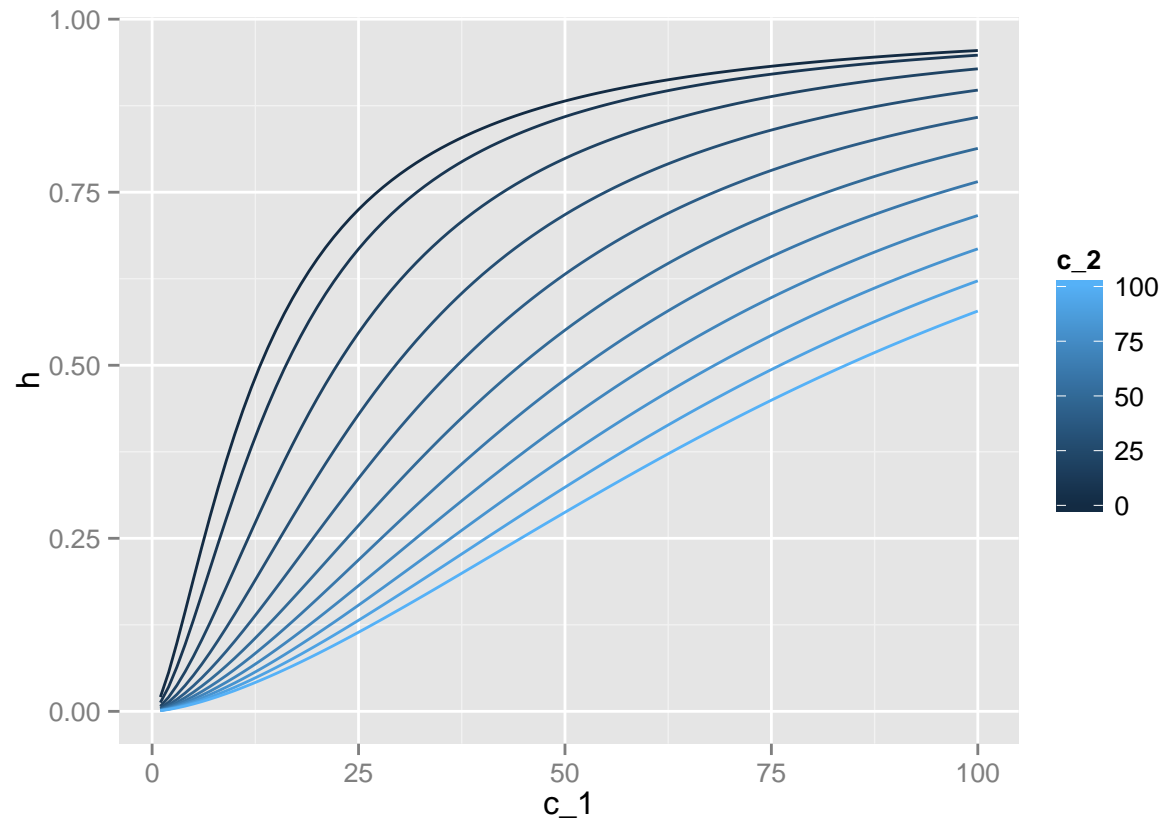


$s = 1$

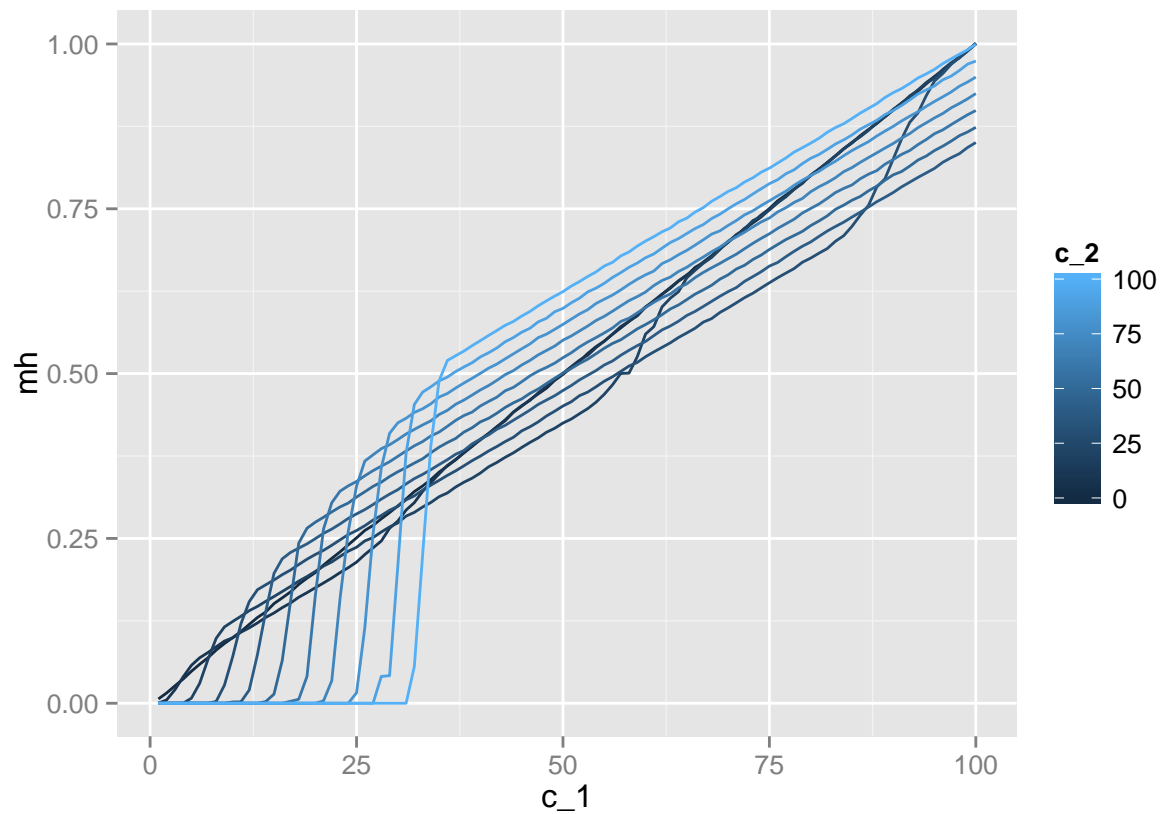


Weights as a function of c_1 for various c_2 values

```
c_2w <- seq(0, 100, 10)
dfw <- expand.grid(c_1=c_1, c_2=c_2w)
dfw$h <- heeger(dfw$c_1, dfw$c_2, c_50, n)
dfw$mh <- vew(dfw$c_1, dfw$c_2, s, s_c)
ggplot(dfw, aes(x=c_1, y=h, group=c_2, colour=c_2)) + geom_line()
```



```
ggplot(dfw, aes(x=c_1, y=mh, group=c_2, colour=c_2)) + geom_line()
```



Attempts to fit `ma_halpern` to `heeger` using Root Mean Squared Error

```
rmse <- function(s)
{
  sig <- s * 100
  df$mh <- vew(df$c_1, df$c_2, sig[1], sig[2])
  sqrt(mean(df$h-df$mh)^2)
}

df <- expand.grid(c_1=c_1, c_2=c_2)
df$h <- heeger(df$c_1, df$c_2, c_50, n)
sigs <- matrix(, 1, 2)
s <- runif(2)
o <- optimx(s, fn=rmse, lower=c(0.001,0.001), upper=c(1,1))
sigs <- c(coef(o[1,]))
for(i in 1:100)
{
  s <- runif(2)
  o <- optimx(s, fn=rmse, lower=c(0.001,0.001), upper=c(1,1))
  sigs <- rbind(sigs, c(coef(o[1,])))
}
```

Output: [,1] [,2] ———— | ————

```
0.102474341 0.99999366
0.001073785 0.17760943
0.001331999 0.05370380
0.001000000 0.12245692
```

| | |
|-------------|-------------|
| 1.000000000 | 1.000000000 |
| 0.523826106 | 1.000000000 |
| 0.001000000 | 0.68692467 |
| 0.034453589 | 0.97508780 |
| 0.035796153 | 0.70151735 |
| 0.235620318 | 0.56200022 |
| 0.001000000 | 0.10929099 |
| 0.001000000 | 1.000000000 |
| 0.001000000 | 1.000000000 |
| 0.060842346 | 0.73784384 |
| 0.877873887 | 0.81970844 |
| 0.001000000 | 0.99999210 |
| 0.002070543 | 0.39062208 |
| 0.001000000 | 0.91017253 |
| 0.588303091 | 0.64327413 |
| 0.019842521 | 0.84865678 |
| 0.380784704 | 0.97671970 |
| 0.001000000 | 0.20374478 |
| 0.162126114 | 0.70389242 |
| NA | NA |
| 0.130002015 | 0.94395401 |
| 0.001000000 | 0.62332399 |
| 0.001000000 | 0.91877175 |
| 0.752145685 | 0.75643039 |
| 0.664027868 | 0.60621882 |
| 0.001397428 | 0.04557056 |
| 0.036321368 | 0.97444112 |
| 0.067691674 | 0.94630337 |
| 0.001000000 | 0.26583646 |
| 0.194855734 | 0.94667137 |
| 0.001000000 | 0.78815492 |
| 0.165732899 | 0.94983276 |
| 0.209202475 | 0.77716948 |
| NA | NA |
| 0.005144855 | 0.52347423 |
| NA | NA |
| 0.185466911 | 1.000000000 |
| 0.305819857 | 0.71238889 |
| 0.042294377 | 0.97879865 |
| 0.001008274 | 0.60425602 |
| 0.001000000 | 0.38168447 |
| 0.003483951 | 0.16990160 |
| 0.001000000 | 0.08226764 |
| 0.884808321 | 0.50784126 |
| 0.676000128 | 0.76220164 |
| 0.001000000 | 0.40243909 |
| 0.001864401 | 0.52082322 |
| 0.001307906 | 1.000000000 |
| 0.001000000 | 0.09400947 |
| 0.001000000 | 0.07699153 |
| 0.001000000 | 0.04995524 |
| 0.001917776 | 0.08466055 |
| 0.001000000 | 0.88712935 |
| 0.001705685 | 0.16506352 |


```

0.001000000 0.38012771
0.001000000 0.67266465
0.168197220 0.91529773
0.001000000 0.40688204
0.687467534 0.71330363
0.551109082 1.00000000
0.001000000 0.80188037
0.001000000 0.13530647
0.014786885 0.21743388
0.001000000 0.98298715
0.976375711 0.89341372
0.339211515 0.99094302
0.162931799 0.16748831
0.586377455 0.75871343
0.005731327 0.21324106
0.802124335 0.98552978
0.001000000 0.14104017
0.001000000 0.46379863
0.001000000 0.43688628
0.750813197 0.78920705
0.001000000 1.00000000
0.001061592 0.83566690
0.027032159 0.30080081
      NA      NA
0.131404365 0.85659281
      NA      NA
0.242076534 0.82647055
0.152611767 0.61058017
      NA      NA
0.002080649 0.41614029
0.268077675 0.65690376
0.001000000 0.91871726
0.001000000 0.12695971
0.585379743 0.14646889
0.066418728 0.98325028
0.016374681 0.87123387
0.001000000 0.49615733
0.555834204 1.00000000
0.001000000 0.38922631
0.660527648 0.12988535
0.679869320 1.00000000
      NA      NA
0.459913174 0.83963203

```

```

sds <- sigs[which(sigs[,1]!="NA" & sigs[,2]!="NA"),]
cov(sds)

```

```

      [,1]      [,2]
----- | -----

```

```

[1,] 0.07907139 0.02731668 [2,] 0.02731668 0.10835487

```

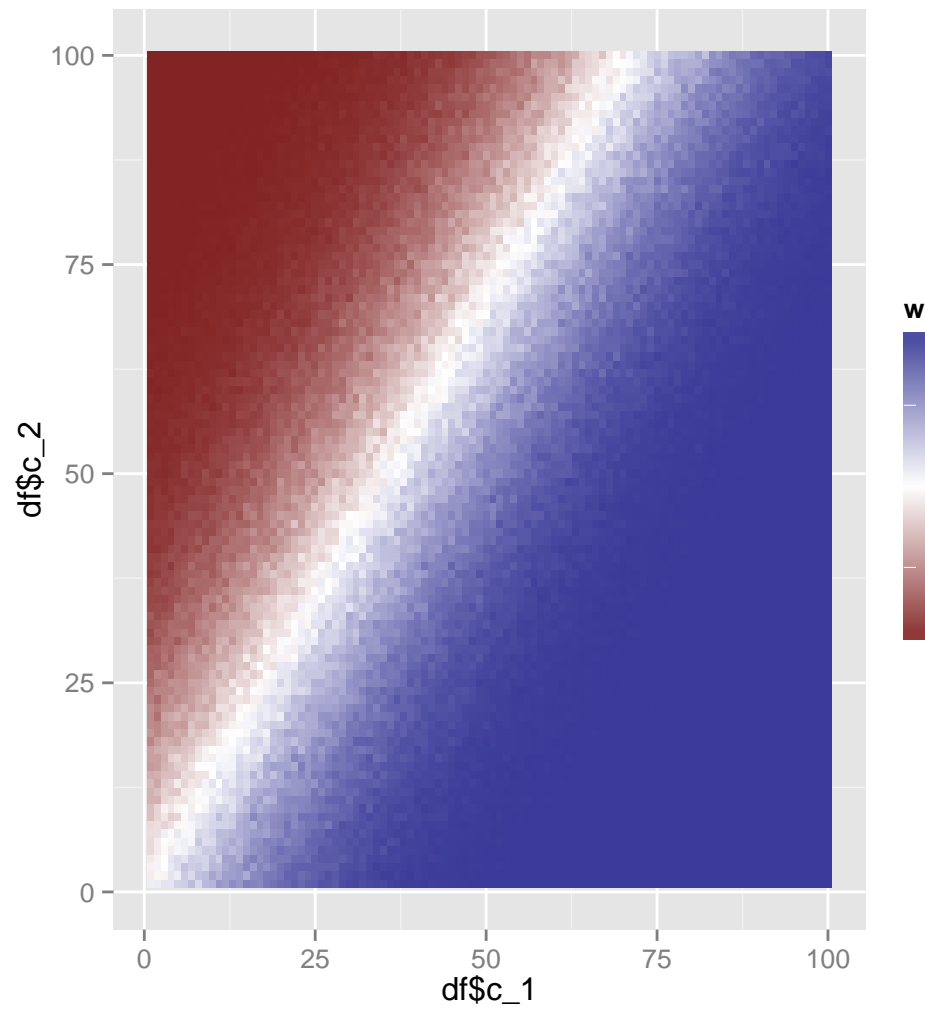
Weights for model 2

```

s = 10
erf <- function(x, mu, s) 2 * pnorm(x * sqrt(2), mean=mu, sd=s) - 1
ma_halpern2 <- function(x_1, x_2, s)
{
  integrand <- function(c)
  {
    d <- dnorm(x_1, mean=c, sd=s) * erf(c, x_2, s)
    d
  }
  i <- sum(integrand(seq(-200, 200, .01)))/100
  w <- .5 + .5 * i
  w
}
vmh <- Vectorize(ma_halpern2, c("x_1", "x_2"))
estimate_weights2 <- function(c_1, c_2, s)
{
  x_1s = rnorm(100, mean=c_1, sd=s)
  x_2s = rnorm(100, mean=c_2, sd=s)
  w = vmh(x_1s, x_2s, s)
  mean(w)
}
vew2 <- Vectorize(estimate_weights2, c("c_1", "c_2"))

df <- expand.grid(c_1=c_1, c_2=c_2)
df$w <- vew2(df$c_1, df$c_2, s)

```



Normalization Project/MH Model 2 plot.pdf