

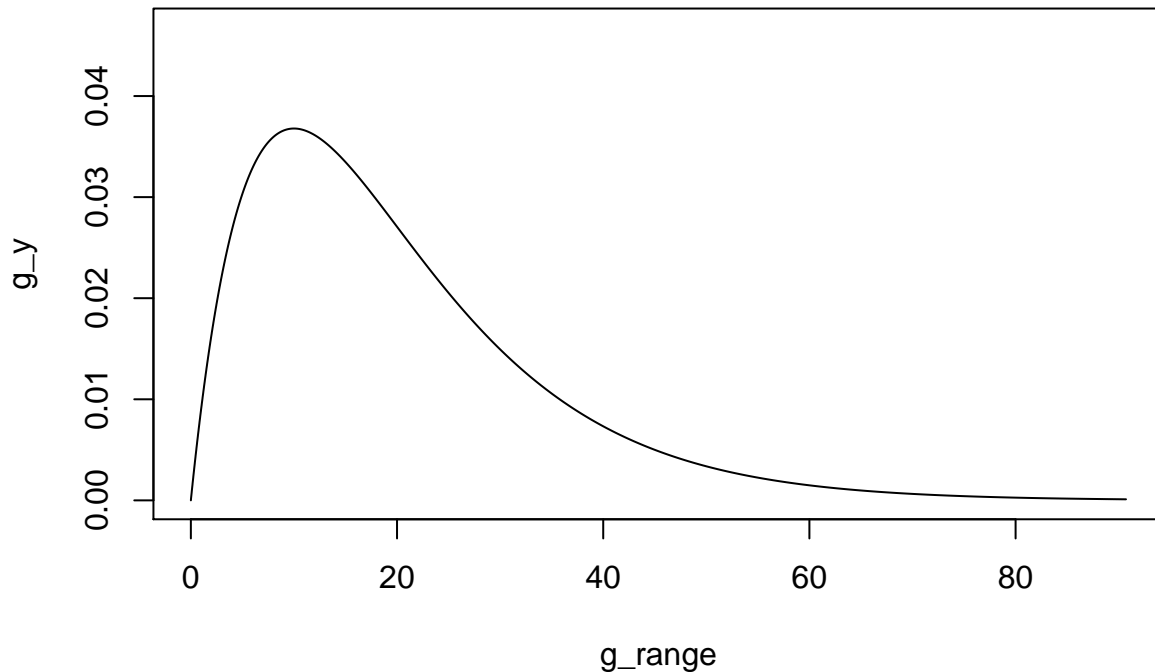
Version 1: U,V re-parameterization

Simulate data

```
I <- 50
K <- 2
S <- 10

# choose diffuse priors for gamma
a_gamma <- 2
b_gamma <- 10

avrg <- a_gamma * b_gamma
std.dv <- sqrt(a_gamma*b_gamma^2)
g_range = seq(0, avrg + 5*std.dv, 0.01)
g_y = dgamma(g_range, a_gamma, rate = 1/b_gamma)
plot(g_range, g_y, type = "l", ylim=c(0, max(g_y) + 0.01))
```



```
set.seed(123)

a <- matrix(NA, nrow=K, ncol=S)
b <- matrix(NA, nrow=K, ncol=S)
for (s in 1:S) {
  a[, s] <- rgamma(K, a_gamma, rate = 1/b_gamma)
  b[, s] <- rgamma(K, a_gamma, rate = 1/b_gamma)
}

pi <- as.vector(rdirichlet(1, rep(1, K)))
z <- sample(1:K, size = I, replace = T, prob = pi)
```

```

w <- matrix(NA, nrow=I, ncol=S)
for (s in 1:S) {
  w[, s] <- rbeta(I, a[,s][z], b[,s][z])
}

tcn <- matrix(2, nrow=I, ncol=S)
m <- matrix(rep(sample(1:2, size = I, replace = T), S), nrow=I, ncol=S)

calcTheta <- function(m, tcn, w) {
  (m * w) / (tcn * w + 2*(1-w))
}
theta <- calcTheta(m, tcn, w)

n <- replicate(S, rpois(I, 100))
y <- matrix(NA, nrow=I, ncol=S)
for (i in 1:I) {
  for (s in 1:S) {
    y[i, s] <- rbinom(1, n[i, s], theta[i,s])
  }
}

```

JAGS

```

jags.file <- file.path(working.dir, "model_uv.jags")

test.data <- list("I" = I, "S" = S, "K" = K,
                 "y" = y, "n" = n,
                 "m" = m, "tcn" = tcn)
jags.m <- jags.model(jags.file, test.data,
                    n.chains = 1,
                    n.adapt = 500)

```

```

## Compiling model graph
##   Resolving undeclared variables
##   Allocating nodes
## Graph information:
##   Observed stochastic nodes: 500
##   Unobserved stochastic nodes: 591
##   Total graph size: 6449
##
## Initializing model

```

```

params <- c("z", "w", "a", "b", "U", "V")
samps <- coda.samples(jags.m, params, n.iter=10000)
s <- summary(window(samps, start=5001))
effectiveSize(window(samps, start=5001))

```

```

##   U[1,1]   U[2,1]   U[1,2]   U[2,2]   U[1,3]   U[2,3]   U[1,4]
## 1737.3064 1251.6122  932.0783  266.4682 1601.7533 1057.8982 1734.5797
##   U[2,4]   U[1,5]   U[2,5]   U[1,6]   U[2,6]   U[1,7]   U[2,7]
## 1163.4359 1768.4118 1526.9181 2423.0453 1712.2453 1510.5040 1426.7067
##   U[1,8]   U[2,8]   U[1,9]   U[2,9]   U[1,10]  U[2,10]  V[1,1]
## 1373.6653 1428.6309 1667.6863 1014.9223 1914.3396 1947.0435 1014.0316

```

##	V[2,1]	V[1,2]	V[2,2]	V[1,3]	V[2,3]	V[1,4]	V[2,4]
##	580.5630	836.3726	303.2866	968.3463	832.2423	1144.9788	706.3413
##	V[1,5]	V[2,5]	V[1,6]	V[2,6]	V[1,7]	V[2,7]	V[1,8]
##	723.1017	844.3335	1503.4237	1070.7468	931.6339	816.6143	904.1866
##	V[2,8]	V[1,9]	V[2,9]	V[1,10]	V[2,10]	a[1,1]	a[2,1]
##	651.3592	996.6886	721.9430	1034.4395	1173.2720	1015.3632	583.9302
##	a[1,2]	a[2,2]	a[1,3]	a[2,3]	a[1,4]	a[2,4]	a[1,5]
##	831.2953	303.5696	956.3243	834.1906	1287.0225	699.2611	722.2055
##	a[2,5]	a[1,6]	a[2,6]	a[1,7]	a[2,7]	a[1,8]	a[2,8]
##	846.6685	1747.5432	1096.9097	970.6766	817.2995	901.5781	632.5009
##	a[1,9]	a[2,9]	a[1,10]	a[2,10]	b[1,1]	b[2,1]	b[1,2]
##	1019.0326	716.1716	1053.8042	1209.6625	1111.2920	603.7432	863.8119
##	b[2,2]	b[1,3]	b[2,3]	b[1,4]	b[2,4]	b[1,5]	b[2,5]
##	332.6447	1007.2336	833.7315	1100.1639	744.5732	753.2844	854.0967
##	b[1,6]	b[2,6]	b[1,7]	b[2,7]	b[1,8]	b[2,8]	b[1,9]
##	1485.1777	1234.1567	924.6156	822.0127	914.9369	665.0416	996.5853
##	b[2,9]	b[1,10]	b[2,10]	w[1,1]	w[2,1]	w[3,1]	w[4,1]
##	725.4226	1055.3392	1167.8177	1940.0485	2037.5622	2907.6891	3407.0231
##	w[5,1]	w[6,1]	w[7,1]	w[8,1]	w[9,1]	w[10,1]	w[11,1]
##	1948.5355	2863.8599	2874.4170	3038.3071	2837.2631	3166.7102	2627.9429
##	w[12,1]	w[13,1]	w[14,1]	w[15,1]	w[16,1]	w[17,1]	w[18,1]
##	2844.1986	2715.3725	3538.0900	2414.5351	2255.5152	2797.2099	2091.6386
##	w[19,1]	w[20,1]	w[21,1]	w[22,1]	w[23,1]	w[24,1]	w[25,1]
##	2974.4963	3311.8938	3167.3826	3276.4820	2297.5317	2300.2958	3091.6605
##	w[26,1]	w[27,1]	w[28,1]	w[29,1]	w[30,1]	w[31,1]	w[32,1]
##	3347.5316	2961.7986	3409.4187	3443.4038	3135.3001	3362.5569	2780.7840
##	w[33,1]	w[34,1]	w[35,1]	w[36,1]	w[37,1]	w[38,1]	w[39,1]
##	2646.1170	1559.5643	2907.2646	2476.3488	2485.9244	3135.3615	2870.7971
##	w[40,1]	w[41,1]	w[42,1]	w[43,1]	w[44,1]	w[45,1]	w[46,1]
##	3344.3302	3159.2099	2002.5072	2418.3199	2751.9480	3189.5839	3201.3192
##	w[47,1]	w[48,1]	w[49,1]	w[50,1]	w[1,2]	w[2,2]	w[3,2]
##	3398.8459	3388.2906	1888.2178	2774.9306	1055.4626	2151.2825	752.9113
##	w[4,2]	w[5,2]	w[6,2]	w[7,2]	w[8,2]	w[9,2]	w[10,2]
##	2187.6690	1979.3487	2246.6976	976.2168	1548.7709	2226.8899	582.8350
##	w[11,2]	w[12,2]	w[13,2]	w[14,2]	w[15,2]	w[16,2]	w[17,2]
##	912.4261	2546.9388	877.9378	2165.6026	919.6659	533.8978	1817.7767
##	w[18,2]	w[19,2]	w[20,2]	w[21,2]	w[22,2]	w[23,2]	w[24,2]
##	821.4947	865.9877	965.6017	2806.0583	2799.2268	1991.1822	2439.2357
##	w[25,2]	w[26,2]	w[27,2]	w[28,2]	w[29,2]	w[30,2]	w[31,2]
##	2379.7591	725.6279	1963.6392	2124.9014	2830.6751	2731.3718	1864.9999
##	w[32,2]	w[33,2]	w[34,2]	w[35,2]	w[36,2]	w[37,2]	w[38,2]
##	664.3795	1049.5068	967.6334	947.2942	2529.6916	2561.5996	1118.1925
##	w[39,2]	w[40,2]	w[41,2]	w[42,2]	w[43,2]	w[44,2]	w[45,2]
##	634.1383	2833.7691	2236.3717	1878.3449	698.1493	1149.0540	879.5091
##	w[46,2]	w[47,2]	w[48,2]	w[49,2]	w[50,2]	w[1,3]	w[2,3]
##	772.1885	997.4066	943.2370	2876.4995	883.4947	3102.9351	2901.1946
##	w[3,3]	w[4,3]	w[5,3]	w[6,3]	w[7,3]	w[8,3]	w[9,3]
##	3204.2714	2479.9700	1811.5199	2576.3707	2880.3830	3046.7187	2558.8366
##	w[10,3]	w[11,3]	w[12,3]	w[13,3]	w[14,3]	w[15,3]	w[16,3]
##	3110.1778	2863.1738	3137.3433	2625.9944	3282.8462	3031.8401	3088.0340
##	w[17,3]	w[18,3]	w[19,3]	w[20,3]	w[21,3]	w[22,3]	w[23,3]
##	2259.9592	3292.8039	3170.4516	2798.2575	3052.7297	3229.3997	2753.8593
##	w[24,3]	w[25,3]	w[26,3]	w[27,3]	w[28,3]	w[29,3]	w[30,3]
##	2719.3620	2692.4770	2968.0778	2496.7642	2998.7702	3264.4814	3000.0434

##	w[31,3]	w[32,3]	w[33,3]	w[34,3]	w[35,3]	w[36,3]	w[37,3]
##	2663.3458	2831.3546	2891.9082	3288.6401	2768.3372	3383.2327	2992.5501
##	w[38,3]	w[39,3]	w[40,3]	w[41,3]	w[42,3]	w[43,3]	w[44,3]
##	2684.0616	2790.1404	2627.8137	2526.5224	2859.9099	2995.8191	2574.5383
##	w[45,3]	w[46,3]	w[47,3]	w[48,3]	w[49,3]	w[50,3]	w[1,4]
##	3177.7305	3162.4392	2719.6952	3009.4439	2953.2007	2899.4058	2971.6904
##	w[2,4]	w[3,4]	w[4,4]	w[5,4]	w[6,4]	w[7,4]	w[8,4]
##	2816.3901	2267.8731	3144.2714	2615.2692	3212.3475	2620.0420	2543.7986
##	w[9,4]	w[10,4]	w[11,4]	w[12,4]	w[13,4]	w[14,4]	w[15,4]
##	2976.0935	3354.5482	2973.6494	2972.4257	2442.3556	3034.6681	3196.0348
##	w[16,4]	w[17,4]	w[18,4]	w[19,4]	w[20,4]	w[21,4]	w[22,4]
##	2913.4565	2679.1650	3051.1874	3370.7351	2228.0159	3146.0168	3168.9961
##	w[23,4]	w[24,4]	w[25,4]	w[26,4]	w[27,4]	w[28,4]	w[29,4]
##	2412.9609	3804.8901	2645.4741	3013.8119	3199.0126	2653.2964	3206.7777
##	w[30,4]	w[31,4]	w[32,4]	w[33,4]	w[34,4]	w[35,4]	w[36,4]
##	2910.4511	2748.7165	3512.2539	3511.4259	3001.0960	3049.3465	2789.7624
##	w[37,4]	w[38,4]	w[39,4]	w[40,4]	w[41,4]	w[42,4]	w[43,4]
##	2126.0170	3038.0784	3114.0189	2735.3402	3360.0777	3176.4612	2625.1373
##	w[44,4]	w[45,4]	w[46,4]	w[47,4]	w[48,4]	w[49,4]	w[50,4]
##	2437.5950	2693.1011	2682.4165	4091.3759	2803.7886	3214.0456	2215.8410
##	w[1,5]	w[2,5]	w[3,5]	w[4,5]	w[5,5]	w[6,5]	w[7,5]
##	2311.9958	2002.6018	3348.2218	2924.7451	2605.5603	3133.6464	2829.2210
##	w[8,5]	w[9,5]	w[10,5]	w[11,5]	w[12,5]	w[13,5]	w[14,5]
##	3006.3422	3084.4295	2201.3000	3150.3610	2552.9135	2595.1793	3154.1317
##	w[15,5]	w[16,5]	w[17,5]	w[18,5]	w[19,5]	w[20,5]	w[21,5]
##	3532.7761	3009.6889	2274.8494	3140.1867	2945.5196	3138.4740	3288.5362
##	w[22,5]	w[23,5]	w[24,5]	w[25,5]	w[26,5]	w[27,5]	w[28,5]
##	3452.7239	1719.6097	3490.9978	2957.7434	3200.2196	3229.6771	2892.6288
##	w[29,5]	w[30,5]	w[31,5]	w[32,5]	w[33,5]	w[34,5]	w[35,5]
##	3000.3504	3143.4099	3533.4272	2233.9692	2826.2993	2505.9331	3644.9372
##	w[36,5]	w[37,5]	w[38,5]	w[39,5]	w[40,5]	w[41,5]	w[42,5]
##	3339.1202	3351.4623	3174.2198	3049.1459	2428.9173	2569.0023	2948.7650
##	w[43,5]	w[44,5]	w[45,5]	w[46,5]	w[47,5]	w[48,5]	w[49,5]
##	3236.1391	3021.4424	2482.5974	2596.6958	2336.6983	2500.0416	3192.7562
##	w[50,5]	w[1,6]	w[2,6]	w[3,6]	w[4,6]	w[5,6]	w[6,6]
##	3011.9181	2278.3093	2611.5379	2410.7482	3310.6563	3454.7527	3143.5007
##	w[7,6]	w[8,6]	w[9,6]	w[10,6]	w[11,6]	w[12,6]	w[13,6]
##	3133.4480	3461.8384	2838.7098	3037.6914	3175.8240	3493.1901	3095.1298
##	w[14,6]	w[15,6]	w[16,6]	w[17,6]	w[18,6]	w[19,6]	w[20,6]
##	3552.8972	2527.5667	2891.8662	3184.2762	3112.3069	3077.3584	2734.8128
##	w[21,6]	w[22,6]	w[23,6]	w[24,6]	w[25,6]	w[26,6]	w[27,6]
##	3418.6753	3341.4959	3338.5886	2967.1103	3471.1008	2954.8390	1207.6273
##	w[28,6]	w[29,6]	w[30,6]	w[31,6]	w[32,6]	w[33,6]	w[34,6]
##	3564.4299	3575.3133	3711.8315	3423.9307	3024.7522	3225.5522	3318.8277
##	w[35,6]	w[36,6]	w[37,6]	w[38,6]	w[39,6]	w[40,6]	w[41,6]
##	2987.1114	1742.7344	3216.1829	2830.7155	3034.4225	3697.2463	3467.5918
##	w[42,6]	w[43,6]	w[44,6]	w[45,6]	w[46,6]	w[47,6]	w[48,6]
##	2979.0979	2963.8757	3336.7874	2408.7214	2960.5450	3195.6152	3195.3521
##	w[49,6]	w[50,6]	w[1,7]	w[2,7]	w[3,7]	w[4,7]	w[5,7]
##	2927.9550	2711.8292	3075.8315	2767.7464	2063.8533	3048.4041	3030.6075
##	w[6,7]	w[7,7]	w[8,7]	w[9,7]	w[10,7]	w[11,7]	w[12,7]
##	2594.8615	3415.0630	3001.1866	2553.8795	3218.3624	2716.4298	2837.5035
##	w[13,7]	w[14,7]	w[15,7]	w[16,7]	w[17,7]	w[18,7]	w[19,7]
##	2977.6400	3285.6899	2039.0275	3126.9193	2757.5923	3161.7188	2969.9458

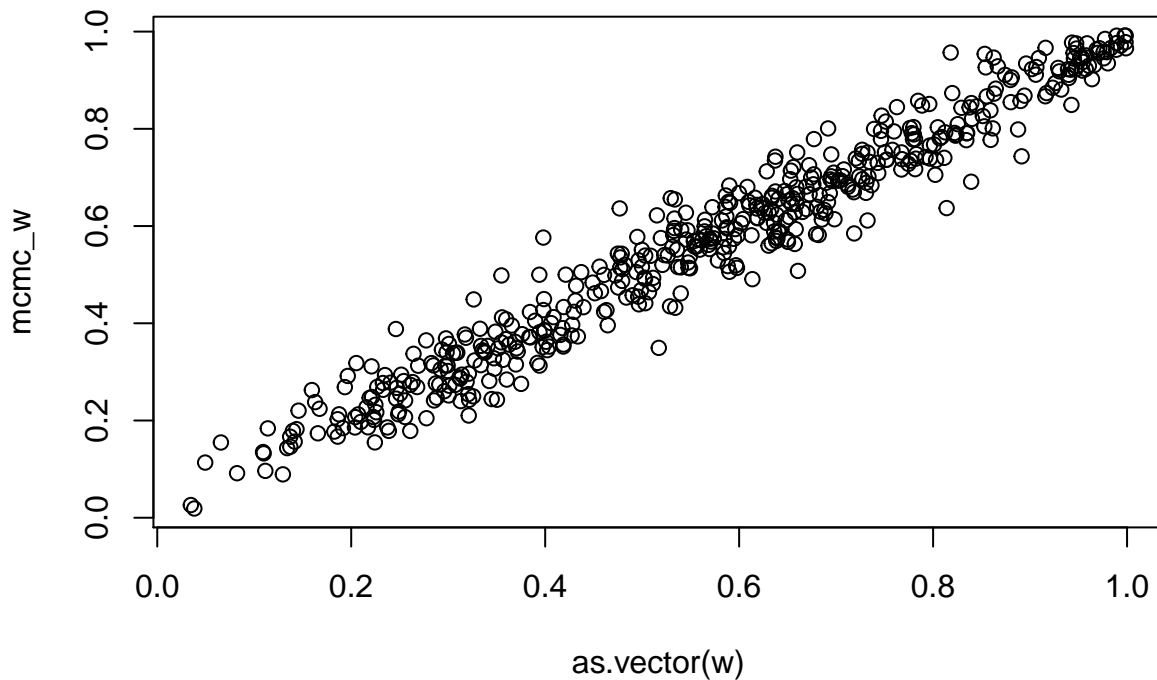
##	w[20,7]	w[21,7]	w[22,7]	w[23,7]	w[24,7]	w[25,7]	w[26,7]
##	1713.1853	2887.9597	3264.4819	2841.6764	2985.1207	3143.8876	2753.8231
##	w[27,7]	w[28,7]	w[29,7]	w[30,7]	w[31,7]	w[32,7]	w[33,7]
##	2616.3430	2348.7239	2631.6914	3269.4712	3087.3752	3164.3332	2689.7977
##	w[34,7]	w[35,7]	w[36,7]	w[37,7]	w[38,7]	w[39,7]	w[40,7]
##	2667.0139	2831.0358	3165.3222	3548.3893	2688.2092	3107.0266	3384.5967
##	w[41,7]	w[42,7]	w[43,7]	w[44,7]	w[45,7]	w[46,7]	w[47,7]
##	3052.9576	2546.3136	2996.8276	2973.0313	3088.3268	2494.3127	3224.3705
##	w[48,7]	w[49,7]	w[50,7]	w[1,8]	w[2,8]	w[3,8]	w[4,8]
##	2849.5899	3299.4658	2927.5265	3019.7666	2823.2283	3063.9867	2745.3818
##	w[5,8]	w[6,8]	w[7,8]	w[8,8]	w[9,8]	w[10,8]	w[11,8]
##	1571.4232	3364.7798	3007.0306	2114.1389	2809.1366	3388.9005	2283.7503
##	w[12,8]	w[13,8]	w[14,8]	w[15,8]	w[16,8]	w[17,8]	w[18,8]
##	2748.2543	2785.9851	3167.0110	2798.7269	3075.8522	2461.6394	2882.7452
##	w[19,8]	w[20,8]	w[21,8]	w[22,8]	w[23,8]	w[24,8]	w[25,8]
##	3239.4079	2924.7618	2790.6009	2860.4248	2464.3811	3025.3918	3040.5871
##	w[26,8]	w[27,8]	w[28,8]	w[29,8]	w[30,8]	w[31,8]	w[32,8]
##	2699.2805	2859.1923	2281.5765	3142.8271	2918.5762	3440.1476	2251.2958
##	w[33,8]	w[34,8]	w[35,8]	w[36,8]	w[37,8]	w[38,8]	w[39,8]
##	2511.5058	2653.2785	2994.3113	2891.4977	3278.9916	1481.6929	2440.4912
##	w[40,8]	w[41,8]	w[42,8]	w[43,8]	w[44,8]	w[45,8]	w[46,8]
##	3254.6108	2806.3156	2612.9394	3368.1034	3149.1805	2612.3459	2486.3725
##	w[47,8]	w[48,8]	w[49,8]	w[50,8]	w[1,9]	w[2,9]	w[3,9]
##	2435.3423	1777.8014	3277.8158	2737.4743	2285.3672	3147.7183	2910.3875
##	w[4,9]	w[5,9]	w[6,9]	w[7,9]	w[8,9]	w[9,9]	w[10,9]
##	2440.9104	3130.9617	3145.5767	2613.6041	3303.9774	3164.5290	2728.2237
##	w[11,9]	w[12,9]	w[13,9]	w[14,9]	w[15,9]	w[16,9]	w[17,9]
##	2637.7374	2433.8801	2987.7402	2653.4551	2941.6615	2602.7072	2763.1700
##	w[18,9]	w[19,9]	w[20,9]	w[21,9]	w[22,9]	w[23,9]	w[24,9]
##	2338.4208	2959.7632	3353.9738	3317.0402	3132.0576	2170.0908	2997.4875
##	w[25,9]	w[26,9]	w[27,9]	w[28,9]	w[29,9]	w[30,9]	w[31,9]
##	3079.8187	3192.1185	3176.2841	3650.3908	2987.5868	2576.4350	2640.8189
##	w[32,9]	w[33,9]	w[34,9]	w[35,9]	w[36,9]	w[37,9]	w[38,9]
##	2805.3805	2992.7552	3043.3911	2935.6364	3515.9834	3294.5173	1621.2399
##	w[39,9]	w[40,9]	w[41,9]	w[42,9]	w[43,9]	w[44,9]	w[45,9]
##	3106.8675	3255.6682	3387.9571	2586.4731	2686.1745	2531.8134	2384.7262
##	w[46,9]	w[47,9]	w[48,9]	w[49,9]	w[50,9]	w[1,10]	w[2,10]
##	3364.1531	2833.2614	2523.3419	3368.2112	2153.2922	3368.9216	2082.1899
##	w[3,10]	w[4,10]	w[5,10]	w[6,10]	w[7,10]	w[8,10]	w[9,10]
##	3657.4033	3441.4577	2684.1316	3711.0335	2912.1494	2987.4887	3139.5549
##	w[10,10]	w[11,10]	w[12,10]	w[13,10]	w[14,10]	w[15,10]	w[16,10]
##	3051.5764	3343.6240	3665.7745	3110.4403	3338.9129	2408.1489	3303.3981
##	w[17,10]	w[18,10]	w[19,10]	w[20,10]	w[21,10]	w[22,10]	w[23,10]
##	2932.1945	3323.6369	3050.4628	2994.5177	2681.1125	2952.0443	3064.0758
##	w[24,10]	w[25,10]	w[26,10]	w[27,10]	w[28,10]	w[29,10]	w[30,10]
##	3272.7789	3307.6213	3390.9672	3204.3405	3393.3999	3370.5831	2301.9504
##	w[31,10]	w[32,10]	w[33,10]	w[34,10]	w[35,10]	w[36,10]	w[37,10]
##	3136.6457	3560.7758	2731.8870	3175.6212	3583.0975	3162.6614	3057.2687
##	w[38,10]	w[39,10]	w[40,10]	w[41,10]	w[42,10]	w[43,10]	w[44,10]
##	3484.1452	3705.0467	3356.7012	3242.1693	2795.4688	3113.0144	2842.4535
##	w[45,10]	w[46,10]	w[47,10]	w[48,10]	w[49,10]	w[50,10]	z[1]
##	3371.0628	3459.9362	3265.5612	2840.0674	3435.2093	2950.0626	0.0000
##	z[2]	z[3]	z[4]	z[5]	z[6]	z[7]	z[8]
##	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

```
##      z[9]      z[10]      z[11]      z[12]      z[13]      z[14]      z[15]
##  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
##      z[16]     z[17]     z[18]     z[19]     z[20]     z[21]     z[22]
##  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
##      z[23]     z[24]     z[25]     z[26]     z[27]     z[28]     z[29]
##  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
##      z[30]     z[31]     z[32]     z[33]     z[34]     z[35]     z[36]
##  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
##      z[37]     z[38]     z[39]     z[40]     z[41]     z[42]     z[43]
##  0.0000 144.2295  0.0000  0.0000  0.0000  0.0000  0.0000
##      z[44]     z[45]     z[46]     z[47]     z[48]     z[49]     z[50]
##  0.0000  0.0000  0.0000  0.0000 150.0202  0.0000  0.0000
```

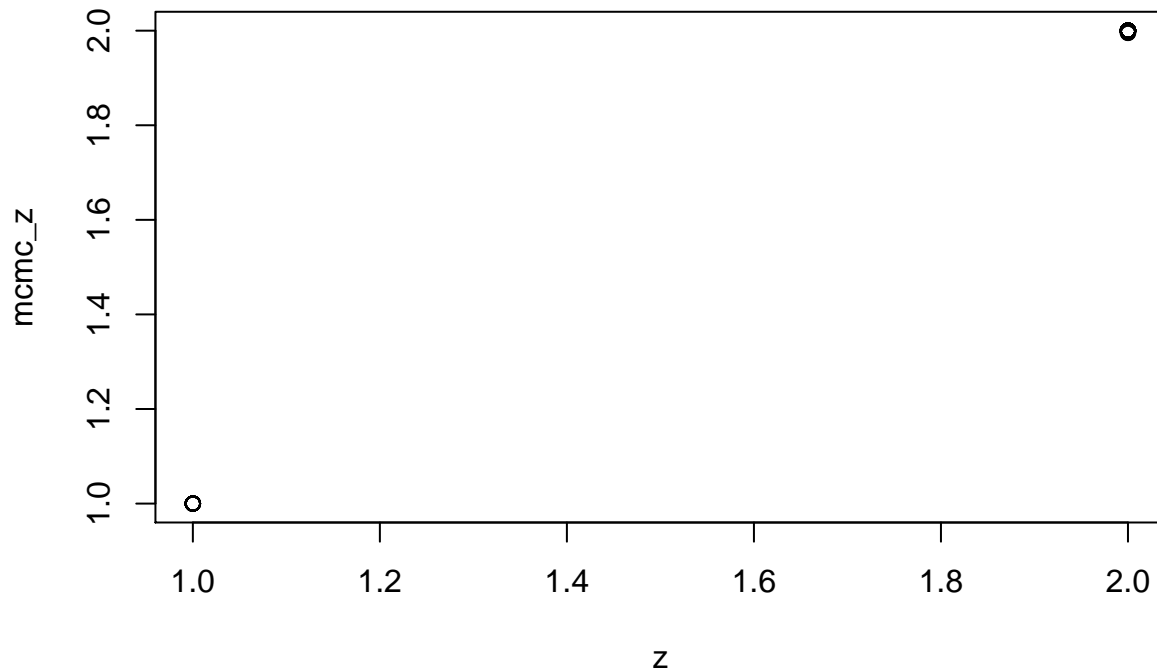
```
pdf(file.path(working.dir, "trace-plots-uv.pdf"))
plot(samps)
dev.off()
```

```
## pdf
## 2
```

```
mcmc_vals <- s$statistics
mcmc_w <- mcmc_vals[substr(rownames(mcmc_vals), 1, 1) == "w", "Mean"]
plot(as.vector(w), mcmc_w, type = "p")
```



```
mcmc_z <- as.vector(mcmc_vals[substr(rownames(mcmc_vals), 1, 1) == "z", "Mean"])
#mcmc_z <- round(mcmc_z, 0)
plot(z, mcmc_z, type = "p")
```



```

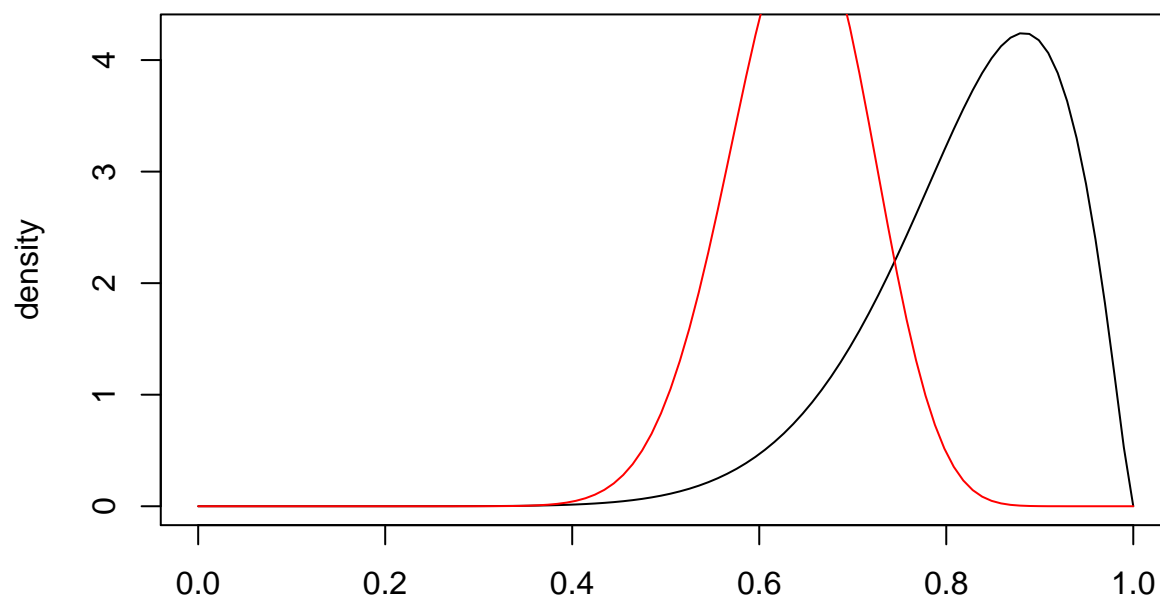
mcmc_U <- mcmc_vals[substr(rownames(mcmc_vals), 1, 1) == "U", "Mean"]
mcmc_U <- matrix(mcmc_U, nrow=K)
mcmc_V <- mcmc_vals[substr(rownames(mcmc_vals), 1, 1) == "V", "Mean"]
mcmc_V <- matrix(mcmc_V, nrow=K)

p <- seq(0, 1, length = 100)
for (s in 1:S) {
  for (k in 1:K) {
    if (k == 1) {
      plot(p, dbeta(p, mcmc_U[k,s] * mcmc_V[k,s], (1-mcmc_U[k,s])*mcmc_V[k,s]),
           main = paste0("S", s, " MCMC mean U,V"),
           ylab = "density", xlab = "w", type = "l", col = k)
    } else {
      lines(p, dbeta(p, mcmc_U[k,s] * mcmc_V[k,s], (1-mcmc_U[k,s])*mcmc_V[k,s]),
            type = "l", col = k)
    }
  }
}

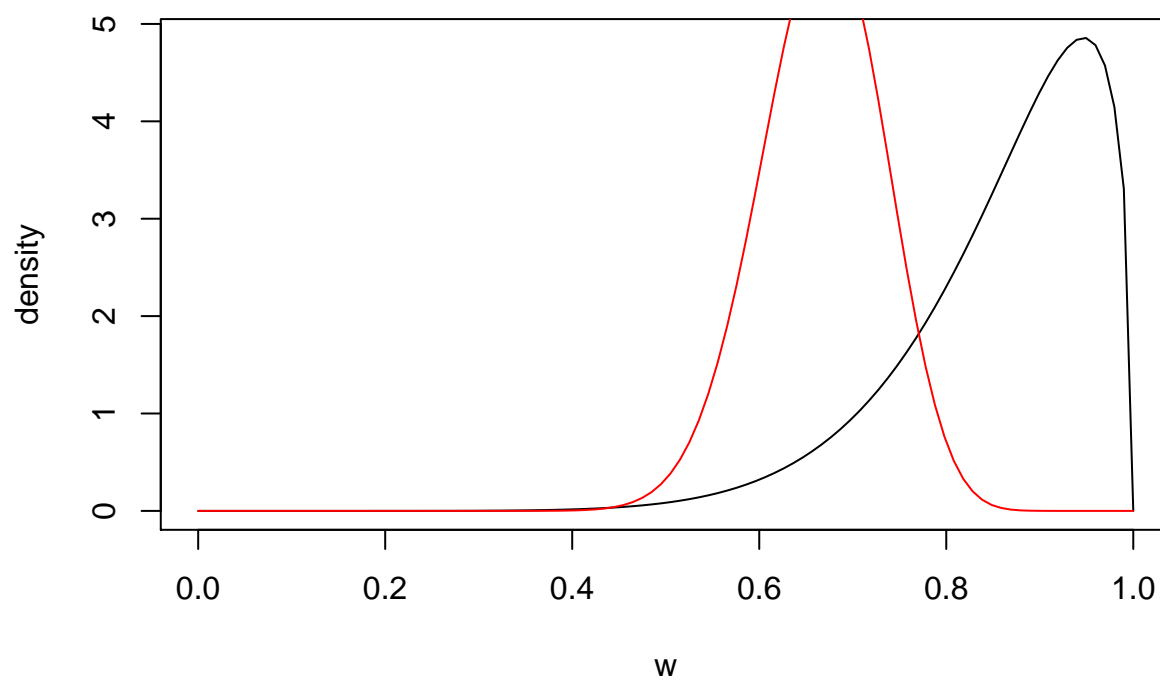
for (k in 1:K) {
  if (k == 1) {
    plot(p, dbeta(p, a[k,s], b[k,s]),
         main = paste0("S", s, " truth"),
         ylab = "density", xlab = "w", type = "l", col = k)
  } else {
    lines(p, dbeta(p, a[k,s], b[k,s]), type = "l", col = k)
  }
}
}

```

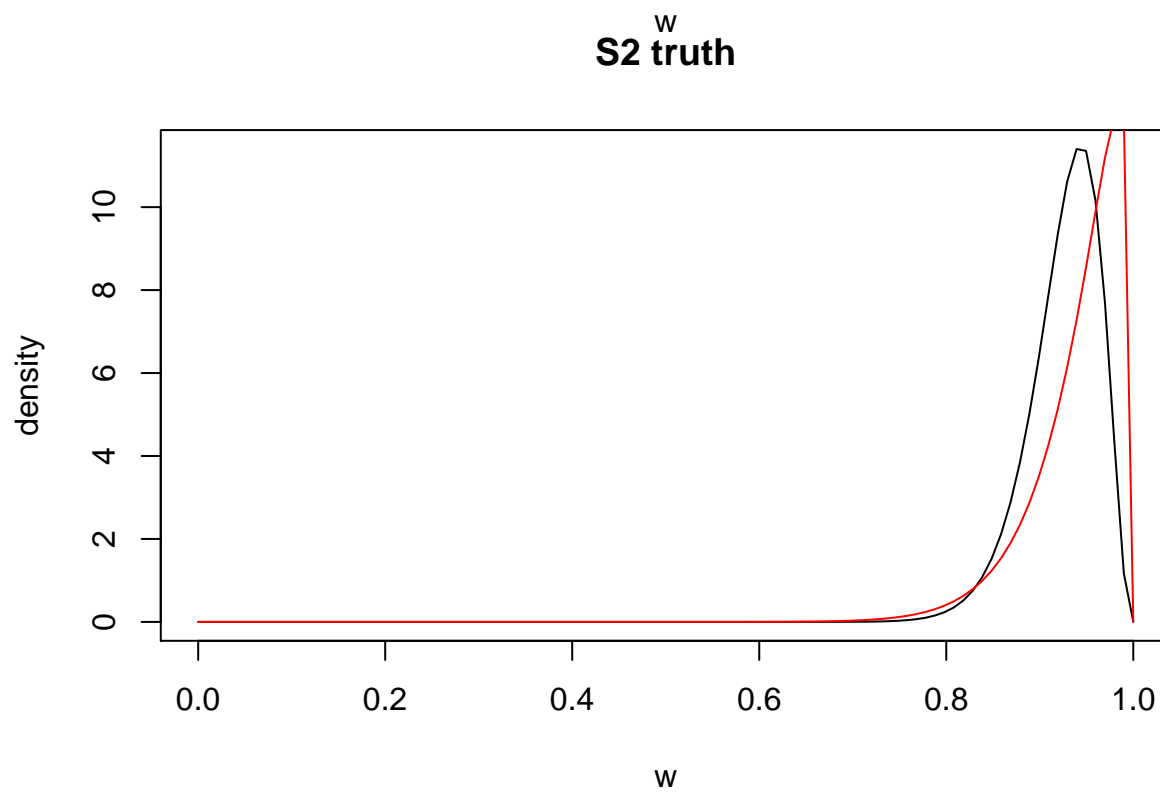
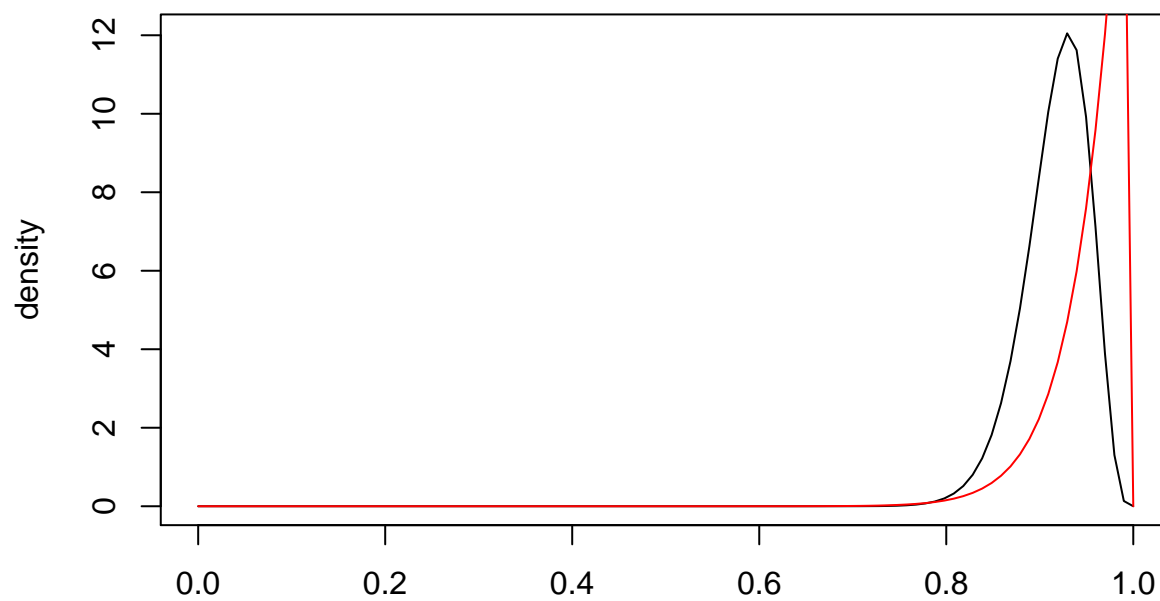
S1 MCMC mean U,V



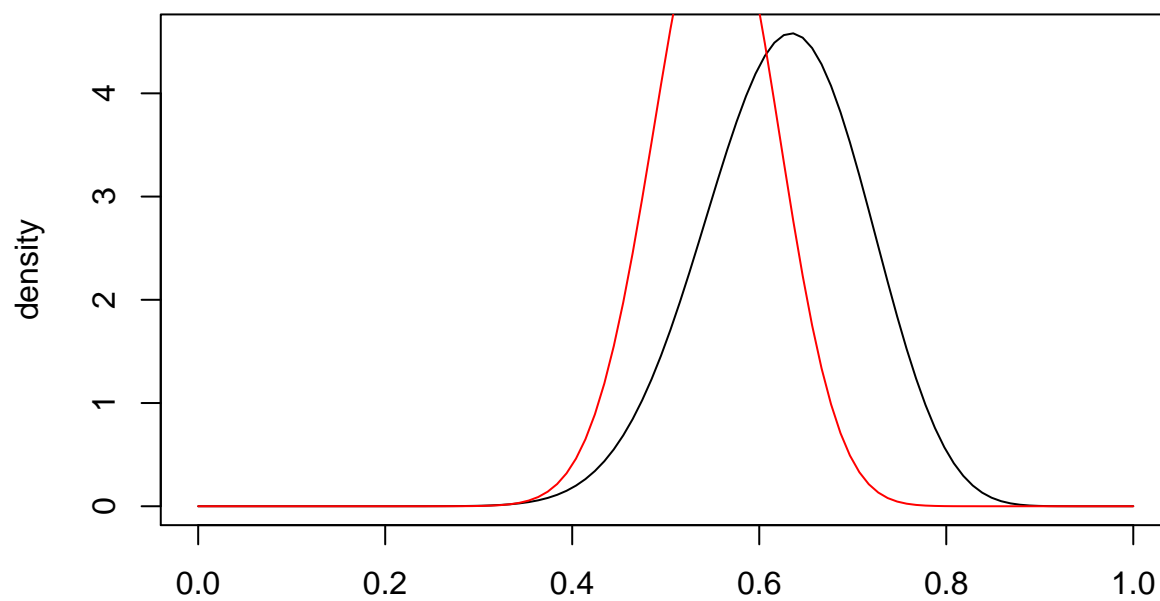
S1 truth^w



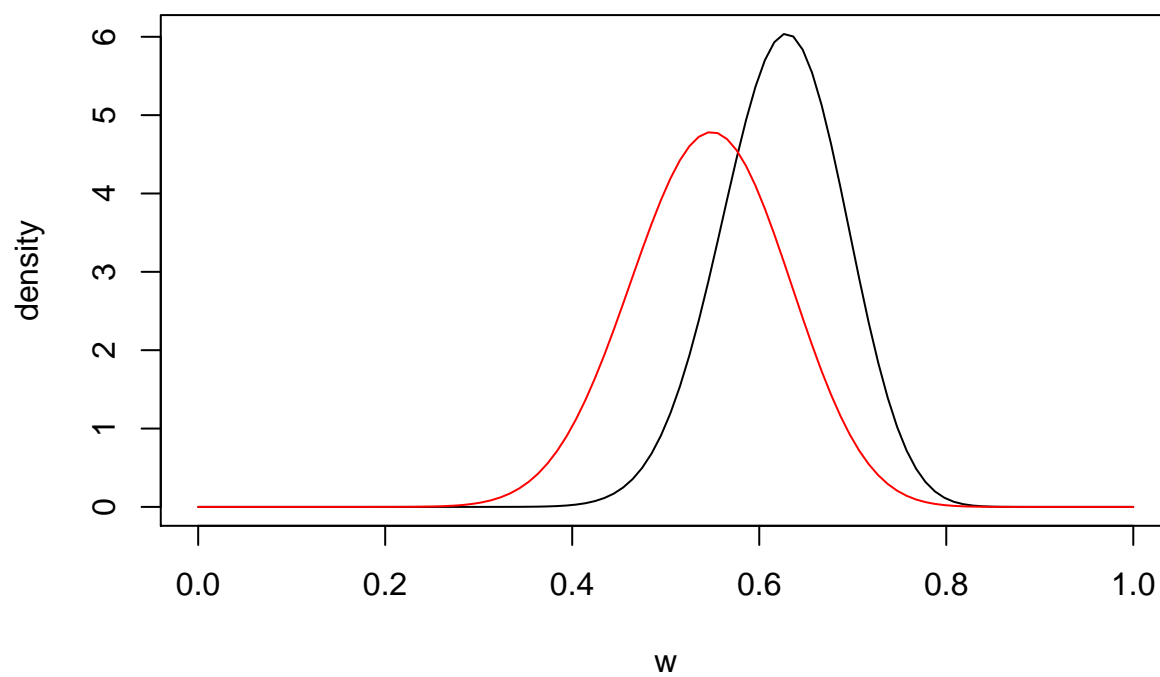
S2 MCMC mean U,V



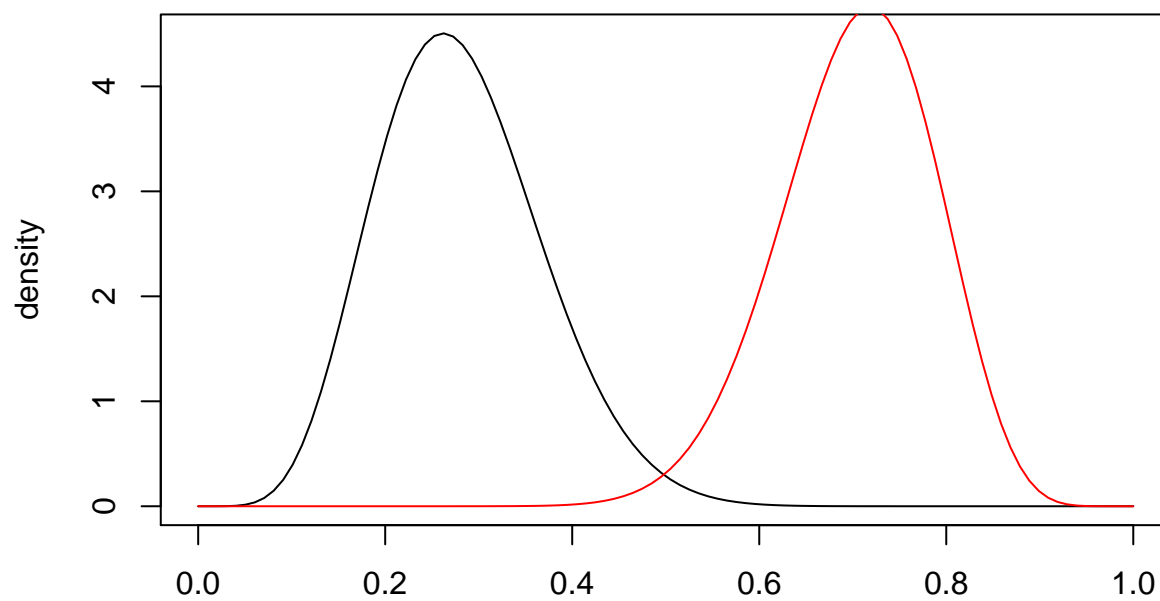
S3 MCMC mean U,V



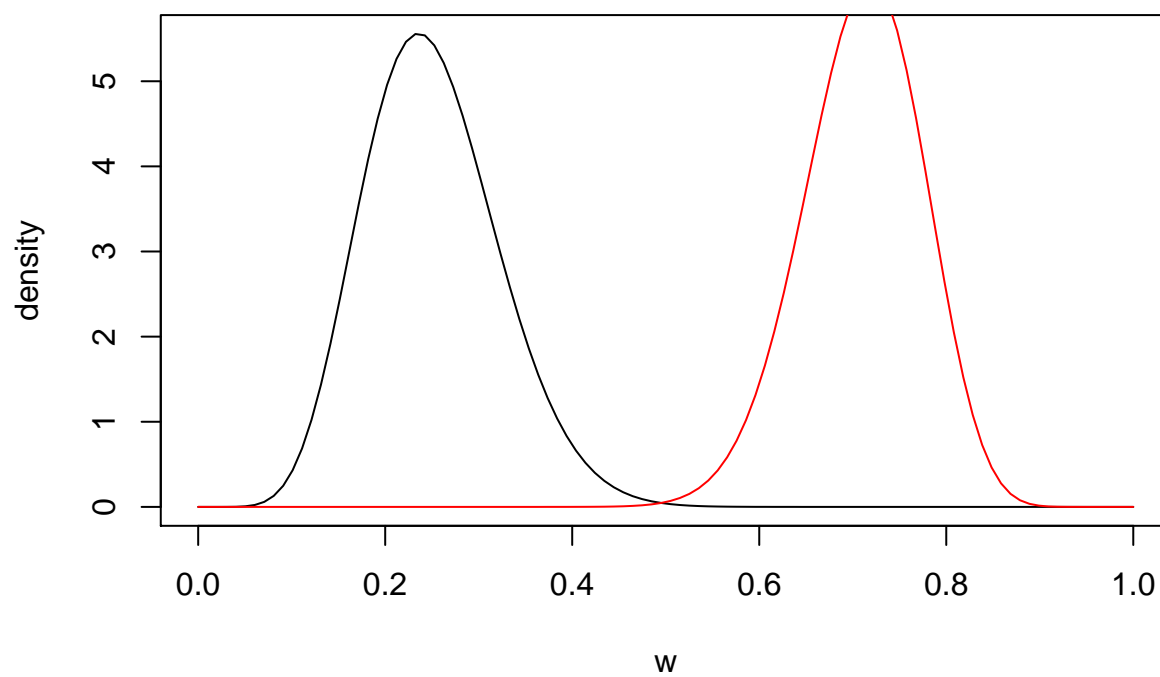
S3 truth^w



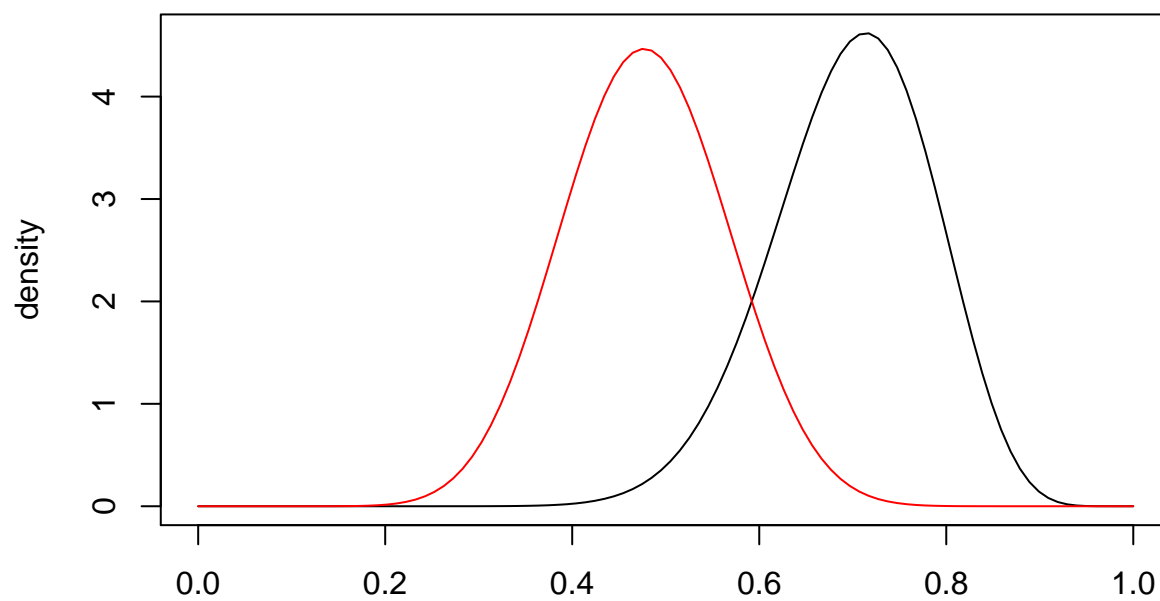
S4 MCMC mean U,V



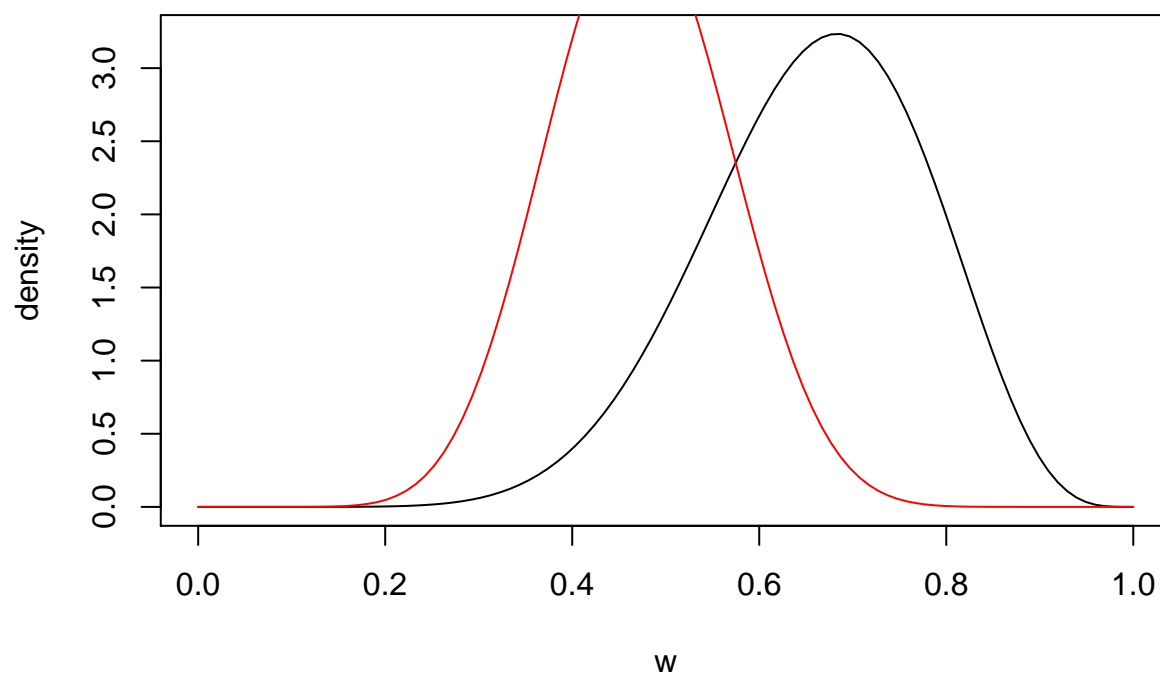
S4 truth^w



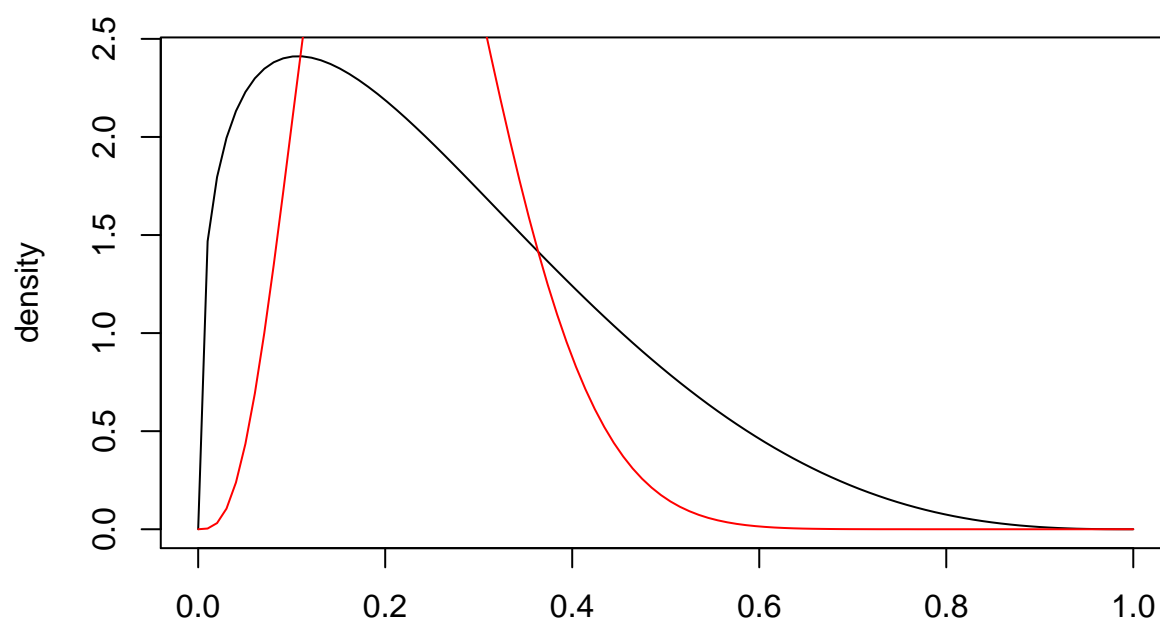
S5 MCMC mean U,V



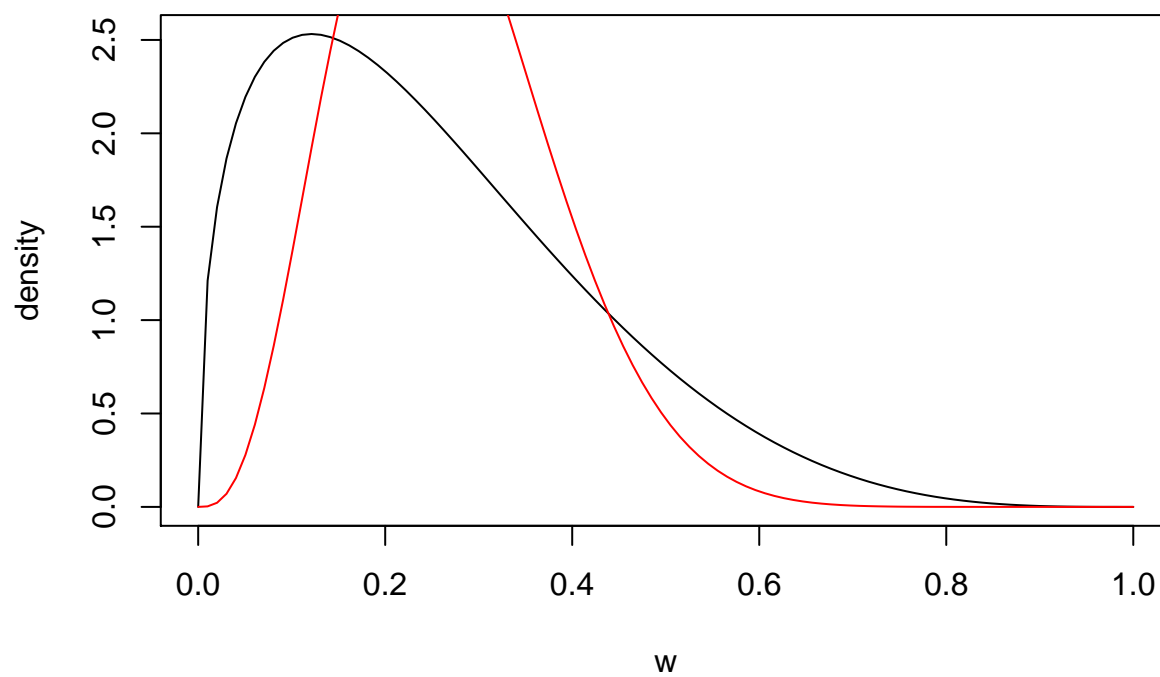
S5 truth^w



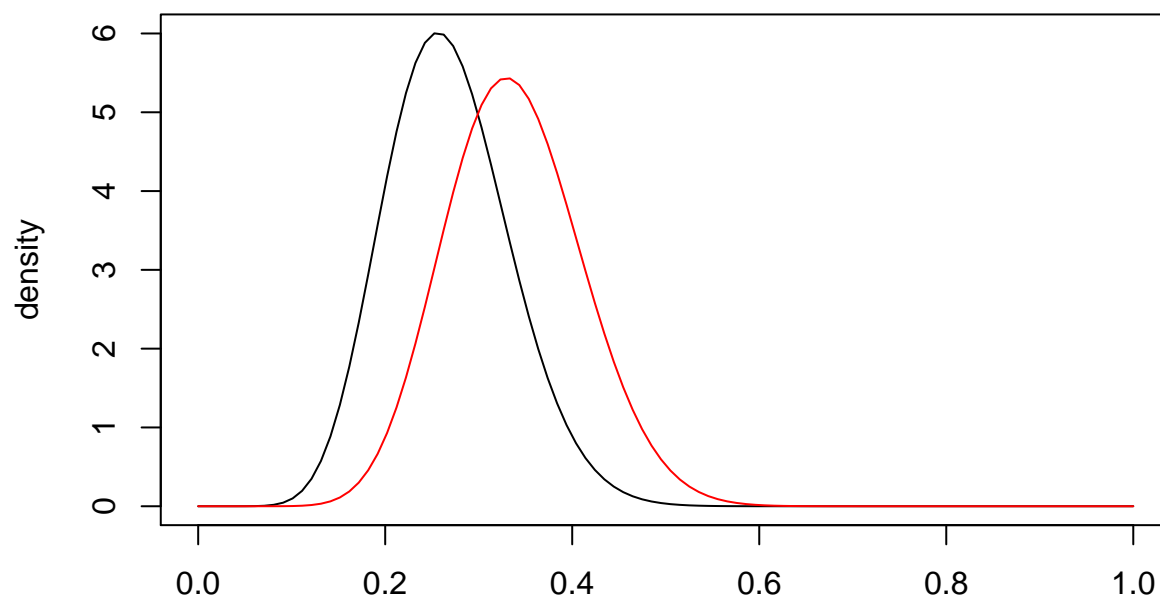
S6 MCMC mean U,V



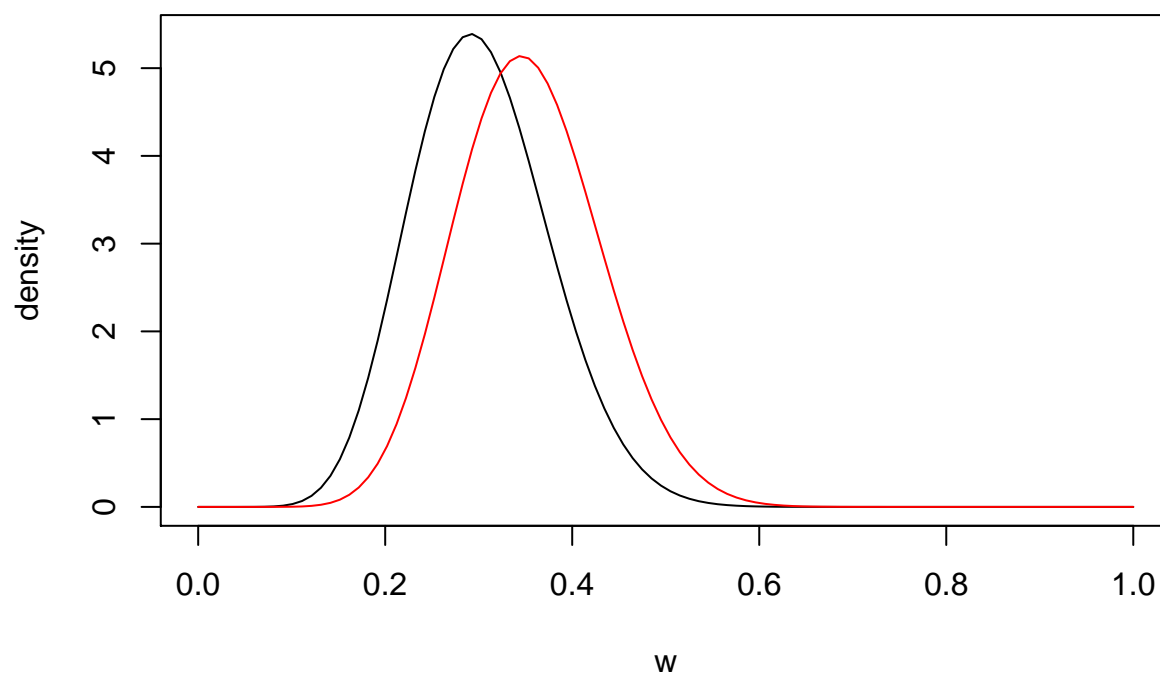
S6 truth^w



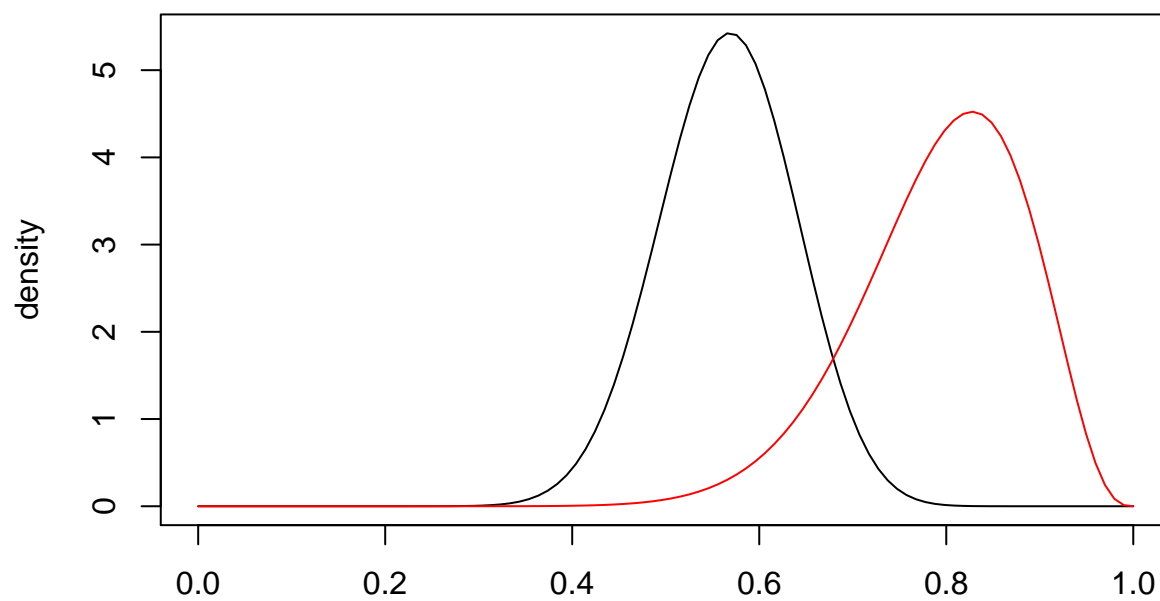
S7 MCMC mean U,V



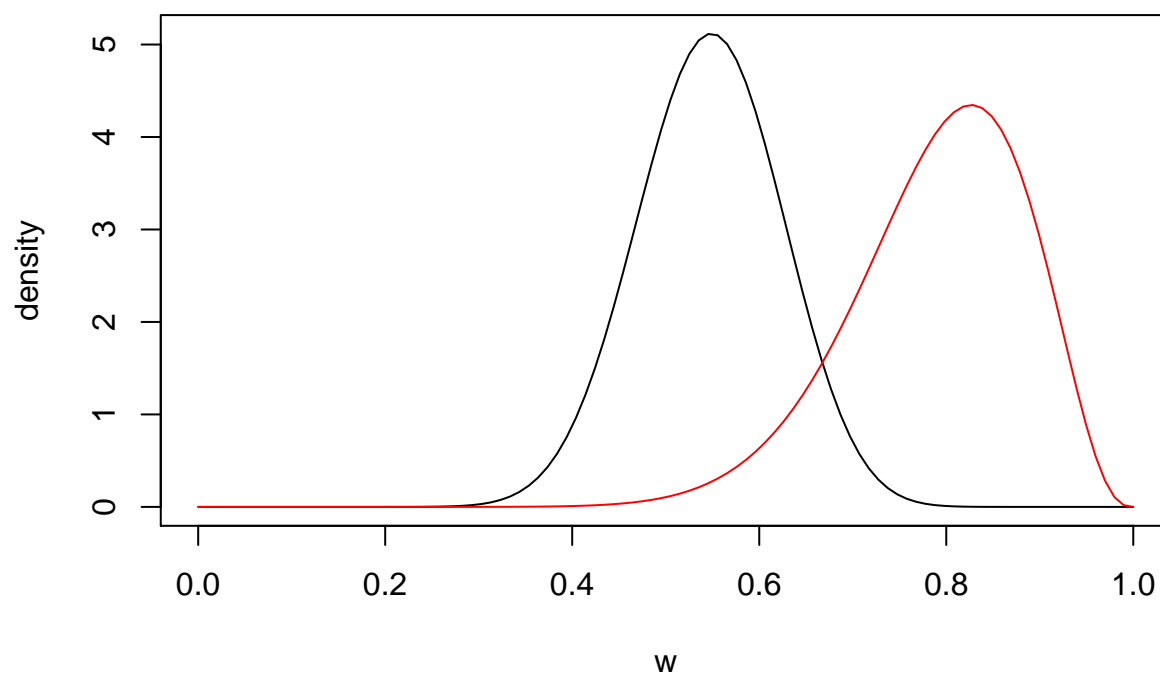
S7 truth



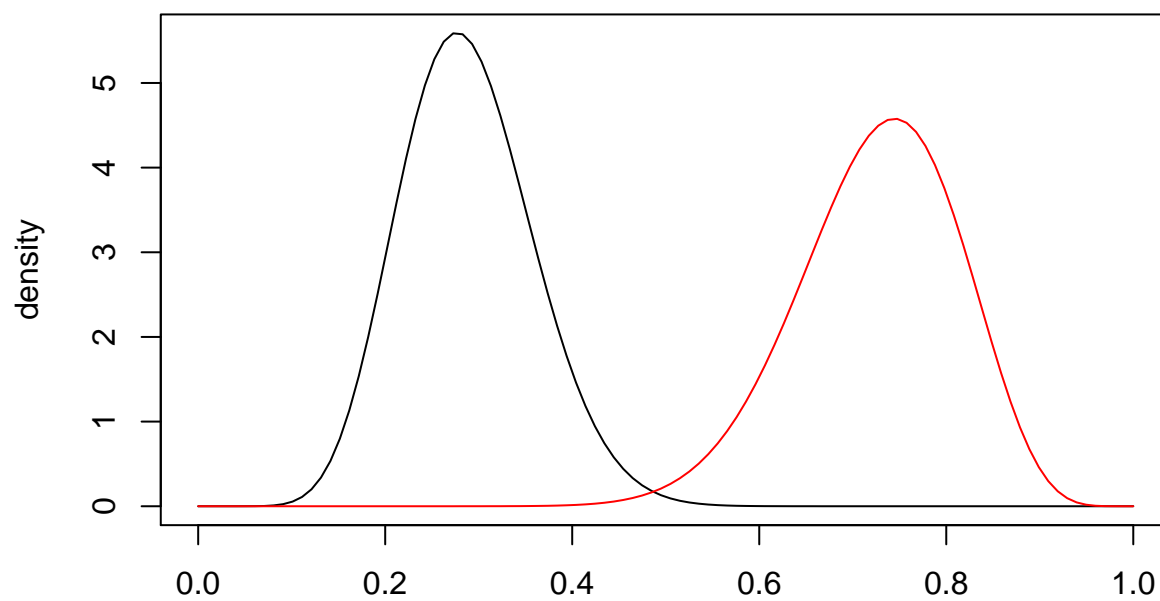
S8 MCMC mean U,V



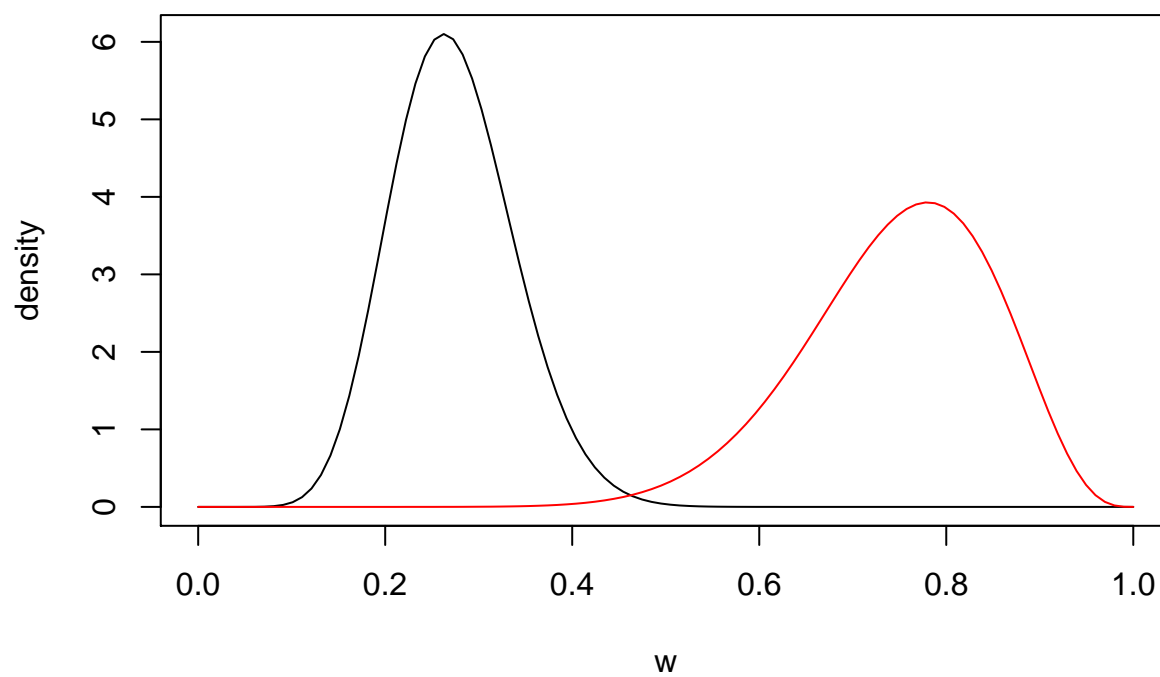
S8 truth^w



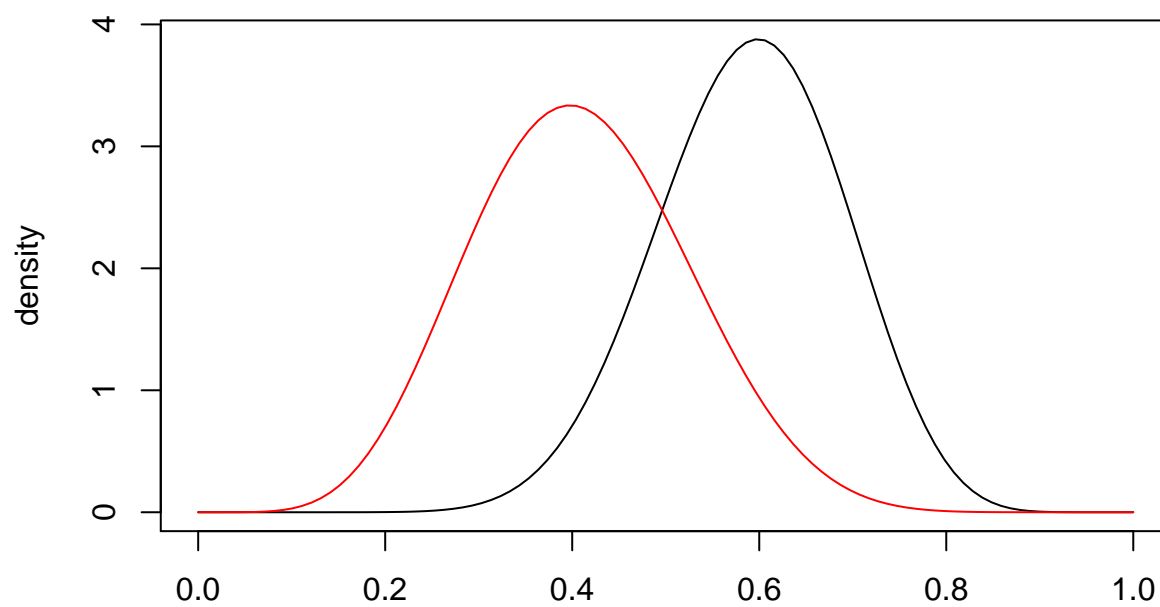
S9 MCMC mean U,V



S9 truth



S10 MCMC mean U,V



S10^w truth

