Exercise

s-takano

```r
set.seed(29)
```

```r
library(LearnBayes)
```

2-1

事前分布の設定

```r
p = seq(0, 1, by=0.125)
prior = c(0.001, 0.001, 0.950, 0.008, 0.008, 0.008, 0.008, 0.008, 0.008)
sum(prior)
```

```
## [1] 1
```

尤度関数

```r
likelihood = function (p) p ^ 6 * (1 - p) ^ 4
```

事後分布

```r
posterior = c()
for (i in 1:length(p)) {
  posterior = c(posterior, likelihood(p[i]) * prior[i])
}
posterior = posterior / sum(posterior)
round(cbind(p, prior, posterior), 3)
```

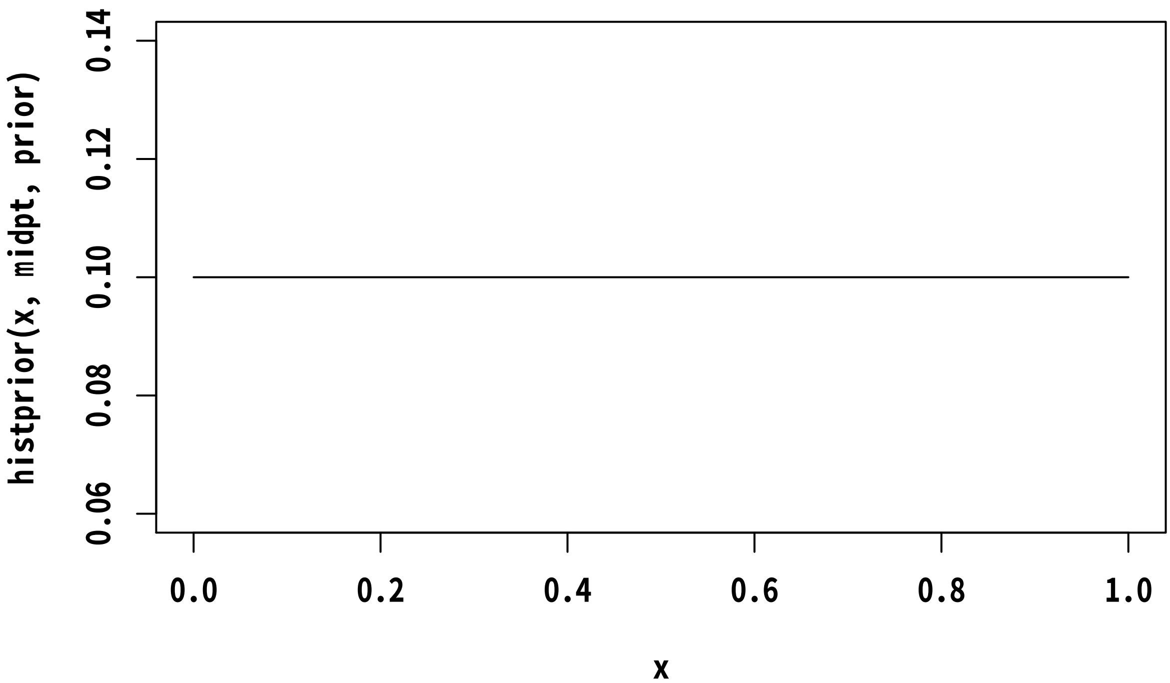
```
##           p prior posterior
##  [1,] 0.000 0.001     0.000
##  [2,] 0.125 0.001     0.000
##  [3,] 0.250 0.950     0.730
##  [4,] 0.375 0.008     0.034
##  [5,] 0.500 0.008     0.078
```

```
##   [6,] 0.625 0.008      0.094
##   [7,] 0.750 0.008      0.055
##   [8,] 0.875 0.008      0.009
##   [9,] 1.000 0.008      0.000
```

2-2

事前分布

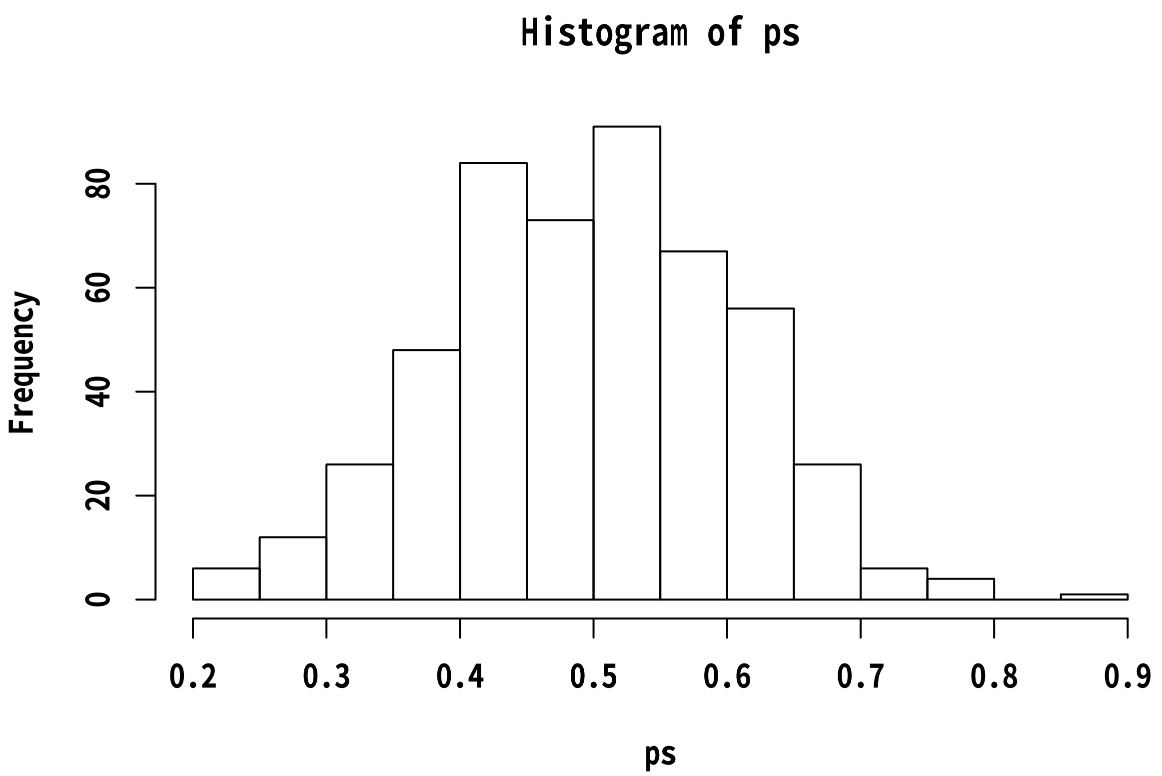
```
midpt = seq(0.05, 0.95, by = 0.1)
prior = rep(0.1, 10) # 無情報事前分布
curve(histprior(x, midpt, prior), from = 0, to = 1)
```



事後分布

```
p = seq(0, 1, length = 500)
posterior = c()
for (i in length(p)) {
  min_idx = which.min(abs(midpt - p[i]))
  posterior = c(posterior, dbeta(p, 10, 10) * prior[min_idx]) # コイン Head 10, Tail = 10
}
posterior = posterior / sum(posterior)
ps = sample(p, replace = TRUE, prob = posterior)
```

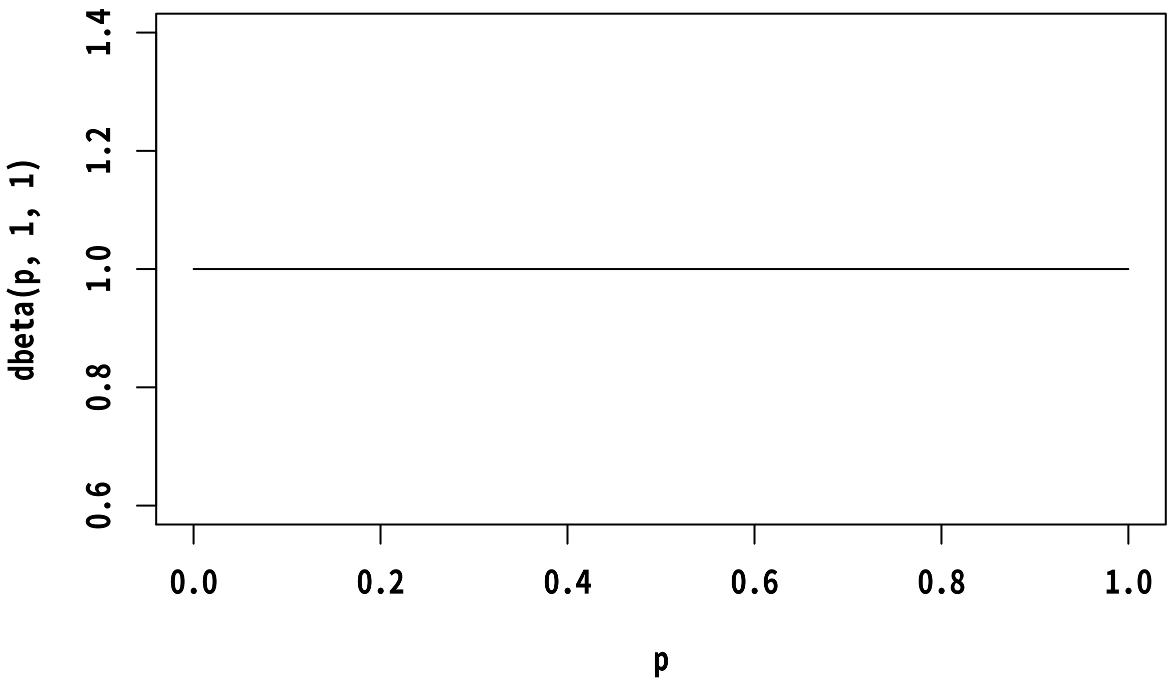
```r
hist(ps)
```



2-3

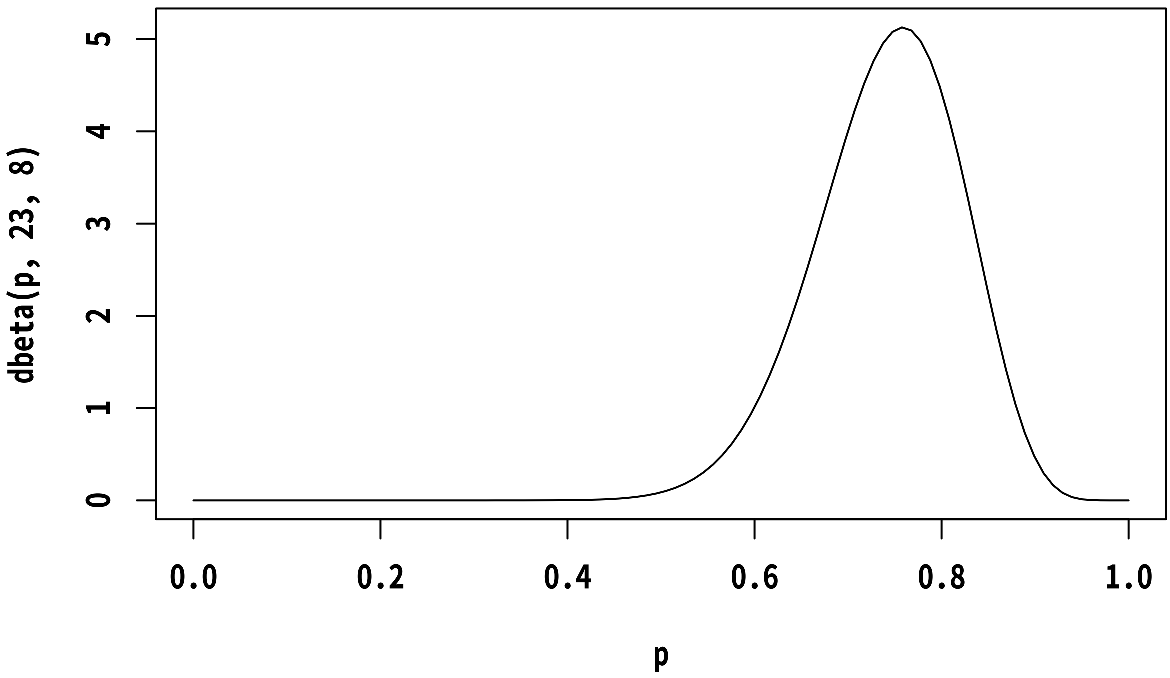
事前分布

```r
p = seq(0, 1, length=100)
plot(p, dbeta(p, 1, 1), type ="l")
```



事後分布

```
p = seq(0, 1, length=100)
plot(p, dbeta(p, 23, 8), type ="l")
```



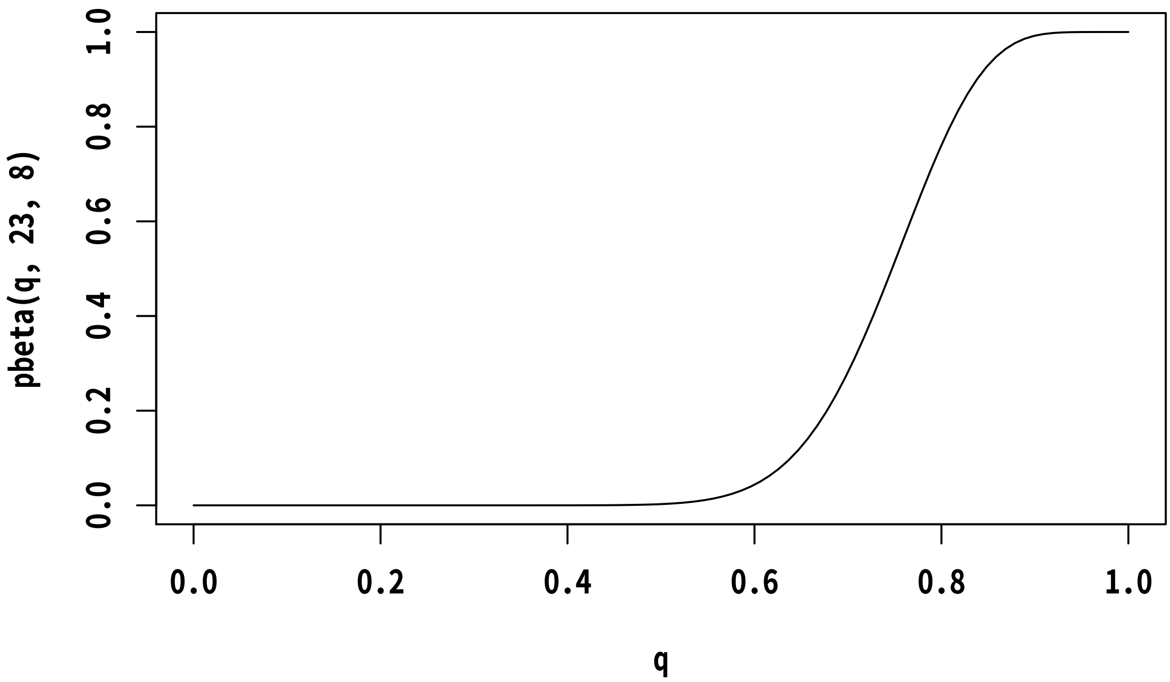
(a)

```
qbeta(c(0.5, 0.95), 23, 8)
```

```
## [1] 0.7471911 0.8598149
```

(b)

```
q = seq(0, 1, length=100)
plot(q, pbeta(q, 23, 8), type ="l")
```




```
1 - pbeta(0.6, 23, 8)
```

```
## [1] 0.9564759
```

(c)

```
rbeta(1000, 23, 8)
```

```
##    [1] 0.8070411 0.8255705 0.8477878 0.8108964 0.5317507 0.6689845 0.7476998
##    [8] 0.5928586 0.8221321 0.6747017 0.8718341 0.6920089 0.6583218 0.6658947
##   [15] 0.7164831 0.7388286 0.7370553 0.6340145 0.5931665 0.8259576 0.8674522
##   [22] 0.6588919 0.7786812 0.8250223 0.7543352 0.7646318 0.8076675 0.7075955
##   [29] 0.5620429 0.6311084 0.7760624 0.7828018 0.8001533 0.7533651 0.7204592
```

```
##  [36] 0.7126806 0.6963089 0.7200499 0.7241966 0.6954281 0.7771320 0.7758104
##  [43] 0.7877369 0.6878208 0.6566363 0.7038708 0.7457192 0.7594498 0.7318217
##  [50] 0.7833370 0.7323663 0.7595893 0.8384151 0.7640964 0.7508796 0.7798371
##  [57] 0.8113989 0.7495155 0.7004356 0.8477141 0.7682620 0.8378440 0.8100615
##  [64] 0.7901226 0.7423015 0.7588627 0.8255460 0.7433915 0.7981177 0.6595531
##  [71] 0.6546487 0.7899955 0.8291286 0.7201989 0.7269412 0.8071294 0.6084613
##  [78] 0.7088205 0.5755315 0.5836573 0.8408905 0.6462143 0.7539278 0.7029940
##  [85] 0.6070669 0.8072564 0.6368789 0.7252253 0.6867064 0.7091260 0.7579631
##  [92] 0.7055456 0.7030036 0.7053452 0.6349164 0.5357193 0.7662252 0.8386757
##  [99] 0.6946933 0.7423767 0.7814807 0.6893070 0.8334592 0.7954917 0.7478556
## [106] 0.7371275 0.8718947 0.7836972 0.6021016 0.7637951 0.6345157 0.8431801
## [113] 0.8037862 0.7230440 0.5500331 0.7371094 0.7422600 0.7501449 0.8072638
## [120] 0.7352074 0.7383002 0.6148018 0.7291778 0.6964099 0.8196658 0.8482959
## [127] 0.7675779 0.8182170 0.8151568 0.7886347 0.6429939 0.7646496 0.8817957
## [134] 0.6851345 0.6696884 0.7783403 0.7863805 0.7697624 0.7955889 0.8253679
## [141] 0.7418397 0.7582969 0.8290724 0.6987242 0.7283435 0.7632081 0.7547123
## [148] 0.7419842 0.8171104 0.5886940 0.9290006 0.6332434 0.6698825 0.7730383
## [155] 0.7879697 0.8929195 0.7335481 0.7399387 0.7832727 0.9222832 0.7702642
## [162] 0.7390387 0.7636731 0.7469337 0.7200959 0.6497416 0.8242184 0.7581898
## [169] 0.5190659 0.6497718 0.8684946 0.6674372 0.6521164 0.8013382 0.6950792
## [176] 0.7706440 0.7077917 0.8831823 0.6644828 0.6240237 0.8315838 0.7399149
## [183] 0.7649174 0.7452437 0.8858062 0.6824067 0.7632463 0.6645576 0.8393200
## [190] 0.8165355 0.8032959 0.8195353 0.7260750 0.8391408 0.7188959 0.6192670
## [197] 0.7599689 0.6548915 0.8878969 0.7628708 0.7885406 0.8283998 0.6493139
## [204] 0.7682822 0.8349291 0.7840529 0.6568872 0.5961189 0.6580423 0.6554338
## [211] 0.7488142 0.8080573 0.7680159 0.7891617 0.8185359 0.5432282 0.6931836
## [218] 0.6414934 0.7840668 0.7518499 0.6729806 0.8566883 0.6947518 0.7173330
## [225] 0.5577187 0.8557033 0.5904681 0.7283920 0.6248874 0.7212711 0.7954505
## [232] 0.6285365 0.7124906 0.7748415 0.7419581 0.7542591 0.6710782 0.7689502
## [239] 0.6163790 0.7503094 0.7726267 0.7371270 0.6984388 0.9049367 0.7393811
## [246] 0.7106324 0.8291581 0.6383072 0.7383878 0.8803357 0.6951761 0.8307502
## [253] 0.8302550 0.7704465 0.7542698 0.6884721 0.6627794 0.6559919 0.7221214
## [260] 0.5760306 0.7308500 0.6932339 0.6867618 0.7448929 0.7579052 0.7311930
## [267] 0.6671492 0.5220428 0.6240272 0.7335555 0.8288648 0.6173744 0.6046446
## [274] 0.6939766 0.7638322 0.6920132 0.6545225 0.7641409 0.6543275 0.7591319
## [281] 0.8598825 0.6979572 0.7953118 0.6935233 0.7290015 0.7672425 0.7884847
## [288] 0.7356503 0.7403201 0.8012068 0.7121388 0.7376999 0.7762729 0.6256374
## [295] 0.7294961 0.6149107 0.6873084 0.8253913 0.7396672 0.7893567 0.7584653
## [302] 0.7911967 0.7226496 0.7021289 0.6970805 0.7024061 0.7458640 0.7920454
## [309] 0.7164925 0.5837000 0.7585913 0.7461914 0.8344382 0.7005801 0.7084225
```

```
## [316] 0.8381066 0.7230324 0.8659108 0.7009245 0.7651718 0.6743535 0.8248178
## [323] 0.6060997 0.8429871 0.7400295 0.8357519 0.7086955 0.6839245 0.7535816
## [330] 0.6800196 0.7633497 0.7916140 0.6994650 0.8273658 0.6665231 0.7514537
## [337] 0.7595686 0.8031783 0.8325575 0.6480269 0.7431247 0.8202178 0.6378393
## [344] 0.6990551 0.6717766 0.7168446 0.8169514 0.6606395 0.7536755 0.7432141
## [351] 0.8071000 0.6321109 0.6716755 0.7166762 0.7824989 0.7126562 0.7766154
## [358] 0.5469188 0.8103667 0.6324445 0.8365032 0.6972919 0.7321594 0.7162194
## [365] 0.7610763 0.8847999 0.8316827 0.7982894 0.7094006 0.7744260 0.8202926
## [372] 0.7379838 0.8210720 0.7109790 0.7775590 0.5855964 0.7368752 0.6855817
## [379] 0.7239417 0.7668478 0.6716641 0.7215254 0.7790076 0.7714230 0.7496748
## [386] 0.8191977 0.7541359 0.6941162 0.6565750 0.7862854 0.7553311 0.7996159
## [393] 0.8001712 0.7197960 0.7016781 0.8194246 0.7323581 0.7870930 0.7812284
## [400] 0.7197432 0.8639072 0.8040182 0.6827346 0.7613100 0.6458100 0.6602294
## [407] 0.7640870 0.7634302 0.7474213 0.6843148 0.8018948 0.7341175 0.8126592
## [414] 0.7419077 0.6988210 0.7640474 0.7526541 0.7315503 0.5875670 0.8657462
## [421] 0.8137308 0.7260861 0.5642157 0.7094959 0.7594672 0.8761789 0.6892891
## [428] 0.7540278 0.7619244 0.6498313 0.6948838 0.7209213 0.7618178 0.8816037
## [435] 0.9027434 0.7468568 0.8079820 0.7286921 0.7457050 0.7457244 0.8355136
## [442] 0.7888787 0.8841203 0.8444017 0.5764540 0.7245525 0.8745875 0.5662076
## [449] 0.7667947 0.8355173 0.6834709 0.8444353 0.7404095 0.8172330 0.7615900
## [456] 0.7826947 0.7480997 0.6698544 0.8358671 0.8150572 0.6800893 0.8146567
## [463] 0.7086208 0.7917042 0.7131607 0.7416712 0.7719918 0.7360932 0.7540286
## [470] 0.8030830 0.8240125 0.8568588 0.7136817 0.6636194 0.7380640 0.7613453
## [477] 0.8201007 0.6533509 0.5366528 0.8827580 0.7591728 0.7605287 0.7517873
## [484] 0.8424112 0.7723770 0.6667527 0.6367214 0.5847001 0.6270906 0.6616011
## [491] 0.7868461 0.7440033 0.6588211 0.7130909 0.7427975 0.6641598 0.5909791
## [498] 0.6212812 0.5091258 0.6523157 0.8618974 0.5407816 0.6433221 0.7836734
## [505] 0.8185689 0.6808883 0.7757852 0.8350461 0.7494525 0.8827560 0.7756879
## [512] 0.7925148 0.7983371 0.7119857 0.7022579 0.8176977 0.6722857 0.7230909
## [519] 0.7195931 0.7404330 0.7159768 0.7900078 0.7303143 0.7640456 0.7582611
## [526] 0.7812874 0.7711055 0.8182638 0.6760750 0.7663402 0.7780504 0.5513241
## [533] 0.7382238 0.7354082 0.7137422 0.6990012 0.6981339 0.5549035 0.6903421
## [540] 0.7339360 0.7247849 0.5585907 0.6314387 0.8490969 0.8180988 0.7990809
## [547] 0.7324207 0.7552275 0.6973272 0.8218995 0.8122667 0.8054935 0.7614757
## [554] 0.6571675 0.6499711 0.7226440 0.6784612 0.6934226 0.7349363 0.7635993
## [561] 0.6881128 0.7468843 0.7957562 0.7933500 0.7248374 0.6887946 0.7189276
## [568] 0.8501931 0.8796187 0.7035483 0.6513043 0.7074881 0.7927238 0.5790580
## [575] 0.9141130 0.7439655 0.7024687 0.8048930 0.8529540 0.7873532 0.7459709
## [582] 0.6863400 0.7610995 0.6979695 0.8479236 0.7332971 0.8844835 0.7638638
## [589] 0.6656249 0.7742853 0.6327511 0.7004418 0.6306536 0.7892471 0.6812490
```

```
## [596] 0.7088840 0.7987329 0.8083374 0.7434315 0.7941877 0.7239864 0.7042102
## [603] 0.7358163 0.7601263 0.6593779 0.6252387 0.6449974 0.6790615 0.7760217
## [610] 0.6873391 0.8117366 0.8553483 0.7785969 0.6360114 0.8293971 0.8506065
## [617] 0.8428015 0.7175246 0.7628640 0.6574936 0.8202796 0.6982605 0.8334354
## [624] 0.6917684 0.7681091 0.6699725 0.7642126 0.8233342 0.7635729 0.7768003
## [631] 0.7529088 0.7466654 0.8479153 0.7464774 0.8891925 0.6612791 0.8011480
## [638] 0.5950652 0.7917699 0.6893540 0.8250828 0.8188693 0.8173764 0.7767781
## [645] 0.7250276 0.6698057 0.7603657 0.7732190 0.6956249 0.6967454 0.7704556
## [652] 0.7077197 0.8055186 0.6932845 0.6532176 0.5895797 0.7517023 0.8493274
## [659] 0.7050852 0.5728057 0.7792230 0.6549229 0.7944152 0.7165643 0.7190378
## [666] 0.8800473 0.7506635 0.7948250 0.7003562 0.7565783 0.7401979 0.7528181
## [673] 0.6318098 0.6863607 0.8364215 0.6200527 0.7708836 0.6851276 0.7921321
## [680] 0.8935590 0.8700204 0.6990532 0.6359755 0.7215375 0.7767206 0.8156982
## [687] 0.6600185 0.8512415 0.8219685 0.7369188 0.7854480 0.8981731 0.7518289
## [694] 0.8972693 0.7497250 0.6349855 0.8161118 0.7379646 0.7641588 0.6875408
## [701] 0.6284696 0.6735138 0.8672343 0.6752612 0.7560825 0.7027188 0.7492405
## [708] 0.7163056 0.6537679 0.7096351 0.7749781 0.8164420 0.6758390 0.7716105
## [715] 0.7801563 0.6013999 0.7752269 0.7538624 0.7172303 0.8145564 0.8171091
## [722] 0.8290877 0.9077562 0.5562815 0.7956330 0.7698635 0.7546775 0.8943774
## [729] 0.8204628 0.8718571 0.6704152 0.7964022 0.7253794 0.8432499 0.6448720
## [736] 0.8700887 0.7544522 0.8418569 0.7368648 0.6371424 0.7630977 0.7034541
## [743] 0.5927178 0.9148894 0.7756911 0.8640674 0.7855556 0.8014474 0.7834533
## [750] 0.8911126 0.7775732 0.7750922 0.6028524 0.5781951 0.7708878 0.7799956
## [757] 0.8635073 0.7487080 0.6427317 0.7950094 0.7736588 0.7063747 0.6871343
## [764] 0.8233889 0.8104394 0.5645456 0.7601336 0.7397553 0.8304406 0.7293278
## [771] 0.7964878 0.6869039 0.7742797 0.6931350 0.8203228 0.8137063 0.7708619
## [778] 0.6145557 0.7738697 0.8328718 0.9010301 0.8139411 0.7925639 0.8428802
## [785] 0.7426531 0.7254006 0.8897288 0.7565929 0.7545048 0.8091935 0.6993878
## [792] 0.7730862 0.7778719 0.8630763 0.7481346 0.7480236 0.6700612 0.7480113
## [799] 0.7306402 0.7299858 0.6601421 0.6505850 0.7568708 0.7999926 0.7587764
## [806] 0.8185629 0.8394630 0.7067662 0.6742711 0.8018482 0.7968193 0.8711253
## [813] 0.6652598 0.7264986 0.6757255 0.8486482 0.6540638 0.8601133 0.7099514
## [820] 0.7208451 0.7428981 0.7909909 0.8262645 0.8701404 0.6642313 0.7839375
## [827] 0.6583020 0.7323864 0.7740838 0.9131760 0.7224556 0.7583922 0.7405111
## [834] 0.6915171 0.7782397 0.5954660 0.8348261 0.8050470 0.6696192 0.8349710
## [841] 0.7685809 0.6831547 0.7968662 0.7642553 0.6002234 0.7353941 0.7627284
## [848] 0.7130976 0.8264627 0.6391013 0.7821046 0.6860732 0.7493760 0.9097083
## [855] 0.7326167 0.6781458 0.8335945 0.8085110 0.7585149 0.6987125 0.6776713
## [862] 0.6112777 0.7374377 0.8570343 0.7630491 0.8346732 0.6948891 0.6524121
## [869] 0.8430591 0.8183503 0.8689370 0.7642658 0.5932184 0.5841087 0.7978077
```

```
## [876] 0.8418604 0.6476233 0.7420127 0.8378920 0.8687992 0.5272376 0.6715673
## [883] 0.7350945 0.6592986 0.7172388 0.7642672 0.6832443 0.7926537 0.8146516
## [890] 0.7676372 0.6689440 0.8481611 0.7369680 0.6996969 0.5663915 0.7581222
## [897] 0.6916518 0.7066210 0.7807791 0.6800818 0.7960751 0.6188562 0.6649213
## [904] 0.7370349 0.7183507 0.7927439 0.8062239 0.5959084 0.6429020 0.7621810
## [911] 0.6839450 0.6340319 0.6151831 0.8205883 0.8575990 0.6386667 0.6549673
## [918] 0.8217612 0.8170561 0.6288033 0.8063646 0.8282542 0.7952929 0.7166753
## [925] 0.8219808 0.5883009 0.7009122 0.8513631 0.8168992 0.7798705 0.6701205
## [932] 0.5744546 0.8235316 0.7994704 0.7624079 0.6845470 0.7512998 0.5913220
## [939] 0.8525227 0.7880262 0.7584610 0.7029587 0.7777436 0.7702033 0.7696372
## [946] 0.7290124 0.6610844 0.7261746 0.6267911 0.6189707 0.6207055 0.8149847
## [953] 0.7872201 0.5052213 0.6155507 0.8418012 0.7163843 0.8423057 0.7701817
## [960] 0.7845378 0.6769304 0.8018098 0.8477477 0.7565764 0.7765238 0.7326627
## [967] 0.7578607 0.8392483 0.7536428 0.7477384 0.7299586 0.7138586 0.8806826
## [974] 0.7794980 0.8129605 0.7381021 0.6962678 0.7871156 0.7770618 0.7138019
## [981] 0.7126055 0.5874733 0.6682449 0.7709809 0.8535689 0.7055220 0.7451246
## [988] 0.5317241 0.7095301 0.6127224 0.7279205 0.7814333 0.7892464 0.8561399
## [995] 0.7799275 0.7564642 0.6992476 0.7659759 0.7648003 0.6738487
```

(d)

さらに、10 人いる場合の高校を卒業する人数 X の予測分布を求める.

$$p(x) = \int_0^1 \binom{10}{x} p^x (1-p)^{10-x} Beta(p|23, 8) dp$$

$$\begin{aligned}
p(X=9) &= \int_0^1 \binom{10}{9} p^9 (1-p)^{10-9} Beta(p|23, 8) dp \\
&= \int_0^1 10 \cdot p^9 (1-p) Beta(p|23, 8) dp \\
&= \frac{10}{B(23, 8)} \int_0^1 p^9 (1-p) p^{22} (1-p)^7 dp \\
&= \frac{10}{B(23, 8)} \int_0^1 p^{31} (1-p)^8 dp \\
&= \frac{10}{B(23, 8)} \int_0^1 p^{32-1} (1-p)^{9-1} dp \\
&= \frac{10}{B(23, 8)} B(32, 9)
\end{aligned}$$

```
10 * beta(32, 9) / beta(23, 8)
```

```
## [1] 0.1902656
```

$$
\begin{aligned}
p(X=10) &= \int_0^1 \binom{10}{10} p^{10}(1-p)^{10-10} Beta(p|23,8)dp \\
&= \int_0^1 p^{10} Beta(p|23,8)dp \\
&= \frac{1}{B(23,8)} \int_0^1 p^{10}p^{22}(1-p)^7 dp \\
&= \frac{1}{B(23,8)} \int_0^1 p^{32}(1-p)^7 dp \\
&= \frac{1}{B(23,8)} \int_0^1 p^{33-1}(1-p)^{8-1} dp \\
&= \frac{1}{B(23,8)} B(33,8)
\end{aligned}
$$

```r
beta(33, 8) / beta(23, 8)
```

```
## [1] 0.07610622
```

サンプリングすると以下のようにできる.

よって、このときの $X = 9 \; or \; X = 10$ となる確率を求めれば良い.

$$
p \sim Beta(p|23,8) x \sim Bin(x|10,p)
$$

```r
p = rbeta(10000, 23, 8)
x = rbinom(10000, 10, p)
table(x) / 10000
```

```
## x
##      1      2      3      4      5      6      7      8      9     10
## 0.0002 0.0024 0.0094 0.0322 0.0787 0.1480 0.2250 0.2493 0.1818 0.0730
```

理論値と近い値となっていることが確認できる.

2-4

(a)

```r
p = seq(0.1, 0.5, by=0.1)
p
```

```
## [1] 0.1 0.2 0.3 0.4 0.5
```

```r
prior = c(0.50, 0.2, 0.2, 0.05, 0.05)
mean = sum(p * prior)
```

```r
mean
```

```
## [1] 0.195
```

```r
sd = sqrt(sum((p - mean)^2 * prior))
sd
```

```
## [1] 0.1160819
```

$Beta(p|3, 12)$ の場合、サンプリング近似を行う.

```r
sample = rbeta(10000, shape1 = 3, shape2 = 12)
mean(sample)
```

```
## [1] 0.1993212
```

```r
sd(sample)
```

```
## [1] 0.1002495
```

(b)
通学者数 Y に関して、以下の予測分布が計算できる.

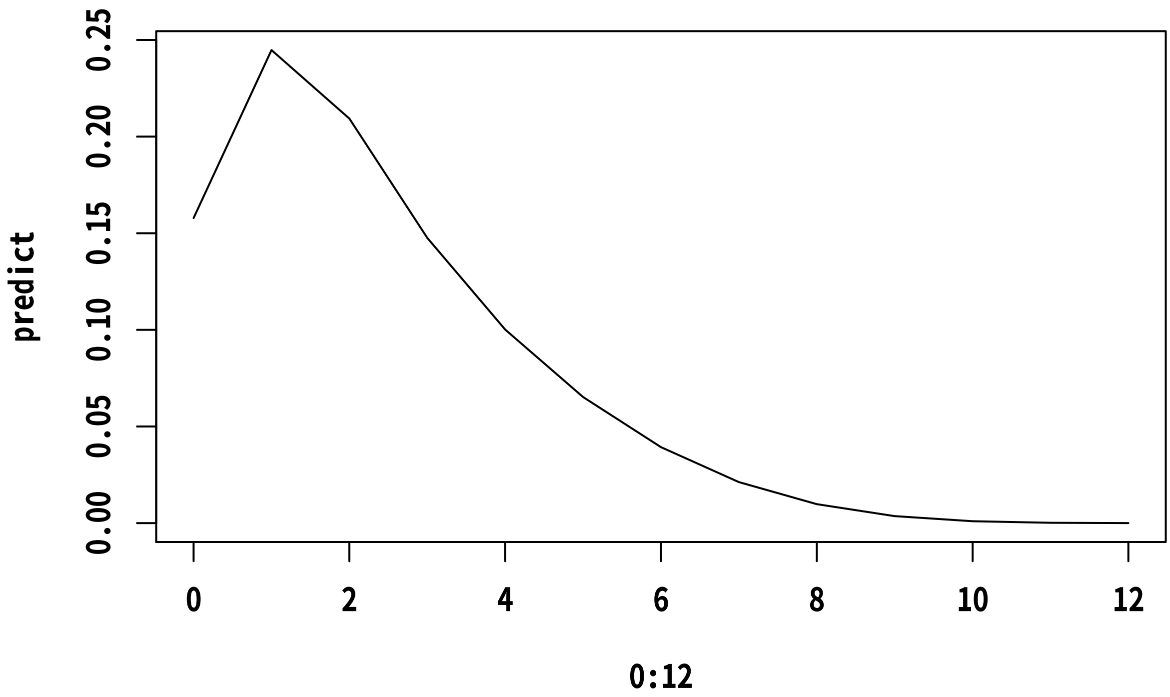
■離散事前分布

$$p(y) = \sum_p \binom{12}{y} p^y (1-p)^{12-y} g(p)$$

```r
predict = c()
for (y in 0:12) {
  p_y = 0
  for (i in 1:length(p)) {
    p_y = p_y + choose(12, y) * p[i]^y * (1 - p[i])^(12 - y) * prior[i]
  }
  predict = c(predict, p_y)
}
predict
```

```
##  [1] 0.1578479672 0.2447719936 0.2093137913 0.1475812240 0.1001416403
##  [6] 0.0652427436 0.0393037888 0.0212231429 0.0098095892 0.0036170714
## [11] 0.0009692955 0.0001645993 0.0000131530
```

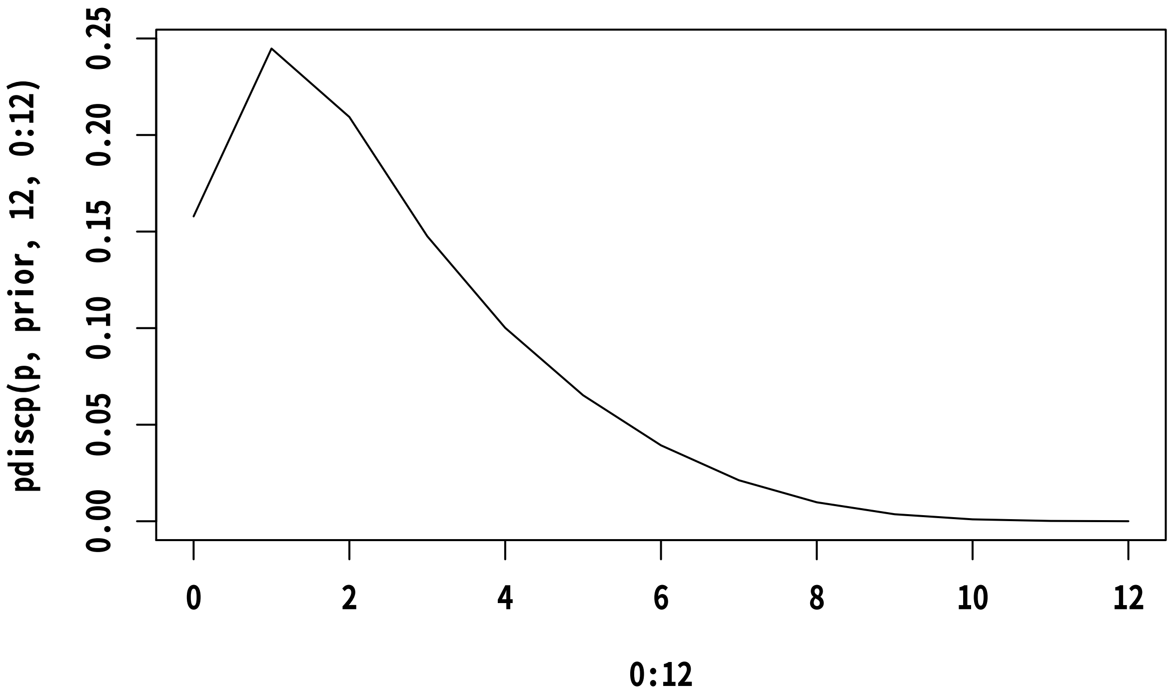
```r
plot(0:12, predict, type = "l")
```




```r
pdiscp(p, prior, 12, 0:12)
```

```
##  [1] 0.1578479672 0.2447719936 0.2093137913 0.1475812240 0.1001416403
##  [6] 0.0652427436 0.0393037888 0.0212231429 0.0098095892 0.0036170714
## [11] 0.0009692955 0.0001645993 0.0000131530
```

```r
plot(0:12, pdiscp(p, prior, 12, 0:12), type = "l")
```



■ベータ分布

$$
\begin{aligned}
p(y) &= \int_0^1 \binom{12}{y} p^y (1-p)^{12-y} \frac{1}{B(3,12)} p^{3-1}(1-p)^{12-1} dp \\
&= \binom{12}{y} \frac{1}{B(3,12)} \int_0^1 p^{y+2}(1-p)^{23-y} dp \\
&= \binom{12}{y} \frac{1}{B(3,12)} B(y+3, 24-y)
\end{aligned}
$$

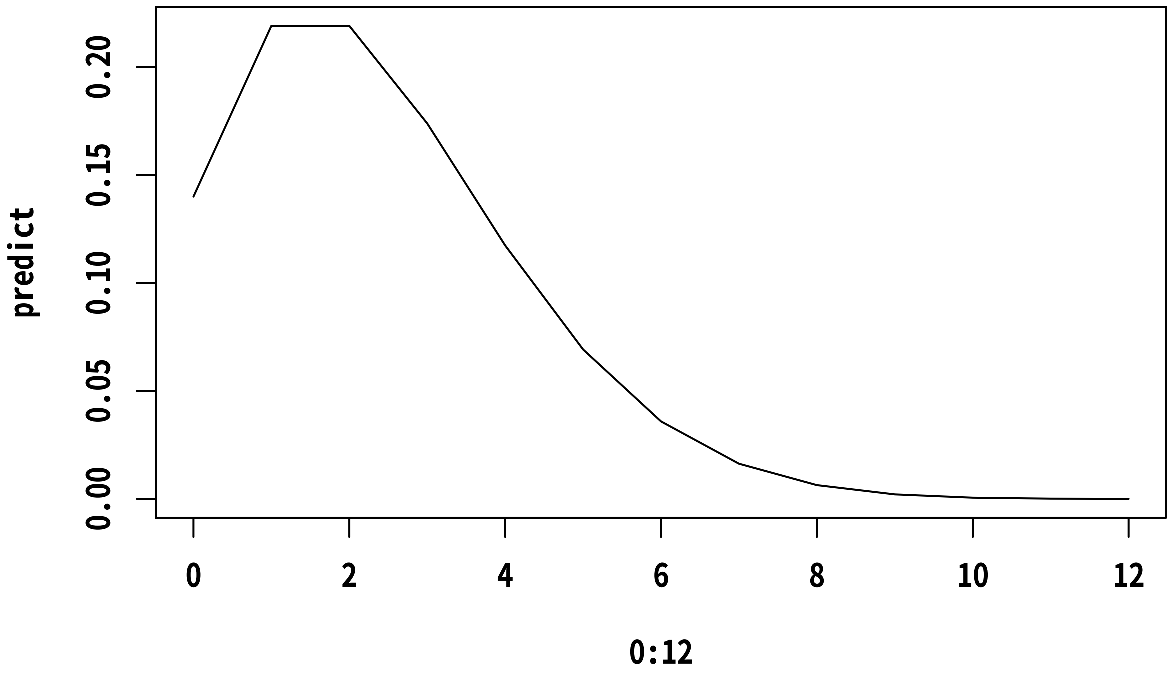
```
predict = c()
for (y in 0:12) {
  p_y = choose(12, y) * beta(y + 3, 24 - y) / beta(3, 12)
  predict = c(predict, p_y)
}
predict
```

```
##  [1] 1.400000e-01 2.191304e-01 2.191304e-01 1.739130e-01 1.173913e-01
##  [6] 6.919908e-02 3.588101e-02 1.628214e-02 6.360210e-03 2.072957e-03
## [11] 5.330462e-04 9.691749e-05 9.422533e-06
```

```
sum(predict)
```

```
## [1] 1
```

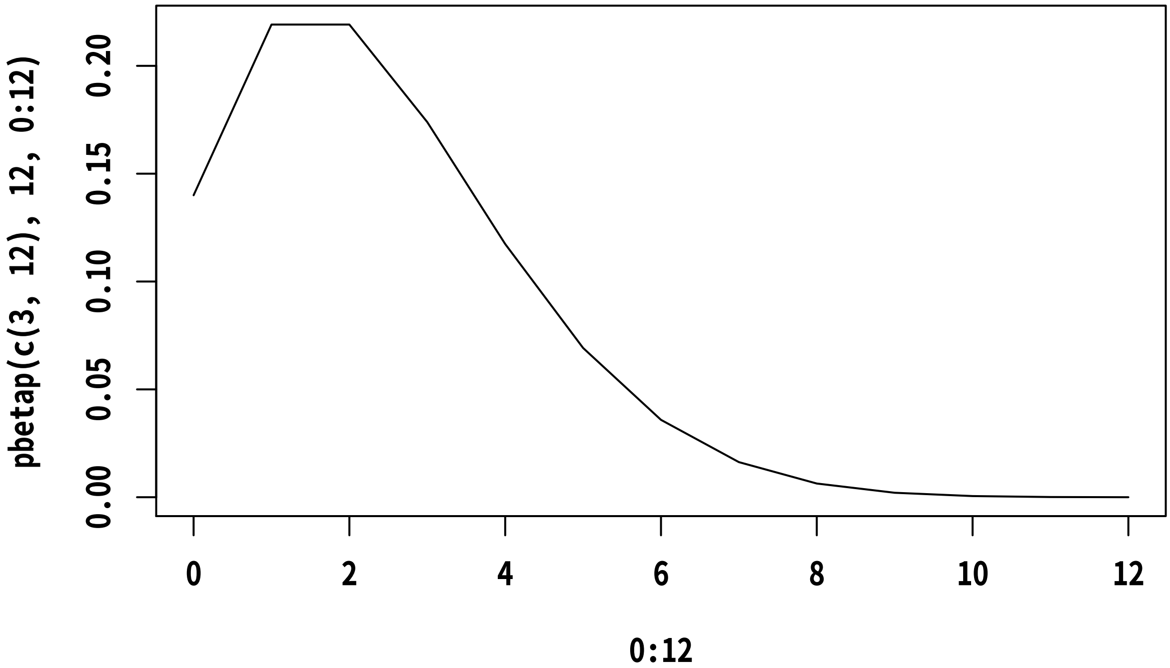
```r
plot(0:12, predict, type = "l")
```




```r
pbetap(c(3, 12), 12, 0:12)
```

```
##  [1] 1.400000e-01 2.191304e-01 2.191304e-01 1.739130e-01 1.173913e-01
##  [6] 6.919908e-02 3.588101e-02 1.628214e-02 6.360210e-03 2.072957e-03
## [11] 5.330462e-04 9.691749e-05 9.422533e-06
```

```r
plot(0:12, pbetap(c(3, 12), 12, 0:12), type = "l")
```



2-5

(a)

```
mu = seq(20, 70, by = 10)
prior = c(0.1, 0.15, 0.25, 0.25, 0.15, 0.1)
mu
```

```
## [1] 20 30 40 50 60 70
```

```
prior
```

```
## [1] 0.10 0.15 0.25 0.25 0.15 0.10
```

(b)

```
y = c(38.6, 42.4, 57.5, 40.5, 51.7, 67.1, 33.4, 60.9, 64.1, 40.1, 40.7, 6.4)
ybar = mean(y)
ybar
```

```
## [1] 45.28333
```

(c)

```r
likelihood = function (mu) exp(-1 * length(y) / (2 * 100) * (mu - ybar)^2)
like = likelihood(mu)
like
```

```
## [1] 2.201480e-17 8.192991e-07 1.873425e-01 2.632064e-01 2.272076e-06
## [6] 1.205079e-16
```

(d)

```r
post = prior * like
post = post / sum(post)
post
```

```
## [1] 1.954479e-17 1.091063e-06 4.158078e-01 5.841881e-01 3.025731e-06
## [6] 1.069871e-16
```

(e)

```r
dist = cbind(mu, post)
dist
```

```
##      mu         post
## [1,] 20 1.954479e-17
## [2,] 30 1.091063e-06
## [3,] 40 4.158078e-01
## [4,] 50 5.841881e-01
## [5,] 60 3.025731e-06
## [6,] 70 1.069871e-16
```

```r
discint(dist, 0.8)
```

```
## $prob
## [1] 0.9999959
##
## $set
## [1] 40 50
```

2-6

(a)

```
lambda = c(0.5, 1, 1.5, 2, 2.5, 3)
prior = c(0.1, 0.2, 0.3, 0.2, 0.15, 0.05)
likelihood = function (lambda) exp(-6 * lambda) * (6 * lambda)^12
post = prior * likelihood(lambda)
post = post / sum(post)
cbind(lambda, prior, round(post, 2))
```

```
##      lambda prior
## [1,]    0.5  0.10 0.00
## [2,]    1.0  0.20 0.04
## [3,]    1.5  0.30 0.36
## [4,]    2.0  0.20 0.37
## [5,]    2.5  0.15 0.20
## [6,]    3.0  0.05 0.03
```

(b)

$$p(y|\lambda) = \exp(-\lambda)\frac{\lambda^y}{y!}$$

7 日間故障が起きない確率は、$p(y=0)^7$

$$p(y=0|\lambda)^7 = \exp(-\lambda)^7 = \exp(-7\lambda)$$

よって、予測確率は

$$p(y=0) = \sum_{\lambda} p(y=0|\lambda)p(\lambda)$$

```
predict = 0
for (i in 1:length(lambda)) {
  predict = predict + exp(-7 * lambda[i]) * post[i]
}
predict
```

```
## [1] 4.640932e-05
```

3-1

```r
y = c(0, 10, 9, 8 ,11, 3, 3, 8, 8 ,11)
```

(a)

```r
grid = seq(-2, 12, by = 0.1)
grid
```

```
##   [1] -2.0 -1.9 -1.8 -1.7 -1.6 -1.5 -1.4 -1.3 -1.2 -1.1 -1.0 -0.9 -0.8 -0.7 -0.6
##  [16] -0.5 -0.4 -0.3 -0.2 -0.1  0.0  0.1  0.2  0.3  0.4  0.5  0.6  0.7  0.8  0.9
##  [31]  1.0  1.1  1.2  1.3  1.4  1.5  1.6  1.7  1.8  1.9  2.0  2.1  2.2  2.3  2.4
##  [46]  2.5  2.6  2.7  2.8  2.9  3.0  3.1  3.2  3.3  3.4  3.5  3.6  3.7  3.8  3.9
##  [61]  4.0  4.1  4.2  4.3  4.4  4.5  4.6  4.7  4.8  4.9  5.0  5.1  5.2  5.3  5.4
##  [76]  5.5  5.6  5.7  5.8  5.9  6.0  6.1  6.2  6.3  6.4  6.5  6.6  6.7  6.8  6.9
##  [91]  7.0  7.1  7.2  7.3  7.4  7.5  7.6  7.7  7.8  7.9  8.0  8.1  8.2  8.3  8.4
## [106]  8.5  8.6  8.7  8.8  8.9  9.0  9.1  9.2  9.3  9.4  9.5  9.6  9.7  9.8  9.9
## [121] 10.0 10.1 10.2 10.3 10.4 10.5 10.6 10.7 10.8 10.9 11.0 11.1 11.2 11.3 11.4
## [136] 11.5 11.6 11.7 11.8 11.9 12.0
```

(b)

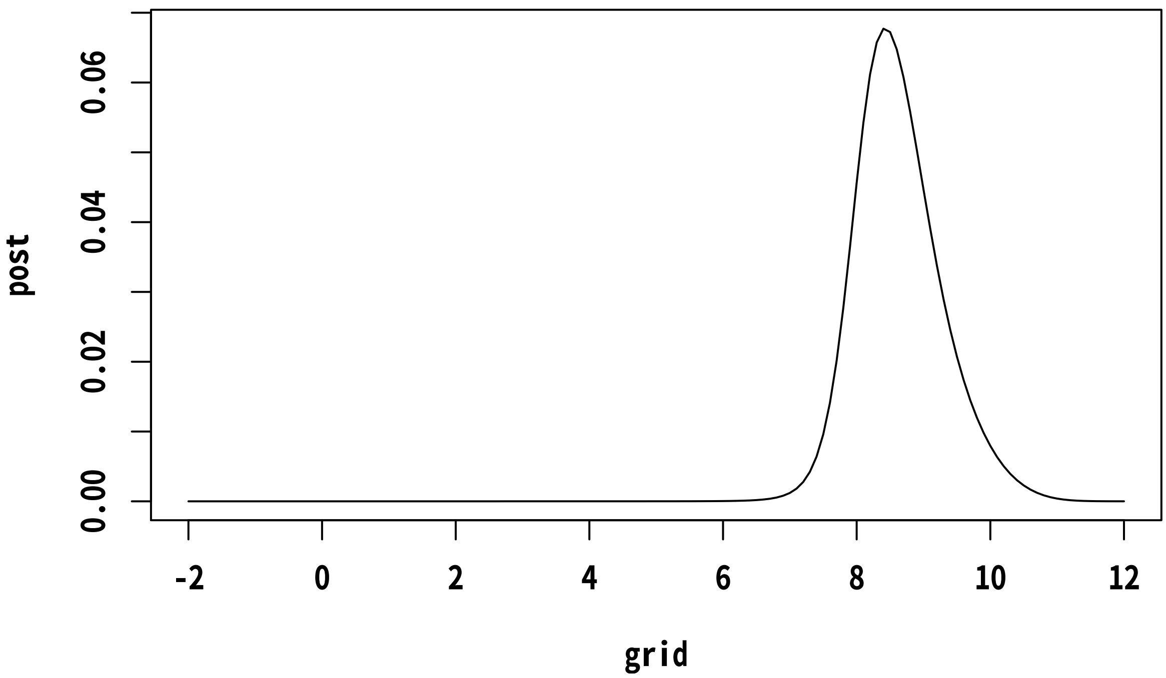
```r
post = c()
for (i in 1:length(grid)) {
  post = c(post, prod(1 / (1 + (y - grid[i])^2)))
}
post = post / sum(post)
post
```

```
##   [1] 1.126701e-12 1.496731e-12 1.998369e-12 2.681960e-12 3.618341e-12
##   [6] 4.907571e-12 6.691425e-12 9.171224e-12 1.263306e-11 1.748306e-11
##  [11] 2.429564e-11 3.387761e-11 4.734937e-11 6.624058e-11 9.259000e-11
##  [16] 1.290277e-10 1.788063e-10 2.457439e-10 3.340655e-10 4.481838e-10
##  [21] 5.925432e-10 7.716980e-10 9.907641e-10 1.256265e-09 1.577282e-09
##  [26] 1.966808e-09 2.443283e-09 3.032471e-09 3.769918e-09 4.704408e-09
##  [31] 5.902894e-09 7.457609e-09 9.496316e-09 1.219707e-08 1.580947e-08
##  [36] 2.068510e-08 2.732094e-08 3.642051e-08 4.897829e-08 6.639260e-08
##  [41] 9.060846e-08 1.242817e-07 1.709331e-07 2.350204e-07 3.217946e-07
##  [46] 4.367471e-07 5.844460e-07 7.667233e-07 9.805828e-07 1.216761e-06
```

```
##  [51] 1.460147e-06 1.692662e-06 1.897806e-06 2.064831e-06 2.190700e-06
##  [56] 2.279372e-06 2.339409e-06 2.381300e-06 2.415471e-06 2.451242e-06
##  [61] 2.496573e-06 2.558281e-06 2.642503e-06 2.755231e-06 2.902879e-06
##  [66] 3.092862e-06 3.334235e-06 3.638441e-06 4.020242e-06 4.498945e-06
##  [71] 5.100067e-06 5.857626e-06 6.817372e-06 8.041376e-06 9.614637e-06
##  [76] 1.165467e-05 1.432560e-05 1.785902e-05 2.258530e-05 2.898085e-05
##  [81] 3.774050e-05 4.988895e-05 6.695444e-05 9.124112e-05 1.262594e-04
##  [86] 1.774089e-04 2.530649e-04 3.663029e-04 5.376134e-04 7.991090e-04
##  [91] 1.200852e-03 1.819899e-03 2.772146e-03 4.225372e-03 6.408220e-03
##  [96] 9.603697e-03 1.410919e-02 2.014539e-02 2.771668e-02 3.647081e-02
## [101] 4.565149e-02 5.422549e-02 6.116538e-02 6.574884e-02 6.771668e-02
## [106] 6.723693e-02 6.474535e-02 6.077250e-02 5.582661e-02 5.034240e-02
## [111] 4.467224e-02 3.909187e-02 3.380540e-02 2.894759e-02 2.458832e-02
## [116] 2.074307e-02 1.738861e-02 1.448031e-02 1.196713e-02 9.801610e-03
## [121] 7.944263e-03 6.363233e-03 5.031358e-03 3.923044e-03 3.012578e-03
## [126] 2.274241e-03 1.683455e-03 1.217945e-03 8.581987e-04 5.871043e-04
## [131] 3.891602e-04 2.498625e-04 1.556626e-04 9.443623e-05 5.606765e-05
## [136] 3.275997e-05 1.894430e-05 1.089840e-05 6.264988e-06 3.611696e-06
## [141] 2.093835e-06
```

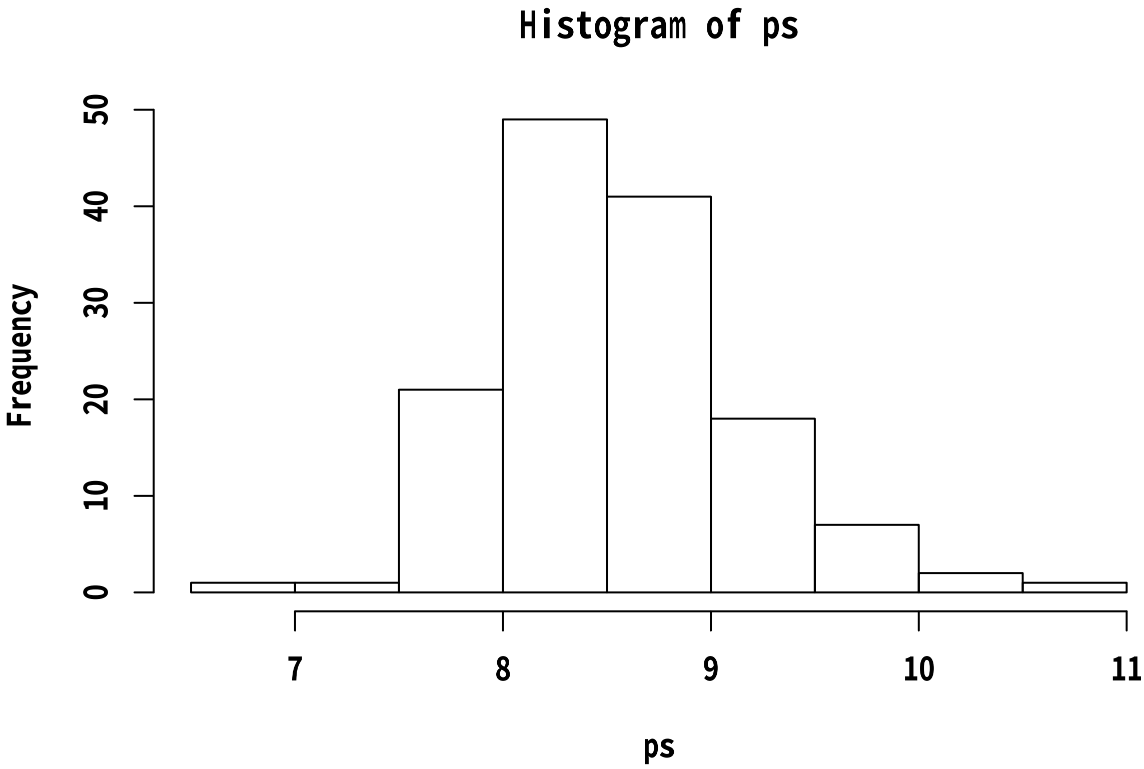
(c)

```
plot(grid, post, type = "l")
```



(d)

```r
ps = sample(grid, replace = TRUE, prob = post)
hist(ps)
```




```r
mean(ps)
```

```
## [1] 8.588652
```

```r
sd(ps)
```

```
## [1] 0.6105141
```

3-2

(a)

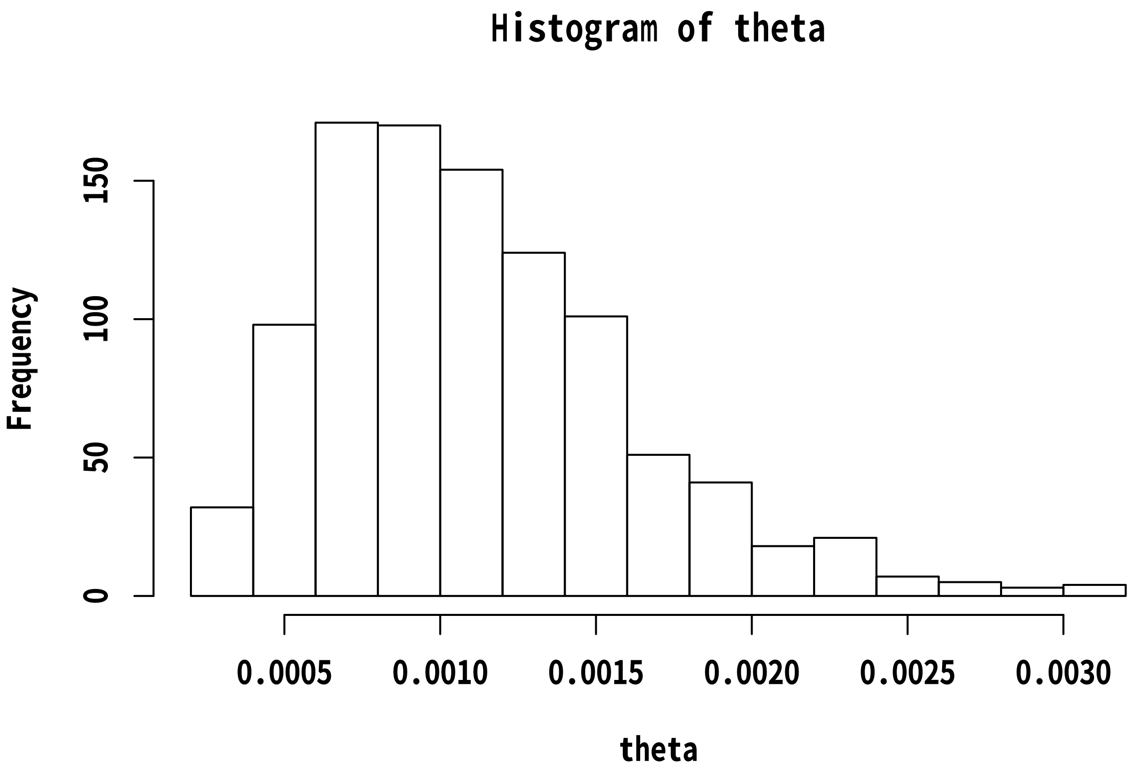
$$g(\lambda|data) \propto \lambda^{-n-1} \exp\left(-\frac{s}{\lambda}\right)$$

$$
\begin{aligned}
g(\theta|data) &\propto \theta^{n+1} \exp(-s\theta) \cdot |\frac{d\lambda}{d\theta}| \\
\theta = \frac{1}{\lambda} \qquad &= \theta^{n+1} \exp(-s\theta) \cdot |\frac{d}{d\theta}\theta^{-1}| \\
&= \theta^{n+1} \exp(-s\theta) \cdot |-1 \cdot \theta^{-2}| \\
&= \theta^{n-1} \exp(-s\theta)
\end{aligned}
$$

(b)

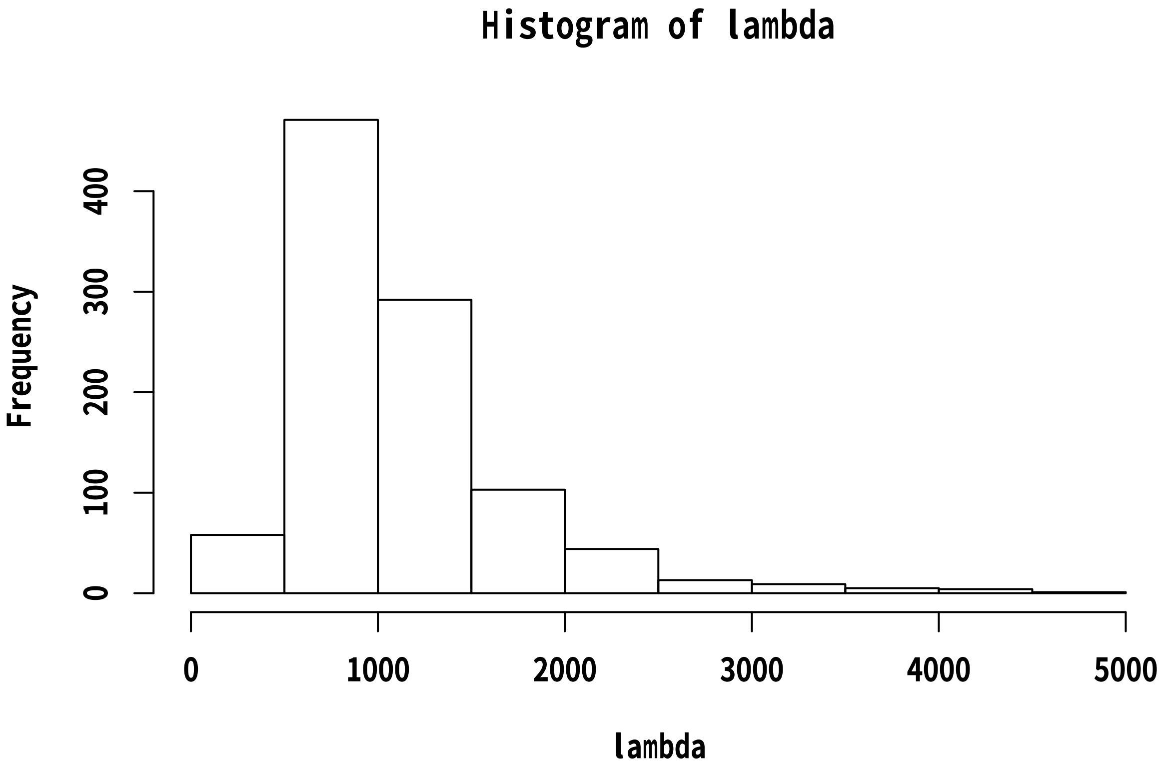
1 つの電球が切れるまでの時間を X とすると、$X \sim Exp(x|\beta)$ と考えられる.

```
x = c(751, 594, 1213, 1126, 819)
n = length(x)
s = sum(x)
theta = rgamma(1000, n, s)
hist(theta)
```



(c)

```
lambda = 1 / theta
hist(lambda)
```



(d)

```r
length(lambda[lambda > 1000]) / 1000
```

```
## [1] 0.471
```

3-3

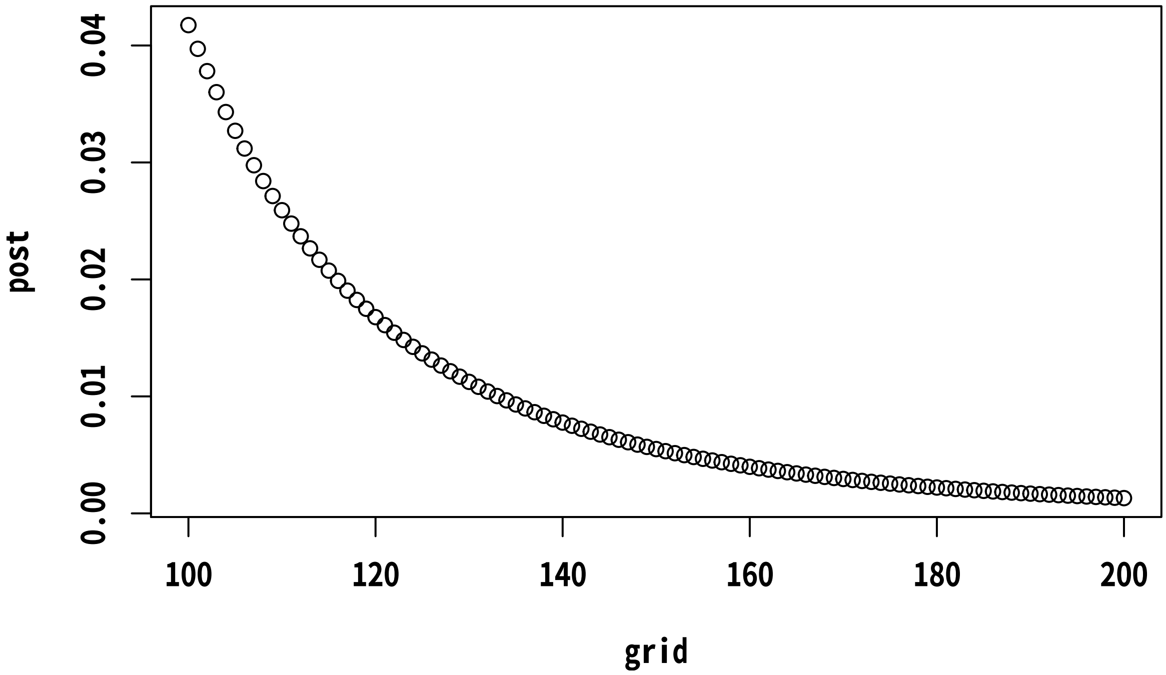
$$g(N|y) \propto \frac{1}{N^n}, \quad y_{(n)} \leq N \leq B$$

ここで、N は最大値の分布であることから、$y_1, \ldots, y_n$ を観測した際には、初期の候補 $1 \leq N \leq B$ から上記へと分布が縮小する．(観測値の最大値以上の値になる)

(a)

```r
n = 5
y = c(43, 24, 100, 35, 85)
B = 200
grid = seq(max(y), B, by = 1)
post = 1 / grid^n
```

```r
post = post / sum(post)
plot(grid, post)
```



(b)

```r
N = sample(grid, size = 1000, replace = TRUE, prob = post)
```

```r
mean(N)
```

```
## [1] 125.043
```

```r
sd(N)
```

```
## [1] 23.26539
```

(c)

```r
length(N[N > 150]) / 1000
```

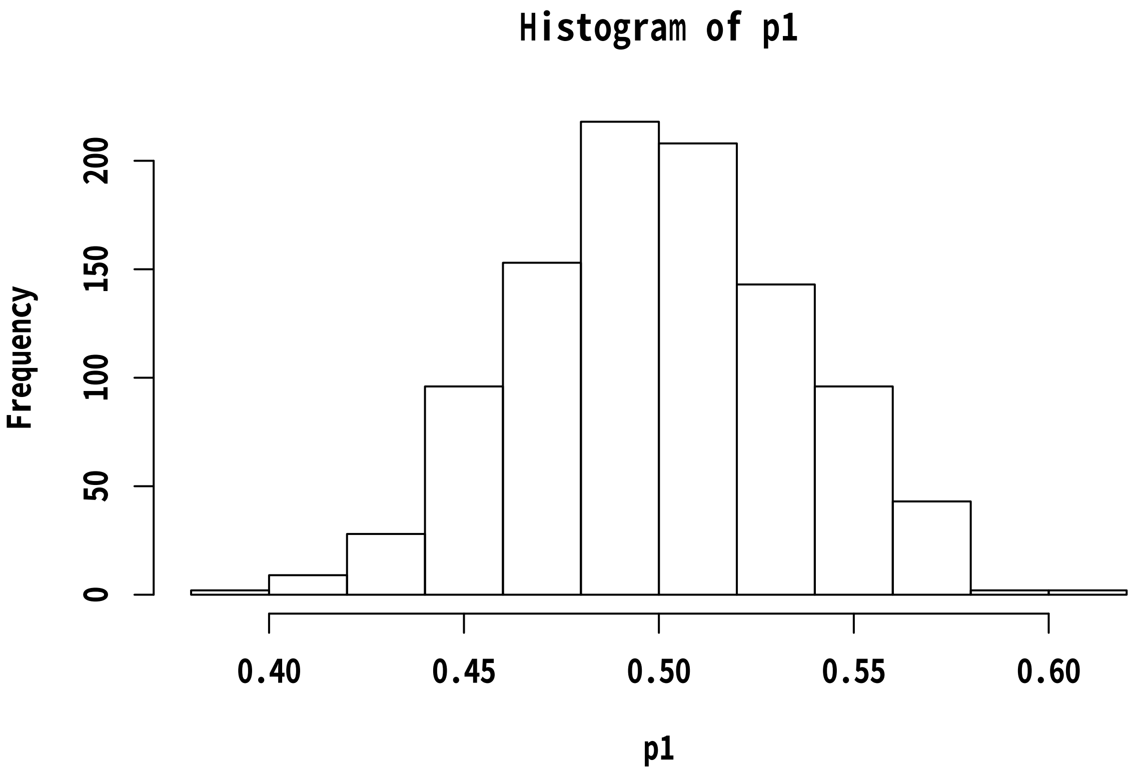
```
## [1] 0.158
```

3-4

(a)

■P1

```r
m = 1000
p1 = rbeta(m, 100, 100)
hist(p1)
```

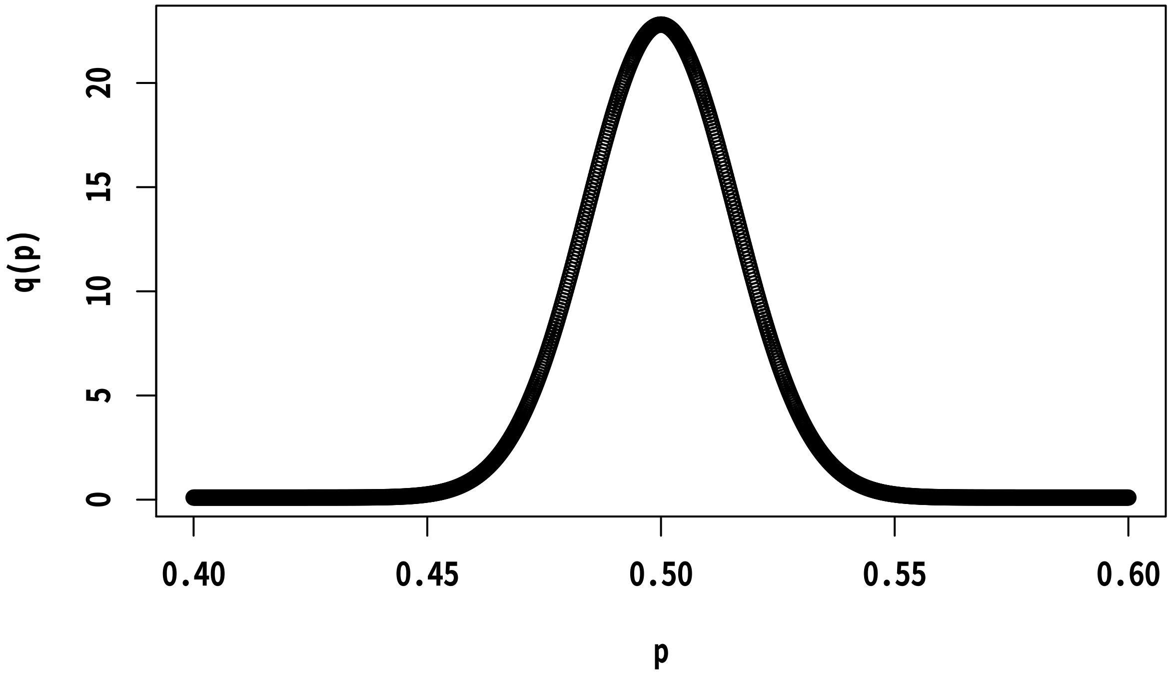



```r
length(p1[0.44 < p1 & p1 < 0.56]) / m
```

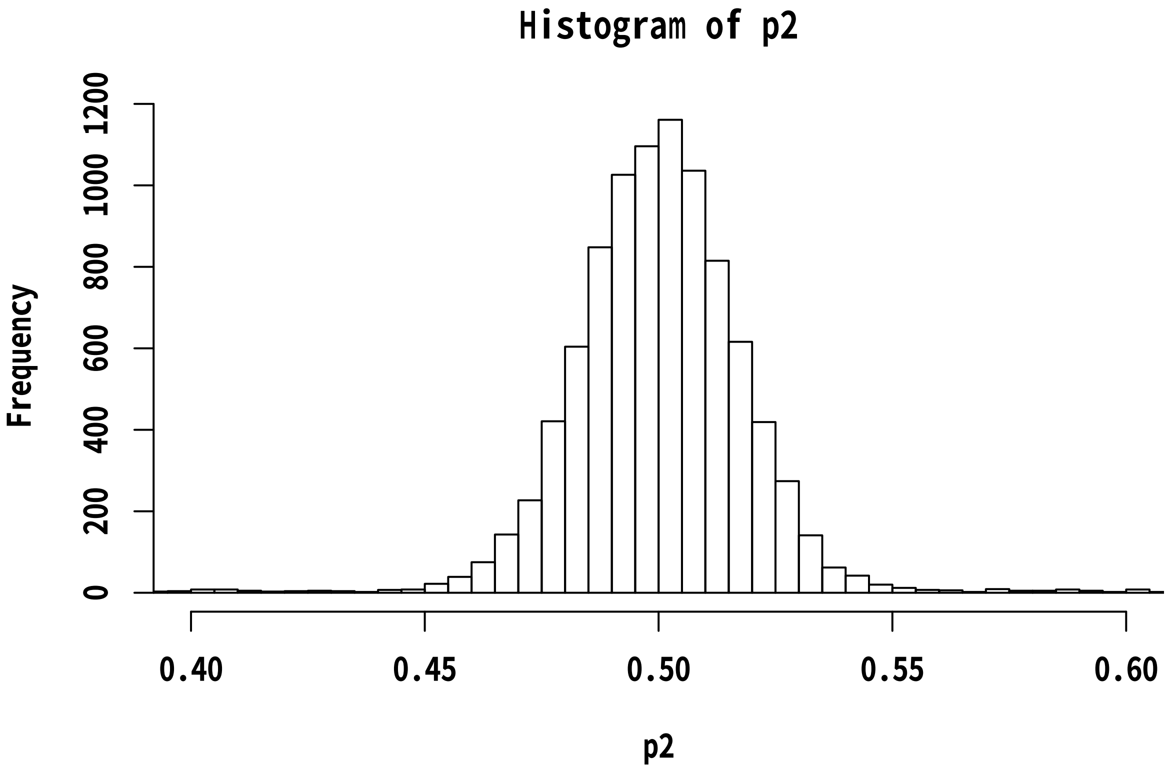
```
## [1] 0.914
```

■P2

```r
q = function (p) 0.9 * dbeta(p, 500, 500) + 0.1 * dbeta(p, 1, 1)
p = seq(0.4, 0.6, length = 1000)
plot(p, q(p))
```




```r
m = 10000 # 1000 では分布の形を見るのには少ない
p2_sampling = function () {
  x = sample(c(0, 1), 1, prob = c(0.1, 0.9))
  if (x == 1) {
    return(rbeta(1, 500, 500))
  } else {
    return(rbeta(1, 1, 1))
  }
}
p2 = replicate(m, p2_sampling())
hist(p2, xlim = c(0.4, 0.6), breaks=seq(0, 1, 0.005))
```




```
length(p2[0.44 < p2 & p2 < 0.56]) / m
```

```
## [1] 0.9121
```

(b)
尤度は以下のようになる.

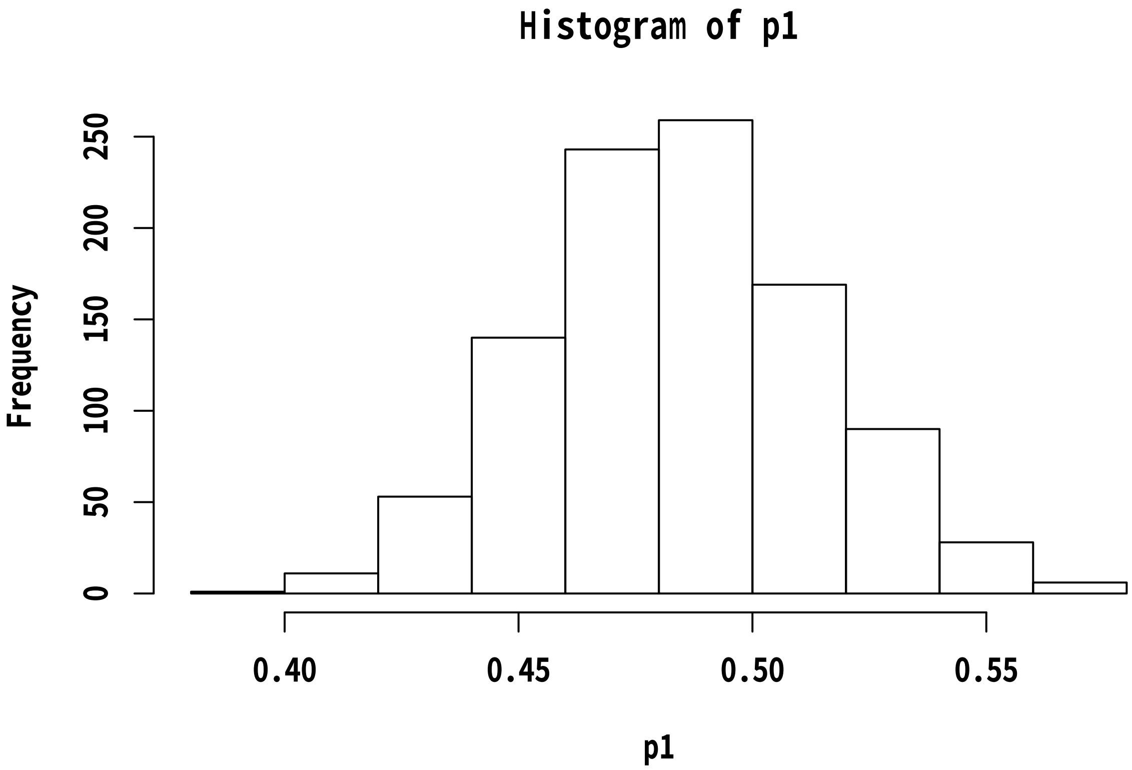
$$L(p) = \binom{100}{45} p^{45}(1-p)^{55}$$

■P1

事後分布は、事前分布が $Beta(100, 100)$ で、表が 45 回出ているので、$Beta(145, 155)$ となる.

$$\begin{aligned}
p(p|data) &\propto L(p)g(p) \\
&= \binom{100}{45} p^{45}(1-p)^{55} Beta(p|100, 100) \\
&\propto p^{45}(1-p)^{55} p^{100-1}(1-p)^{100-1} \\
&= p^{45+100-1}(1-p)^{55+100-1}
\end{aligned}$$

```r
p1 = rbeta(1000, 145, 155)
hist(p1)
```




```r
quantile(p1, c(0.05, 0.95))
```

```
##        5%       95%
## 0.4348357 0.5342714
```

■P2

事後分布は、以下のようになる.

$$
\begin{aligned}
p(p|data) &\propto L(p)g(p) \\
&= \binom{100}{45} p^{45}(1-p)^{55}\{0.9Beta(p|500,500) + 0.1Beta(p|1,1)\} \\
&= 0.9 \cdot \binom{100}{45} p^{45}(1-p)^{55} Beta(p|500,500) + 0.1 \cdot \binom{100}{45} p^{45}(1-p)^{55} Beta(p|1,1) \\
&= 0.9 \cdot \binom{100}{45} p^{45}(1-p)^{55} \frac{1}{B(500,500)} p^{500-1}(1-p)^{500-1} + 0.1 \cdot \binom{100}{45} p^{45}(1-p)^{55} \frac{1}{B(1,1)} p^{1-1}(1-p)^{1-1} \\
&\propto 0.9 \cdot \frac{1}{B(500,500)} p^{500+45-1}(1-p)^{500+55-1} + 0.1 \cdot \frac{1}{B(1,1)} p^{1+45-1}(1-p)^{1+55-1} \\
&= 0.9 \cdot \frac{1}{B(500,500)} p^{500+45-1}(1-p)^{500+55-1} + 0.1 \cdot p^{1+45-1}(1-p)^{1+55-1} \\
&= 0.9 \cdot \frac{B(545,555)}{B(500,500)} \frac{1}{B(545,555)} p^{500+45-1}(1-p)^{500+55-1} + 0.1 \cdot B(46,56) \frac{1}{B(46,56)} p^{1+45-1}(1-p)^{1+55-1} \\
&= 0.9 \cdot \frac{B(545,555)}{B(500,500)} Beta(p|545,555) + 0.1 \cdot B(46,56) Beta(p|46,56)
\end{aligned}
$$

$\int Beta(p|545,555)dp = 1, \int Beta(p|46,56)dp = 1$ より、混合比率に関しては、混合比率を γ とすると、以下の式になる.

$$
\gamma : 1-\gamma = 0.9 \cdot \frac{B(545,555)}{B(500,500)} : 0.1 \cdot B(46,56)
\begin{aligned}
(1-\gamma) \cdot (0.9 \cdot \frac{B(545,555)}{B(500,500)}) &= \gamma \cdot 0.1 \cdot B(46,56) \\
(0.9 \cdot \frac{B(545,555)}{B(500,500)}) - \gamma \cdot (0.9 \cdot \frac{B(545,555)}{B(500,500)}) &= \gamma \cdot 0.1 \cdot B(46,56) \\
\{(0.9 \cdot \frac{B(545,555)}{B(500,500)}) + 0.1 \cdot B(46,56)\}\gamma &= (0.9 \cdot \frac{B(545,555)}{B(500,500)})
\end{aligned}
\gamma = \frac{(0.}{(0.9 \cdot \frac{B(545,}{B(500,}}
$$

よって、各項の係数は以下のようになる.

```
tmp = exp(lbeta(545, 555) - lbeta(500, 500)) # overflow するので、log で計算
gamma = (0.9 * tmp) / (0.9 * tmp + 0.1 * beta(46, 56))
gamma
```

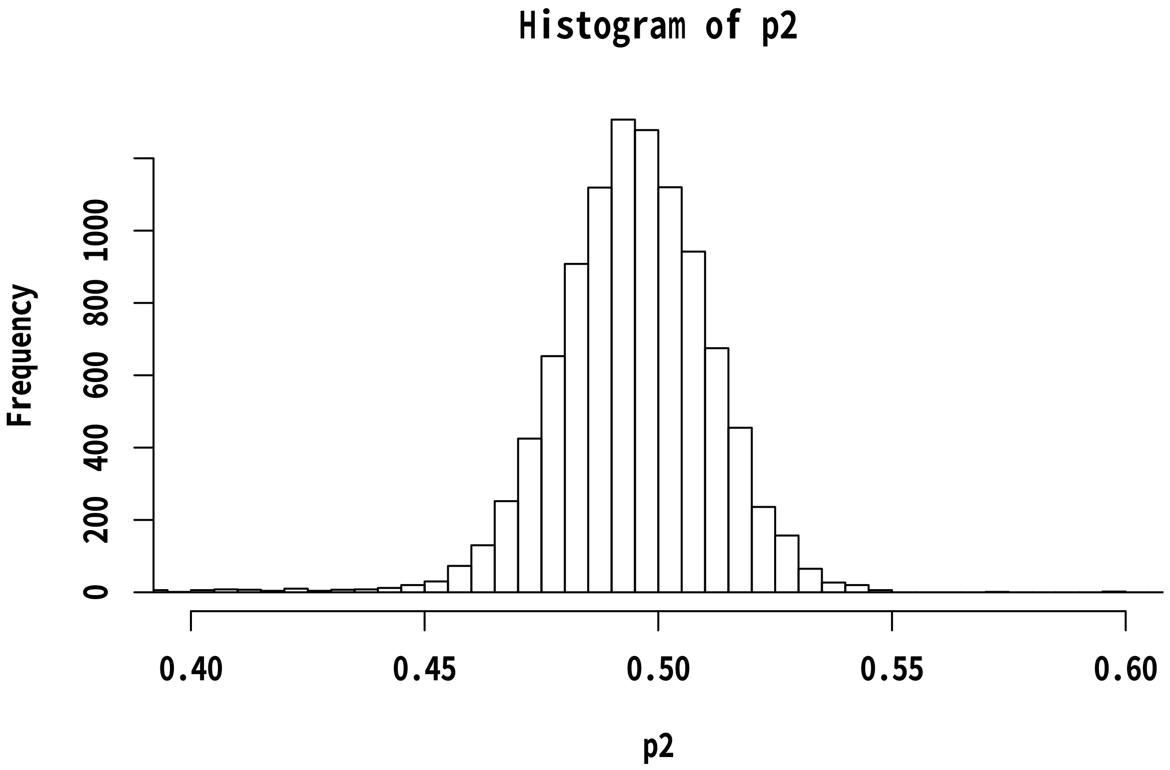
```
## [1] 0.9777615
```

```
1 - gamma
```

```
## [1] 0.02223847
```

```
m = 10000 # 1000 では分布の形を見るのには少ない
p2_sampling = function () {
  x = sample(c(0, 1), 1, prob = c(0.0222, 0.9778))
  if (x == 1) {
    return(rbeta(1, 545, 555))
```

```
  } else {
    return(rbeta(1, 46, 56))
  }
}
p2 = replicate(m, p2_sampling())
hist(p2, xlim = c(0.4, 0.6), breaks=seq(0, 1, 0.005))
```




```
quantile(p2, c(0.05, 0.95))
```

```
##        5%       95%
## 0.4683059 0.5202097
```

LearnBayes を用いると以下になる.

```
probs = c(0.9, 0.1)
beta.par1 = c(500, 500)
beta.par2 = c(1, 1)
betapar = rbind(beta.par1, beta.par2)
data = c(45, 55)
post = binomial.beta.mix(probs, betapar, data)
post
```

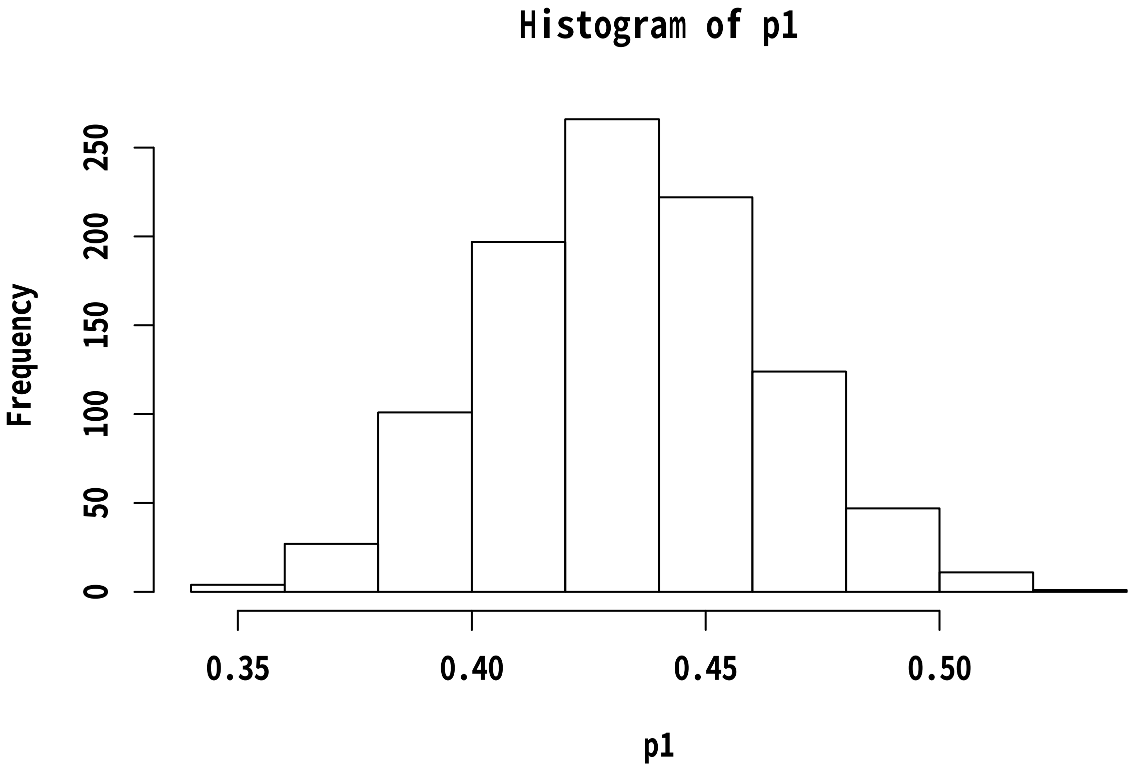
```
## $probs
```

```
##   beta.par1   beta.par2
## 0.97776153 0.02223847
##
## $betapar
##             [,1] [,2]
## beta.par1   545   555
## beta.par2    46    56
```

(c)

■P1

```r
p1 = rbeta(1000, 130, 170)
hist(p1)
```




```r
quantile(p1, c(0.05, 0.95))
```

```
##        5%       95%
## 0.3852962 0.4818855
```

■P2

同様な計算から、

$$
\begin{aligned}
p(p|data) &\propto L(p)g(p) \\
&= 0.9 \cdot \frac{B(530,570)}{B(500,500)} f_B(p;530,570) + 0.1 \cdot B(31,71) f_B(p;31,71)
\end{aligned}
$$

混合比率に関しても同様に、

$$
\gamma = \frac{(0.9 \cdot \frac{B(530,570)}{B(500,500)})}{(0.9 \cdot \frac{B(530,570)}{B(500,500)}) + 0.1 \cdot B(31,71)}
$$

よって、各項の係数は以下のようになる.

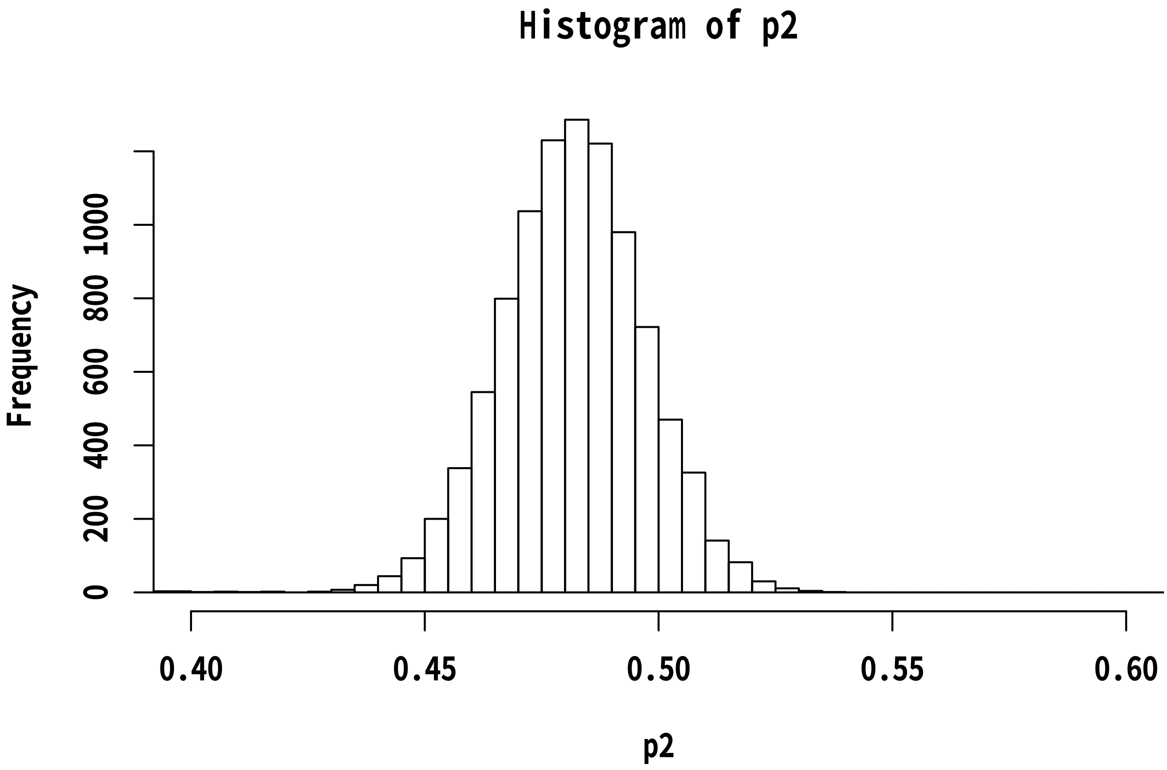
```
tmp = exp(lbeta(530, 570) - lbeta(500, 500)) # overflow するので、log で計算
gamma = (0.9 * tmp) / (0.9 * tmp + 0.1 * beta(31, 71))
gamma
```

```
## [1] 0.0399307
```

```
1 - gamma
```

```
## [1] 0.9600693
```

```
m = 10000 # 1000 では分布の形を見るのには少ない
p2_sampling = function () {
  x = sample(c(0, 1), 1, prob = c(0.0399, 0.9601))
  if (x == 1) {
    return(rbeta(1, 530, 570))
  } else {
    return(rbeta(1, 31, 71))
  }
}
p2 = replicate(m, p2_sampling())
hist(p2, xlim = c(0.4, 0.6), breaks=seq(0, 1, 0.005))
```




```r
quantile(p2, c(0.05, 0.95))
```

```
##        5%       95%
## 0.4458550 0.5062934
```

LearnBayes を用いると以下になる.

```r
probs = c(0.9, 0.1)
beta.par1 = c(500, 500)
beta.par2 = c(1, 1)
betapar = rbind(beta.par1, beta.par2)
data = c(30, 70)
post = binomial.beta.mix(probs, betapar, data)
post
```

```
## $probs
## beta.par1 beta.par2
## 0.0399307 0.9600693
##
## $betapar
##           [,1] [,2]
## beta.par1  530  570
## beta.par2   31   71
```

(d)

	45	30
P1	0.4348357 ~ 0.5342714	0.3852962 ~ 0.4818855
P2	0.4683059 ~ 0.5202097	0.4458550 ~ 0.5062934

上記の表から、P2 の方が頑健性が高い.

3-5

(a)

$$p(X = 8) = \binom{20}{8} p^8 (1 - p)^{20-8}$$

```
dbinom(8, 20, 0.2)
```

```
## [1] 0.02216088
```

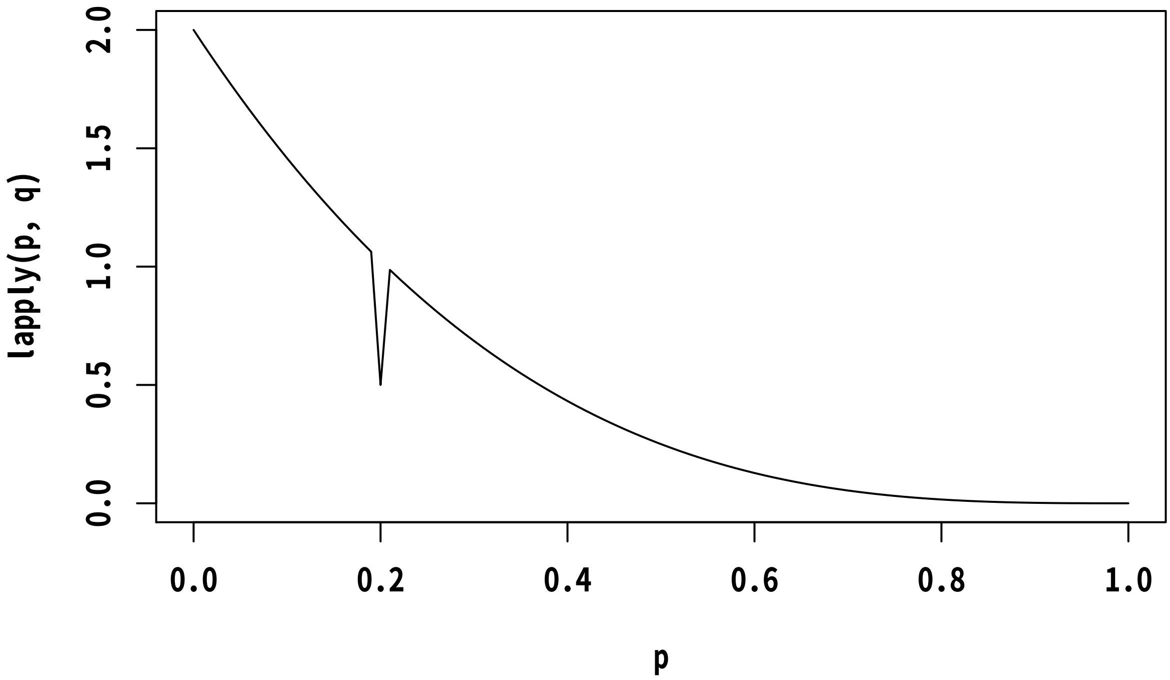
```
choose(20, 8) * (0.2) ^ 8 * (1 - 0.2) ^ (20 - 8)
```

```
## [1] 0.02216088
```

(b)

$$g(p) = 0.5I(p = 0.2) + 0.5I(p \neq 0.2)Beta(p|1, 4)$$

```
q = function (p) {
  if (p == 0.2) {
    return(0.5)
  } else {
    return(0.5 * dbeta(p, 1, 4))
  }
}
p = seq(0, 1, by = 0.01)
plot(p, lapply(p, q), type = "l")
```



$$
\begin{aligned}
p(p|data) &\propto L(p)g(p) \\
&= \binom{20}{8} p^8(1-p)^{20-8}\{0.5I(p=0.2)+0.5I(p\neq 0.2)Beta(p|1,4)\} \\
&= \binom{20}{8} p^8(1-p)^{20-8}\frac{1}{2}I(p=0.2)+\binom{20}{8} p^8(1-p)^{20-8}\frac{1}{2}I(p\neq 0.2)Beta(p|1,4) \\
&\propto p^8(1-p)^{20-8}I(p=0.2)+p^8(1-p)^{20-8}I(p\neq 0.2)Beta(p|1,4) \\
&= p^8(1-p)^{20-8}I(p=0.2)+p^8(1-p)^{20-8}I(p\neq 0.2)\frac{1}{B(1,4)}p^{1-1}(1-p)^{4-1} \\
&= p^8(1-p)^{20-8}I(p=0.2)+I(p\neq 0.2)\frac{1}{B(1,4)}p^{9-1}(1-p)^{16-1} \\
&= p^8(1-p)^{20-8}I(p=0.2)+I(p\neq 0.2)\frac{B(9,16)}{B(1,4)}Beta(p|9,16)
\end{aligned}
$$

$\int Beta(p|9,16)=1$ と $p^8(1-p)^{20-8}I(p=0.2)=(0.2)^8(1-0.2)^{12}$ より、係数に関しては以下のようになる.

$$
\gamma : 1-\gamma = (0.2)^8(1-0.2)^{12} : \frac{B(9,16)}{B(1,4)} \quad
\begin{aligned}
(1-\gamma)\cdot(0.2)^8(1-0.2)^{12} &= \gamma\cdot\frac{B(9,16)}{B(1,4)} \\
(0.2)^8(1-0.2)^{12}-\gamma\cdot(0.2)^8(1-0.2)^{12} &= \gamma\cdot\frac{B(9,16)}{B(1,4)}
\end{aligned}
\quad \gamma = \frac{(0.2)^8(1-0.2)^{12}}{(0.2)^8(1-0.2)^{12}+\frac{B(9,16)}{B(1,4)}}
$$

```r
gamma = (0.2 ^ 8 * (1 - 0.2)^12) / (0.2 ^ 8 * (1 - 0.2)^12 + beta(9, 16)/beta(1, 4))
gamma
```

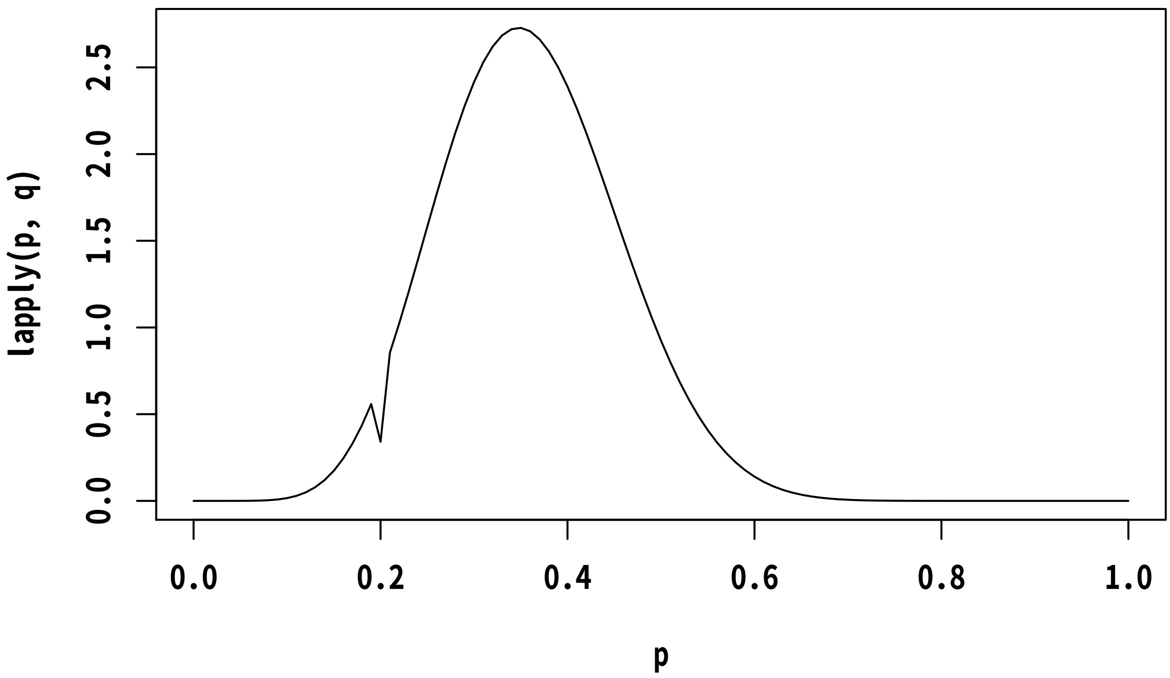
```
## [1] 0.3410395
```

```r
1 - gamma
```

```
## [1] 0.6589605
```

よって、今回の結果だと、$p = 0.2$ のときは、0.3410 となる．一方で、(a) の結果は、0.02216088 である．

```r
q = function (p) {
  if (p == 0.2) {
    return(0.341)
  } else {
    return(0.659 * dbeta(p, 9, 16))
  }
}
p = seq(0, 1, by = 0.01)
plot(p, lapply(p, q), type = "l")
```



LearnBayes を用いると、以下になる．

```r
pbetat(0.2, .5, c(1, 4), c(8, 12))
```

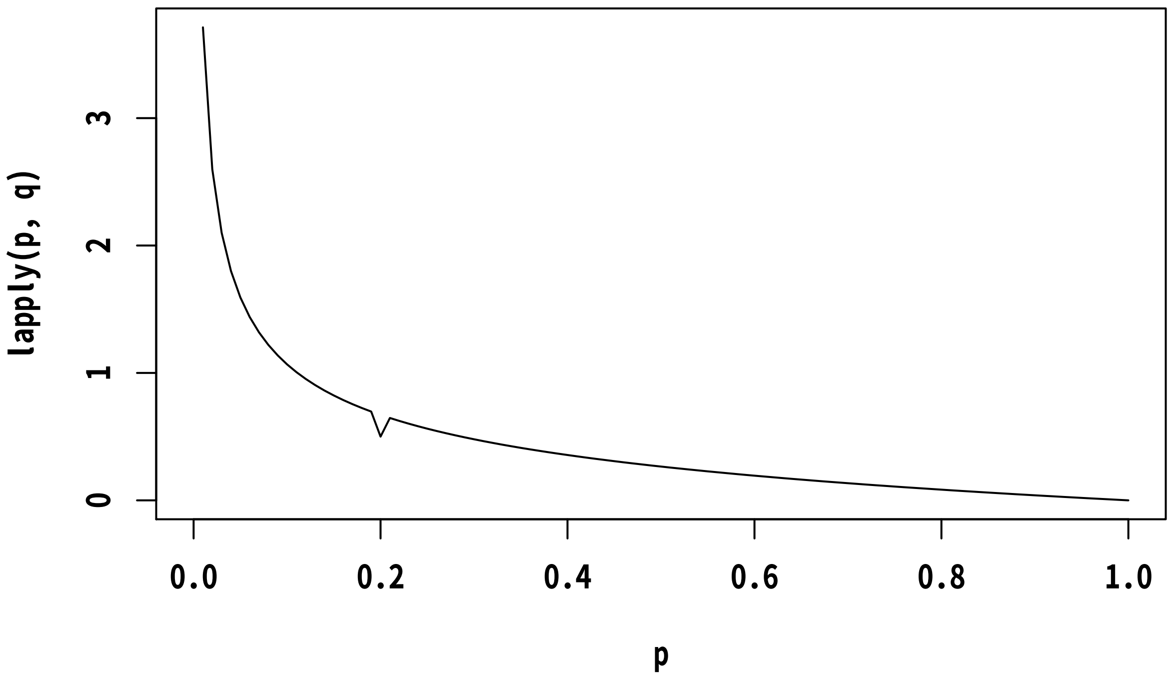
```
## $bf
## [1] 0.5175417
##
## $post
## [1] 0.3410395
```

(c)

■(1)

$$g(p) = 0.5I(p = 0.2) + 0.5I(p \neq 0.2)Beta(p|0.5, 2)$$

```r
q = function (p) {
  if (p == 0.2) {
    return(0.5)
  } else {
    return(0.5 * dbeta(p, 0.5, 2))
  }
}
p = seq(0, 1, by = 0.01)
plot(p, lapply(p, q), type = "l")
```



$$\begin{aligned} p(p|data) &\propto L(p)g(p) \\ &= p^8(1-p)^{20-8}I(p = 0.2) + I(p \neq 0.2)\frac{B(8.5, 14)}{B(0.5, 2)}Beta(p|8.5, 14) \end{aligned}$$

$\int Beta(p|8.5, 14) = 1$ と $p^8(1-p)^{20-8}I(p = 0.2) = (0.2)^8(1-0.2)^{12}$ より、係数に関しては以下のようになる.

$$\gamma : 1-\gamma = (0.2)^8(1-0.2)^{12} : \frac{B(8.5,14)}{B(0.5,2)} \gamma = \frac{(0.2)^8(1-0.2)^{12}}{(0.2)^8(1-0.2)^{12} + \frac{B(8.5,14)}{B(0.5,2)}}$$

```r
gamma = (0.2 ^ 8 * (1 - 0.2)^12) / (0.2 ^ 8 * (1 - 0.2)^12 + beta(8.5, 14)/beta(0.5, 2))
gamma
```

```
## [1] 0.3900752
```

```r
1 - gamma
```

```
## [1] 0.6099248
```

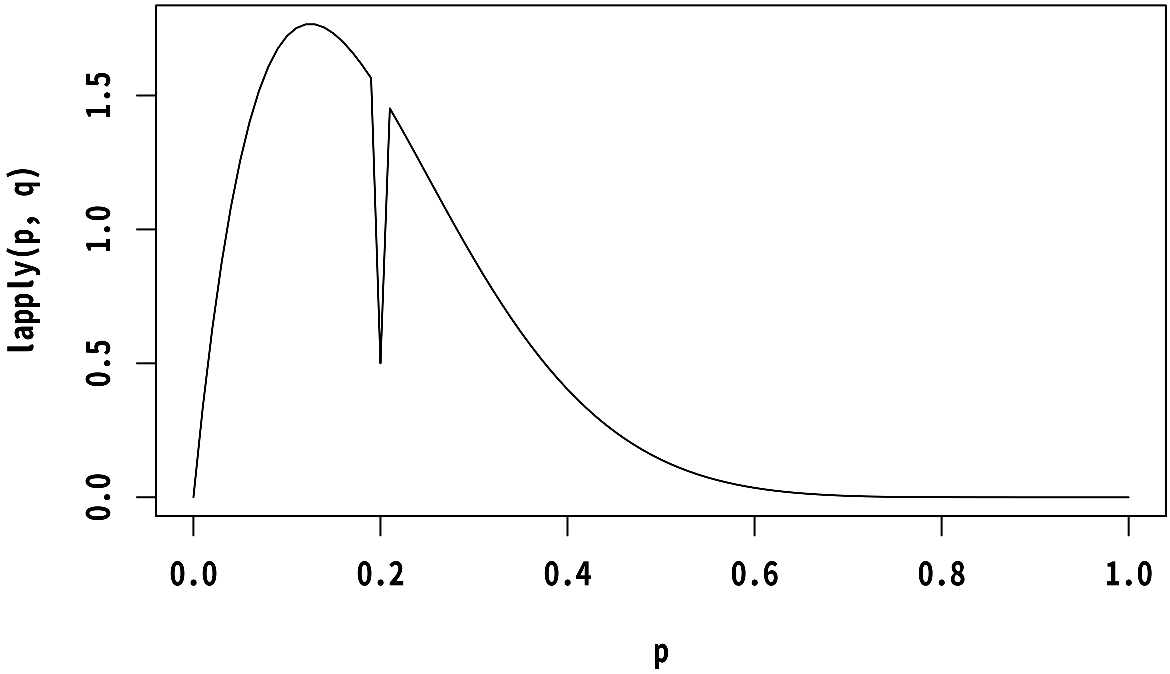
```r
pbetat(0.2, .5, c(0.5, 2), c(8, 12))
```

```
## $bf
## [1] 0.6395464
##
## $post
## [1] 0.3900752
```

■(2)

$$g(p) = 0.5I(p = 0.2) + 0.5I(p \neq 0.2)Beta(p|2,8)$$

```r
q = function (p) {
  if (p == 0.2) {
    return(0.5)
  } else {
    return(0.5 * dbeta(p, 2, 8))
  }
}
p = seq(0, 1, by = 0.01)
plot(p, lapply(p, q), type = "l")
```



$$
\begin{aligned}
p(p|data) &\propto L(p)g(p) \\
&= p^8(1-p)^{20-8}I(p=0.2) + I(p \neq 0.2)\frac{B(10,20)}{B(2,8)}Beta(p|10,20)
\end{aligned}
$$

$\int Beta(p|8.5,14) = 1$ と $p^8(1-p)^{20-8}I(p=0.2) = (0.2)^8(1-0.2)^{12}$ より、係数に関しては以下のようになる.

$$
\gamma : 1-\gamma = (0.2)^8(1-0.2)^{12} : \frac{B(10,20)}{B(2,8)}\gamma = \frac{(0.2)^8(1-0.2)^{12}}{(0.2)^8(1-0.2)^{12} + \frac{B(10,20)}{B(2,8)}}
$$

```
gamma = (0.2 ^ 8 * (1 - 0.2)^12) / (0.2 ^ 8 * (1 - 0.2)^12 + beta(10, 20)/beta(2, 8))
gamma
```

```
## [1] 0.328591
```

```
1 - gamma
```

```
## [1] 0.671409
```

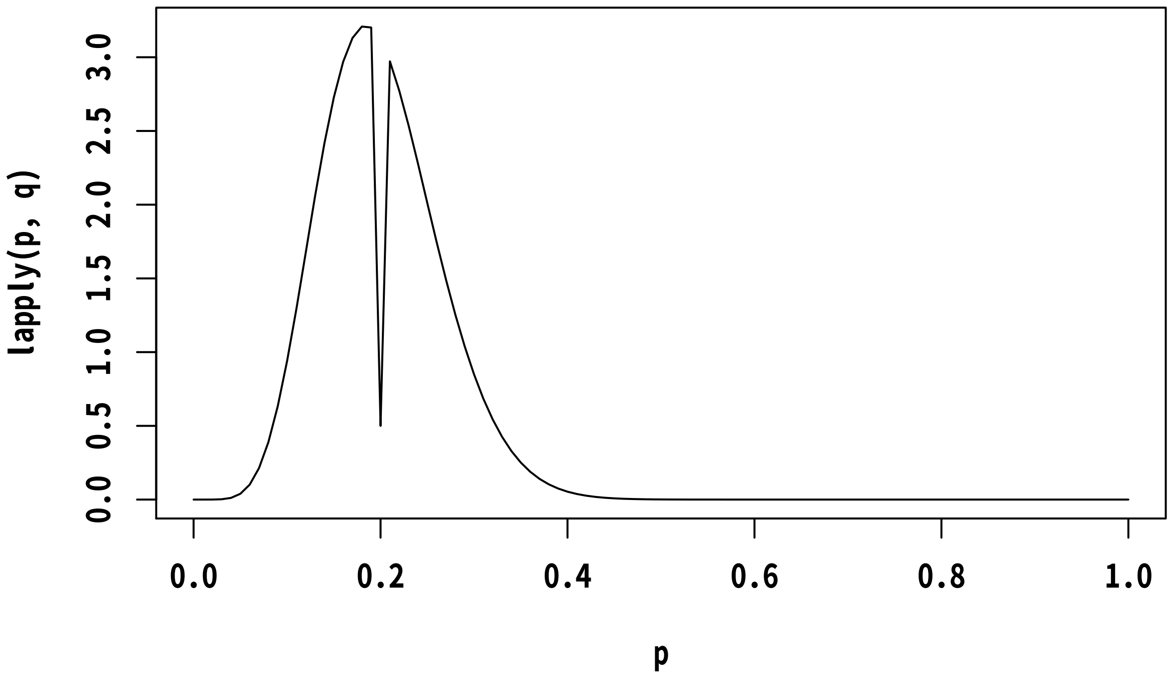
```
pbetat(0.2, .5, c(2, 8), c(8, 12))
```

```
## $bf
## [1] 0.4894051
##
## $post
## [1] 0.328591
```

■(3)

$$g(p) = 0.5I(p = 0.2) + 0.5I(p \neq 0.2)Beta(p|8, 32)$$

```
q = function (p) {
  if (p == 0.2) {
    return(0.5)
  } else {
    return(0.5 * dbeta(p, 8, 32))
  }
}
p = seq(0, 1, by = 0.01)
plot(p, lapply(p, q), type = "l")
```



$$\begin{aligned} p(p|data) &\propto L(p)g(p) \\ &= p^8(1-p)^{20-8}I(p=0.2) + I(p \neq 0.2)\frac{B(16,44)}{B(8,32)}Beta(p|16,44) \end{aligned}$$

$\int Beta(p|16, 44) = 1$ と $p^8(1-p)^{20-8}I(p = 0.2) = (0.2)^8(1-0.2)^{12}$ より、係数に関しては以下のようになる.

$$\gamma : 1-\gamma = (0.2)^8(1-0.2)^{12} : \frac{B(16,44)}{B(8,32)}\gamma = \frac{(0.2)^8(1-0.2)^{12}}{(0.2)^8(1-0.2)^{12}+\frac{B(16,44)}{B(8,32)}}$$

```
gamma = (0.2 ^ 8 * (1 - 0.2)^12) / (0.2 ^ 8 * (1 - 0.2)^12 + beta(16, 44)/beta(8, 32))
gamma
```

```
## [1] 0.3855337
```

```
1 - gamma
```

```
## [1] 0.6144663
```

```
pbetat(0.2, .5, c(8, 32), c(8, 12))
```

```
## $bf
## [1] 0.6274287
##
## $post
## [1] 0.3855337
```

(d)
20 回中 8 回当てる確率は 0.3 程度と考えられるので、ESP があるとは言えない.

3-6

速度平均 μ、標準偏差 $\sigma = 10$ で正規分布であるので、70 マイルで追い越す確率は $P(\mu < 70) = \Phi(70, \mu, 10)$ である. よって、尤度は以下のように表せる.

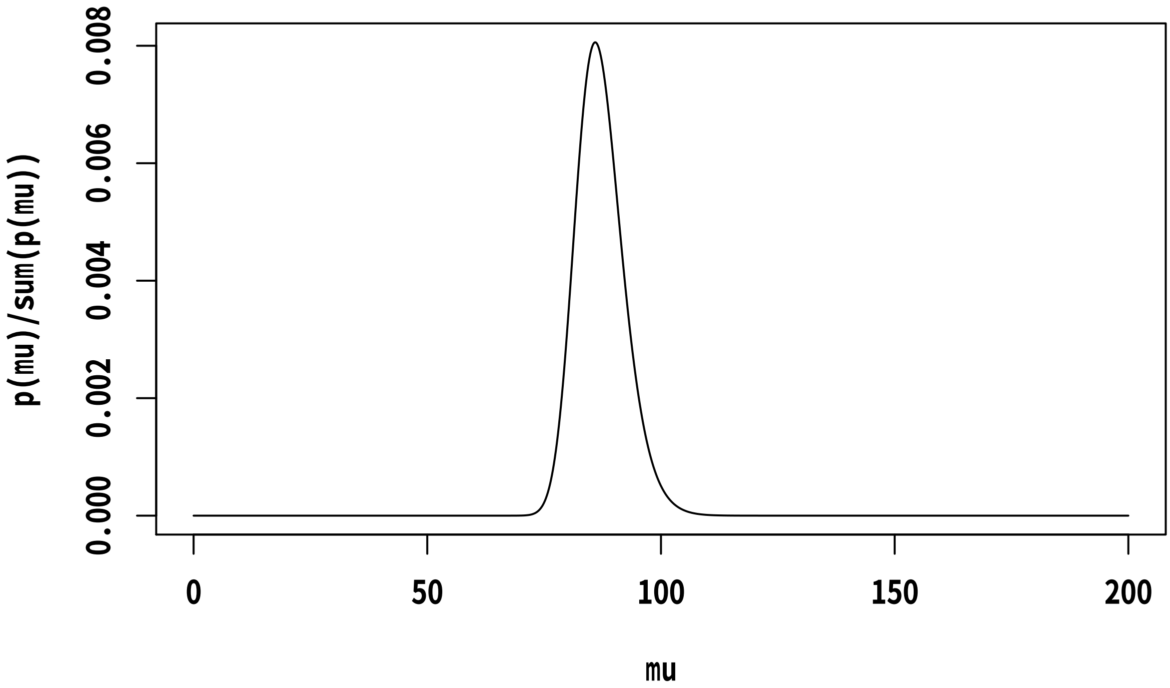
$$L(\mu) \propto \Phi(70,\mu,10)^s(1-\Phi(70,\mu,10))^f$$

(a)
事前分布は一様分布 $(f(x) = C \quad (-\infty < x < \infty))$ であるとすると、

$$\begin{aligned} p(\mu|data) &\propto L(\mu)q(\mu) \\ &= C\cdot\Phi(70,\mu,10)(1-\Phi(70,\mu,10))^{17} \\ &\propto \Phi(70,\mu,10)(1-\Phi(70,\mu,10))^{17} \end{aligned}$$

```
# グリッド近似を用いる
mu = seq(0, 200, by = 0.1) # 速度は 0 以上であるので、加えて、最大値は 200 とした.
p = function (mu) {
```

```r
  pnorm(70, mu, 10) * (1 - pnorm(70, mu, 10))^17
}
plot(mu, p(mu) / sum(p(mu)), type = "l")
```



(b)
グリッド近似した際の、各グリッドの値を μ_i、グリッド数を N とすると、事後平均は以下のように表せる.

$$Mean = \sum_{i=1}^{N} \mu_i \cdot p(\mu_i)$$

```r
# グリッド近似を用いる
mu = seq(0, 200, by = 0.1) # 速度は 0 以上であるので、加えて、最大値は 200 とした.
p = function (mu) {
  pnorm(70, mu, 10) * (1 - pnorm(70, mu, 10))^17
}
post = p(mu) / sum(p(mu))
sum(mu * post) # 事後平均
```

```
## [1] 87.11109
```

(c)
$P(\mu > 80)$ を求めればよいので、以下のようになる.

```r
mu = seq(0, 150, by = 0.1) # 速度は 0 以上であるので、加えて、最大値は 200 とした.
p = function (mu) {
  pnorm(70, mu, 10) * (1 - pnorm(70, mu, 10))^17
}
post = p(mu) / sum(p(mu))
sum(cbind(mu, post)[mu > 80, 2])
```

```
## [1] 0.9300158
```

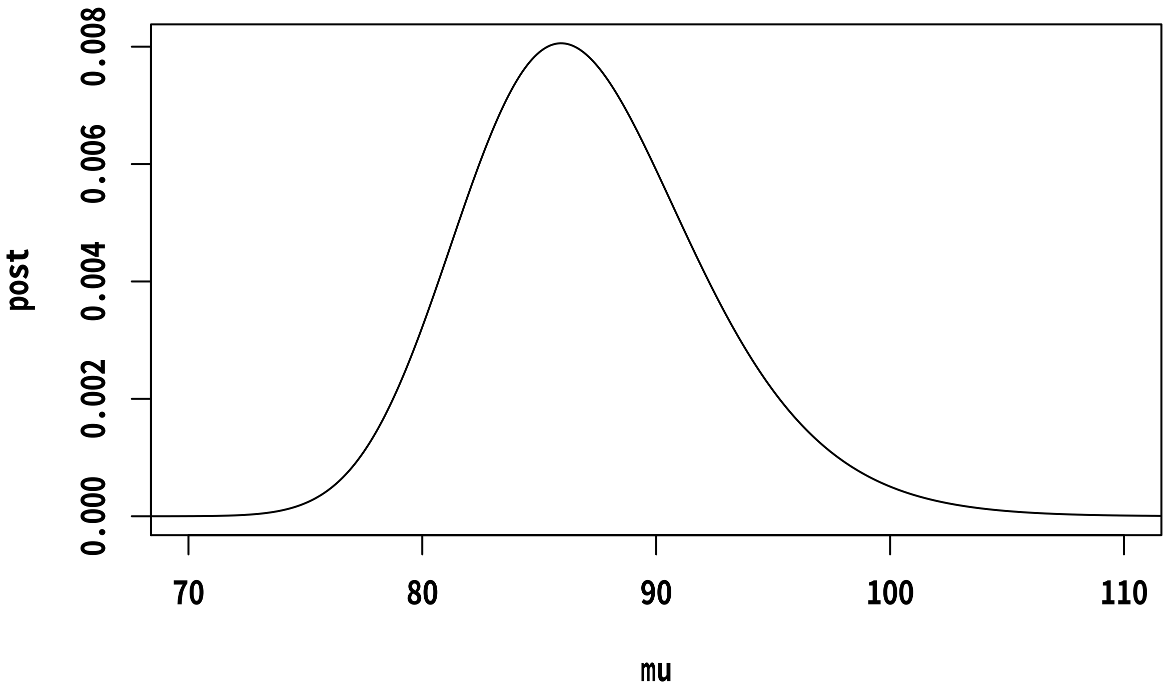
$P(\mu > 80) = 1 - P(\mu \leq 80) = 1 - \int_{-\infty}^{80} p(\mu) d\mu$ の数値積分は以下のようになる.

```r
p = function (mu) {
  pnorm(70, mu, 10) * (1 - pnorm(70, mu, 10))^17
}
z = integrate(p, 0, 150)$value # 正規化定数
int = integrate(p, -Inf, 80)
1 - int$value / z
```

```
## [1] 0.9316374
```

80 マイル近傍の事後分布は以下のようになる.

```r
mu = seq(0, 200, by = 0.1)
p = function (mu) {
  pnorm(70, mu, 10) * (1 - pnorm(70, mu, 10))^17
}
post = p(mu) / sum(p(mu))
plot(mu, post, xlim = c(70, 110), type = "l")
```



3-7

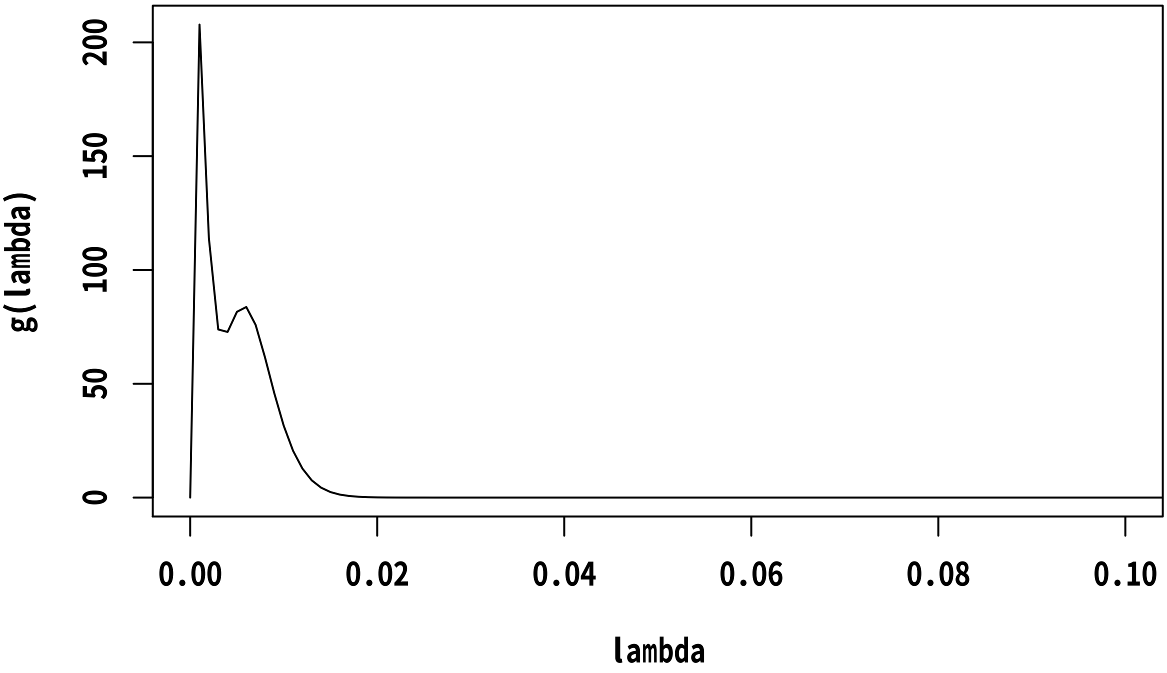
(a)

$$g(\lambda) = 0.5 \cdot gamma(\lambda|1.5, 1000) + 0.5 \cdot gamma(\lambda|7, 1000)$$

ガンマ分布は以下で定義する．

$$gamma(\lambda|\alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)}\lambda^{\alpha-1}\exp(-\beta\lambda) \quad (\lambda > 0)$$

```
lambda = seq(0, 1, by = 0.001)
g = function (lambda) {
  0.5 * dgamma(lambda, shape = 1.5, rate = 1000) + 0.5 * dgamma(lambda, shape = 7, rate = 1000)
}
plot(lambda, g(lambda), xlim = c(0, 0.1), type = "l")
```



(b)

死亡数 y は、暴露数 e と死亡率 λ としたとき、平均 $e\lambda$ のポアソン分布に従うと考えられる.

$$p(y) = Po(e\lambda) = \frac{(e\lambda)^y}{y!} \exp(-e\lambda)$$

よって、$y = 4, e = 1767$ の時の、尤度は以下の用に表せる.

$$L(\lambda) = \frac{(1767\lambda)^4}{4!} \exp(-1767\lambda)$$

この時、λ の事後分布は以下のようになる.

$$\begin{aligned}
p(\lambda|data) &\propto L(\lambda)g(\lambda) \\
&= \frac{(1767\lambda)^4}{4!} \exp(-1767\lambda)\{0.5 \cdot gamma(\lambda|1.5, 1000) + 0.5 \cdot gamma(\lambda|7, 1000)\} \\
&\propto \lambda^4 \exp(-1767\lambda)\{gamma(\lambda|1.5, 1000) + gamma(\lambda|7, 1000)\} \\
&= \lambda^4 \exp(-1767\lambda)\{\frac{1000^{1.5}}{\Gamma(1.5)}\lambda^{1.5-1} \exp(-1000\lambda) + \frac{1000^7}{\Gamma(7)}\lambda^{7-1} \exp(-1000\lambda)\} \\
&= \frac{1000^{1.5}}{\Gamma(1.5)}\lambda^{1.5+4-1} \exp(-2767\lambda) + \frac{1000^7}{\Gamma(7)}\lambda^{7+4-1} \exp(-2767\lambda) \\
&= \frac{1000^{1.5}}{\Gamma(1.5)} \frac{\Gamma(5.5)}{2767^{5.5}} gamma(\lambda|5.5, 2767) + \frac{1000^7}{\Gamma(7)} \frac{\Gamma(11)}{2767^{11}} gamma(\lambda|11, 2767)
\end{aligned}$$

ここで、混合比率を π とすると、

$$\pi : 1-\pi = \frac{1000^{1.5}}{\Gamma(1.5)}\frac{\Gamma(5.5)}{2767^{5.5}} : \frac{1000^{7}}{\Gamma(7)}\frac{\Gamma(11)}{2767^{11}} a = \frac{1000^{1.5}}{\Gamma(1.5)}\frac{\Gamma(5.5)}{2767^{5.5}}, b = \frac{1000^{7}}{\Gamma(7)}\frac{\Gamma(11)}{2767^{11}} \pi = \frac{a}{a+b}$$

```
a = (1000 ^ 1.5 * gamma(5.5)) / (gamma(1.5) * 2767 ^ 5.5)
b = (1000 ^ 7 * gamma(11)) / (gamma(7) * 2767 ^ 11)
pi = a / (a + b)
pi
```

```
## [1] 0.7597182
```

```
1 - pi
```

```
## [1] 0.2402818
```

LearnBayes の場合は以下のようになる.

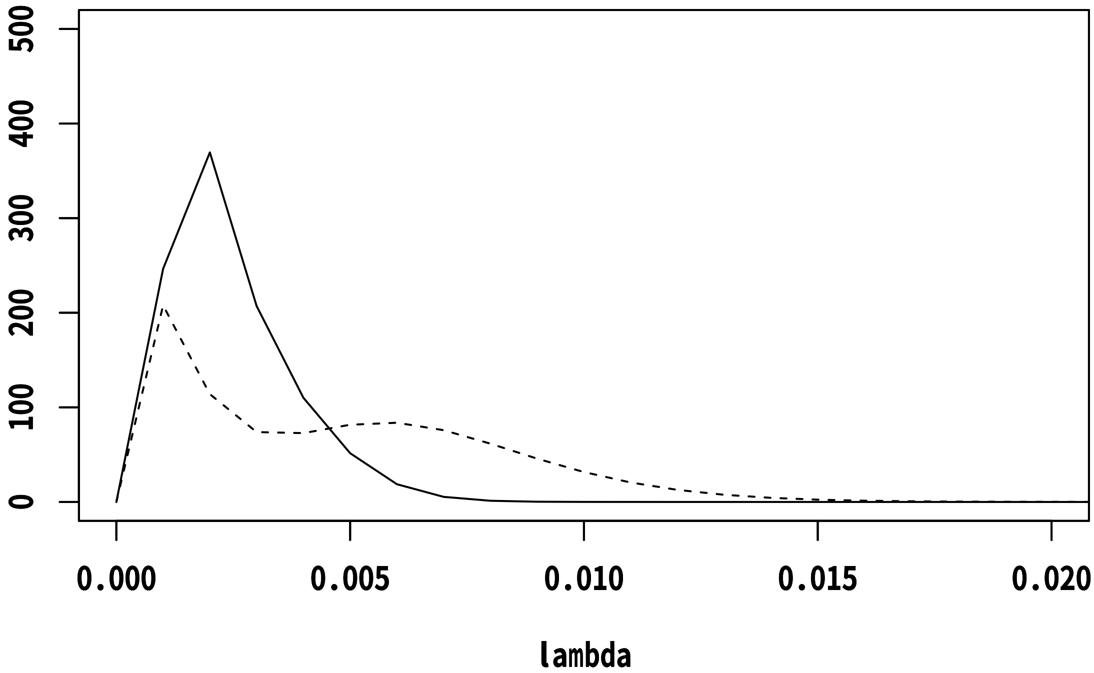
```
probs = c(0.5, 0.5)
gamma.par1 = c(1.5, 1000)
gamma.par2 = c(7, 1000)
gammapar = rbind(gamma.par1, gamma.par2)
data = data.frame(t = 1767, y = 4)
post = poisson.gamma.mix(probs, gammapar, data)
post
```

```
## $probs
## gamma.par1 gamma.par2
##  0.7597182  0.2402818
##
## $gammapar
##            [,1] [,2]
## gamma.par1  5.5 2767
## gamma.par2 11.0 2767
```

(c)

```
lambda = seq(0, 1, by = 0.001)
g = function (lambda) {
  0.5 * dgamma(lambda, shape = 1.5, rate = 1000) + 0.5 * dgamma(lambda, shape = 7, rate = 1000)
}
post = function (lambda) {
  pi * dgamma(lambda, shape = 5.5, rate = 2767) + (1 - pi) * dgamma(lambda, shape = 11, rate = 2767)
}
```

```
plot(lambda, g(lambda), xlim = c(0, 0.02), ylim = c(0, 500), ylab = "", type = "l", lty = 2)
par(new=T)
plot(lambda, post(lambda), xlim = c(0, 0.02), ylim = c(0, 500), ylab = "", type = "l")
```



(d)

$P(\lambda > 0.005) = 1 - P(\lambda \leq 0.005) = 1 - \int_0^{0.005} p(\lambda) d\lambda$ である.

```
# 数値積分
p = function (lambda) {
  return(pi * dgamma(lambda, shape = 5.5, rate = 2767) + (1 - pi) * dgamma(lambda, shape = 11, rate
}
int = integrate(p, lower = 0, upper = 0.005)
1 - int$value
```

```
## [1] 0.04766545
```

```
# サンプリング近似
post_sample = function () {
  x = sample(c(0, 1), 1, prob = c(1 - pi, pi))
  if (x == 1) {
    return(rgamma(1, shape = 5.5, rate = 2767))
  } else {
    return(rgamma(1, shape = 11, rate = 2767))
  }
```

```
}
sample = replicate(100000, post_sample())
sum(sample > 0.005) / 100000
```

```
## [1] 0.04825
```

```
# グリッド近似
lambda = seq(0, 1, by = 0.000001)
p = function (lambda) {
  pi * dgamma(lambda, shape = 5.5, rate = 2767) + (1 - pi) * dgamma(lambda, shape = 11, rate = 2767)
}
post = p(lambda) / sum(p(lambda))
sum(cbind(lambda, post)[lambda > 0.005, 2])
```

```
## [1] 0.04763969
```

(e)

混合確率は、$g_1(\lambda), g_2(\lambda)$ のそれぞれ 0.7597182、0.2402818 となっているので、このデータは $g_1(\lambda)$ に適合していると考えられる.

3-8

$$f(y:\lambda) = Exp(y|\lambda) F(y;\lambda) = \int_{-\infty}^{y} f(y:\lambda)dy$$

であるとする.

平均 λ の指数分布にしたがう電球 12 個のテストする. 4 番目に短い寿命である $y_4 = 100$ であるので、1 ~ 3 番目までは 100 時間までに切れる確率であるので、$F(100;\lambda)^3$ とできる. $y_4 = 100$ となる確率 (密度) は $f(100;\lambda)$ である. 8 番目に短い寿命である $y_8 = 300$ であるので、5 ~ 7 番目までは 100 ~ 300 時間までに切れる確率であるので、$(F(300;\lambda) - F(100;\lambda))^3$ とできる. $y_8 = 300$ となる確率 (密度) は $f(300;\lambda)$ である. 9 ~ 12 番目は 300 ~ 時間で切れる確率であるので、$(1 - F(300;\lambda))^4$ であるので、尤度関数は以下になる.

$$L(\lambda) \propto F(100;\lambda)^3 f(100;\lambda)(F(300;\lambda) - F(100;\lambda))^3 f(100;\lambda)(1 - F(300;\lambda))^4$$

(a)

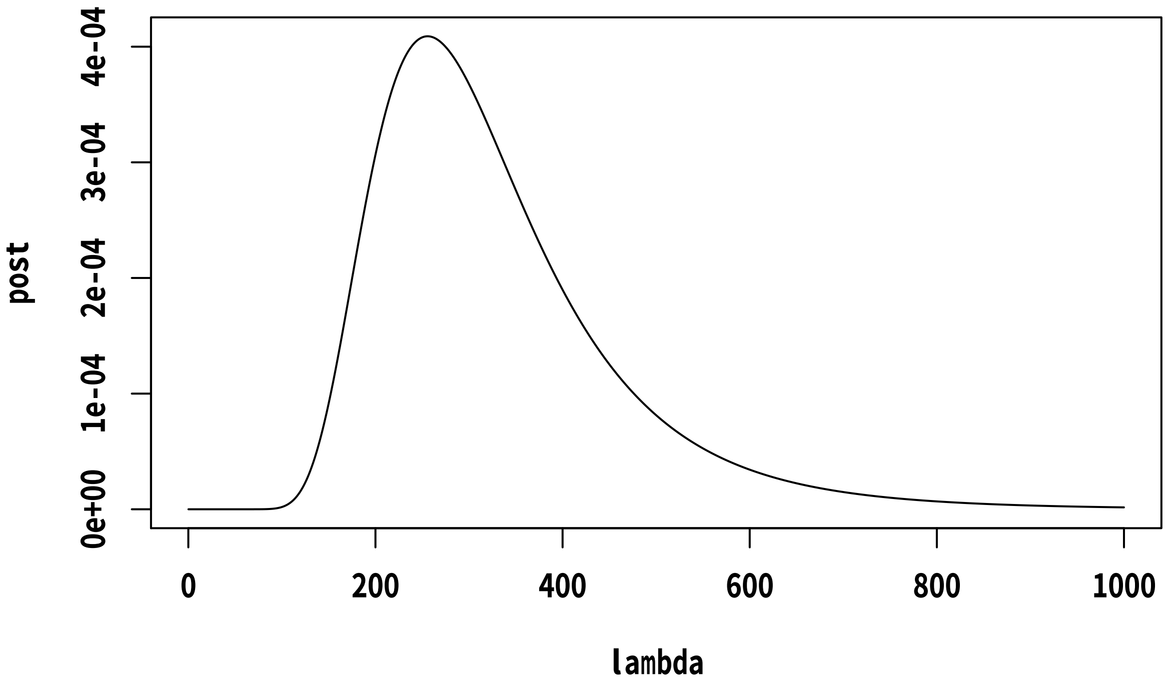
事前分布は以下とする.

$$p(\lambda) \propto \frac{1}{\lambda}$$

事後分布は以下になる.

$$
\begin{aligned}
p(\lambda|data) &\propto L(\lambda)p(\lambda) \\
&= F(100;\lambda)^3 f(100;\lambda)(F(300;\lambda) - F(100;\lambda))^3 f(100;\lambda)(1 - F(300;\lambda))^4 \frac{1}{\lambda}
\end{aligned}
$$

```
# グリッド近似を用いる
lambda = seq(0.1, 1000, by = 0.1)
p = function (lambda) {
  likelihood = pexp(100, 1/lambda)^3 * dexp(100, 1/lambda) * (pexp(300, 1/lambda) - pexp(100, 1/lamb
  return(likelihood / lambda)
}
post = p(lambda)
post = post / sum(post)
plot(lambda, post, type = "l")
```



(b)

```
# グリッド近似を用いる
lambda = seq(0.1, 1000, by = 0.1)
p = function (lambda) {
  likelihood = pexp(100, 1/lambda)^3 * dexp(100, 1/lambda) * (pexp(300, 1/lambda) - pexp(100, 1/lamb
  return(likelihood / lambda)
}
```

```r
post = p(lambda)
post = post / sum(post)
mu = sum(lambda * post)
mu
```

```
## [1] 327.2188
```

```r
sqrt(sum((lambda - mu)^2 * post))
```

```
## [1] 127.6595
```

(c)

```r
# グリッド近似を用いる
lambda = seq(0.1, 1000, by = 0.1)
p = function (lambda) {
  likelihood = pexp(100, 1/lambda)^3 * dexp(100, 1/lambda) * (pexp(300, 1/lambda) - pexp(100, 1/lamb
  return(likelihood / lambda)
}
post = p(lambda)
post = post / sum(post)
sum(cbind(lambda, post)[300 < lambda & lambda < 500, 2])
```

```
## [1] 0.4059514
```

4-1