A/B Testing

Nazmul

2023-10-30

Bayesian A/B testing offers significant advantages over frequentist approaches. In frequentist tests, point estimates are used, leading to interpretation challenges. Statisticians must conduct power tests, communicate results to non-technical stakeholders, and often struggle with interpretability. In contrast, Bayesian A/B testing enhances interpretability by providing direct probabilities of one variant's superiority over another. It replaces p-values with these probabilities, making it easier to understand the results. Bayesian tests offer posterior distributions for parameters, allowing for various summarization methods. They are also robust against data "peeking" and maintain validity if a test is halted prematurely.

Case Study

Scenario

A fast-food chain plans to add a new item to its menu. However, they are still undecided between three possible marketing campaigns for promoting the new product. In order to determine which promotion has the greatest effect on sales, the new item is introduced at locations in several randomly selected markets. A different promotion is used at each location, and the weekly sales of the new item are recorded for the first four weeks.

Goal

Evaluate A/B testing results and decide which marketing strategy works the best.

Columns

MarketID: unique identifier for market
MarketSize: size of market area by sales
LocationID: unique identifier for store location
AgeOfStore: age of store in years
Promotion: one of three promotions that were tested
week: one of four weeks when the promotions were run
SalesInThousands: sales amount for a specific LocationID, Promotion, and week

required libries

```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```r
library(bayesAB)
```

```
##
## Attaching package: 'bayesAB'
##
## The following objects are masked from 'package:dplyr':
##
##     combine, rename
```

```r
dt <- read.csv("WA_Marketing-Campaign.csv")
```

EDA

```r
summary(dt)
```

```
##     MarketID       MarketSize          LocationID      AgeOfStore
##  Min.   : 1.000   Length:548         Min.   :  1.0   Min.   : 1.000
##  1st Qu.: 3.000   Class :character   1st Qu.:216.0   1st Qu.: 4.000
##  Median : 6.000   Mode  :character   Median :504.0   Median : 7.000
##  Mean   : 5.715                      Mean   :479.7   Mean   : 8.504
##  3rd Qu.: 8.000                      3rd Qu.:708.0   3rd Qu.:12.000
##  Max.   :10.000                      Max.   :920.0   Max.   :28.000
##    Promotion          week      SalesInThousands
##  Min.   :1.000   Min.   :1.00   Min.   :17.34
##  1st Qu.:1.000   1st Qu.:1.75   1st Qu.:42.55
##  Median :2.000   Median :2.50   Median :50.20
##  Mean   :2.029   Mean   :2.50   Mean   :53.47
##  3rd Qu.:3.000   3rd Qu.:3.25   3rd Qu.:60.48
##  Max.   :3.000   Max.   :4.00   Max.   :99.65
```

```r
glimpse(dt)
```

```
## Rows: 548
## Columns: 7
## $ MarketID         <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
## $ MarketSize       <chr> "Medium", "Medium", "Medium", "Medium", "Medium", "Me~
## $ LocationID       <int> 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3, 4, 4, 4, 4, 5, 5,~
## $ AgeOfStore       <int> 4, 4, 4, 4, 5, 5, 5, 5, 12, 12, 12, 12, 1, 1, 1, 1, 1~
## $ Promotion        <int> 3, 3, 3, 3, 2, 2, 2, 2, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2,~
## $ week             <int> 1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4, 1, 2,~
## $ SalesInThousands <dbl> 33.73, 35.67, 29.03, 39.25, 27.81, 34.67, 27.98, 27.7~
```

```r
dt$Promotion <- as.character(dt$Promotion)
```

As we can see from this chart, the promotion group 3 has the largest aggregate sales amount (36%). However, each promotion group takes roughly one third of the total sales during the promotion weeks.

```r
#grouping data by Promotion
png(filename="prom_vs_perct.png", res=500, width=3312, height=1600)
prom <- dt %>%
  group_by(Promotion) %>%
  summarize(Sales=sum(SalesInThousands))%>%
  mutate(Percentage=round(Sales/sum(Sales)*100))

prom %>%
  ggplot(aes(x= Promotion, y = Percentage, fill= Promotion)) +
  geom_bar(stat = "identity") +  scale_fill_hue(labels = c("Prom 1", "Prom 2", "Prom 3")) + geom_text(ae
theme_minimal() +
theme(axis.text.x = element_blank(),
axis.title.x = element_blank(),
panel.grid = element_blank())
```

```r
png(filename="marsize.png", res=500, width=3312, height=1600)

msize <- dt %>%
  group_by(Promotion, MarketSize) %>%
  summarize(count= n())
```

```
## `summarise()` has grouped output by 'Promotion'. You can override using the
## `.groups` argument.
```

```r
msize %>% ggplot(aes(x= Promotion, y = count, fill= MarketSize)) +
  geom_bar(stat = "identity", position="dodge") +
  theme_minimal() +
theme(panel.grid = element_blank())
```

The graph reveals that the proportion of market size for each promotion is more or less same comparing to others.

```r
#normality check
shapiro.test(dt$SalesInThousands[dt$Promotion=="1"])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  dt$SalesInThousands[dt$Promotion == "1"]
## W = 0.9153, p-value = 1.977e-08
```

```r
shapiro.test(dt$SalesInThousands[dt$Promotion=="2"])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  dt$SalesInThousands[dt$Promotion == "2"]
## W = 0.91451, p-value = 5.457e-09
```

```r
shapiro.test(dt$SalesInThousands[dt$Promotion=="3"])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  dt$SalesInThousands[dt$Promotion == "3"]
## W = 0.92077, p-value = 1.499e-08
```

```r
# mean, standard deviation
mean(dt$SalesInThousands[dt$Promotion=="1"])
```

```
## [1] 58.09901
```

```r
mean(dt$SalesInThousands[dt$Promotion=="2"])
```

```
## [1] 47.32941
```

```r
mean(dt$SalesInThousands[dt$Promotion=="3"])
```

```
## [1] 55.36447
```

```r
sd(dt$SalesInThousands[dt$Promotion=="1"])
```

```
## [1] 16.55378
```

```r
sd(dt$SalesInThousands[dt$Promotion=="2"])
```

```
## [1] 15.10895
```

```r
sd(dt$SalesInThousands[dt$Promotion=="3"])
```

```
## [1] 16.76623
```

Since the distribution of sales follow normality

```r
prom_1 <- rnorm(172, 58.09901, 16.55378)
prom_2 <- rnorm(188, 47.32941, 15.10895)
prom_3 <- rnorm(188, 55.36447, 16.76623)
```

```r
pm12 <- bayesTest(prom_1, prom_2,
          priors = c('mu' = 5, 'lambda' = 2, 'alpha' = 3, 'beta' = 1), distribution = 'normal')
pm12
```

```
## --------------------------------------------
## Distribution used: normal
## --------------------------------------------
## Using data with the following properties:
##                 A           B
```

```
## Min.    14.04177 -0.04733435
## 1st Qu. 46.52201 34.75831194
## Median  56.58673 46.20265078
## Mean    56.63313 46.08049388
## 3rd Qu. 67.51655 56.74939204
## Max.    93.10449 93.05974912
## --------------------------------------------
## Conjugate Prior Distribution: NormalInvGamma
## Conjugate Prior Parameters:
## $mu
## [1] 5
##
## $lambda
## [1] 2
##
## $alpha
## [1] 3
##
## $beta
## [1] 1
##
## --------------------------------------------
## Calculated posteriors for the following parameters:
## Mu, Sig_Sq
## --------------------------------------------
## Monte Carlo samples generated per posterior:
## [1] 1e+05
```

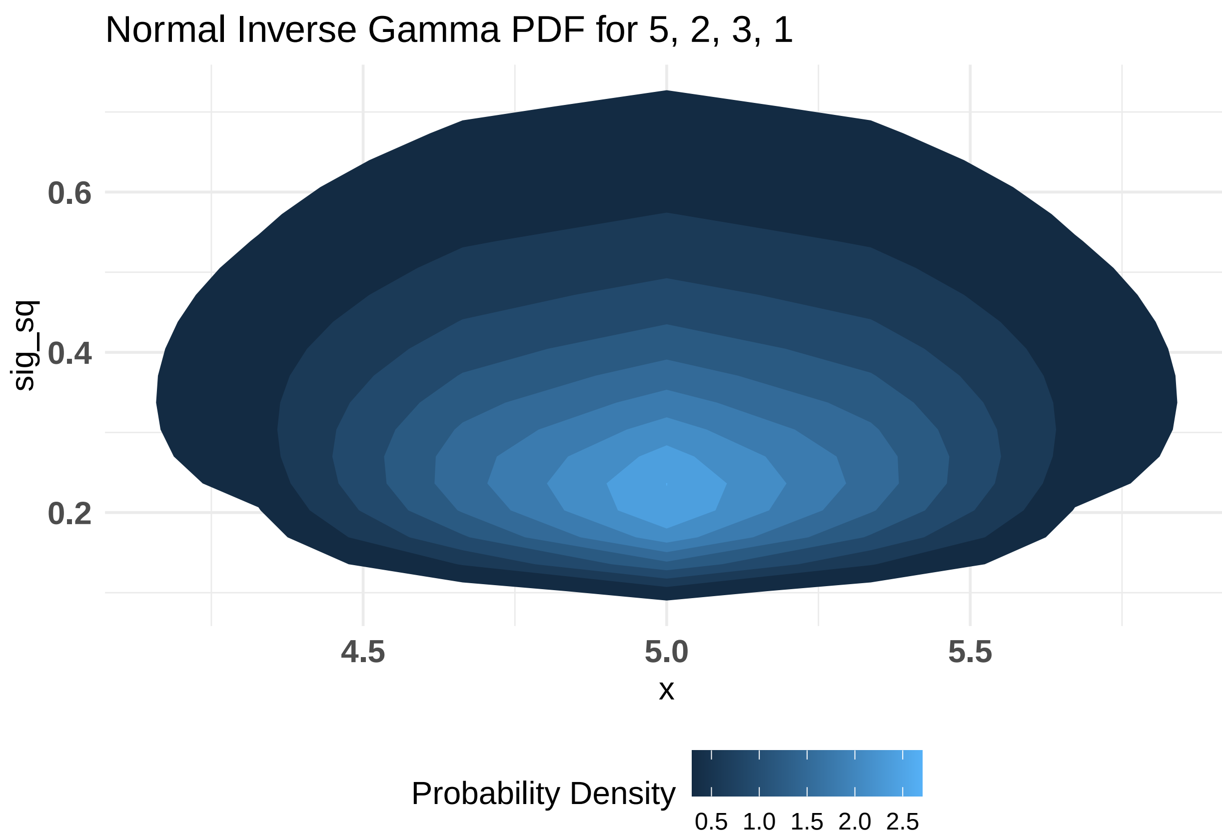
```r
summary(pm12)
```

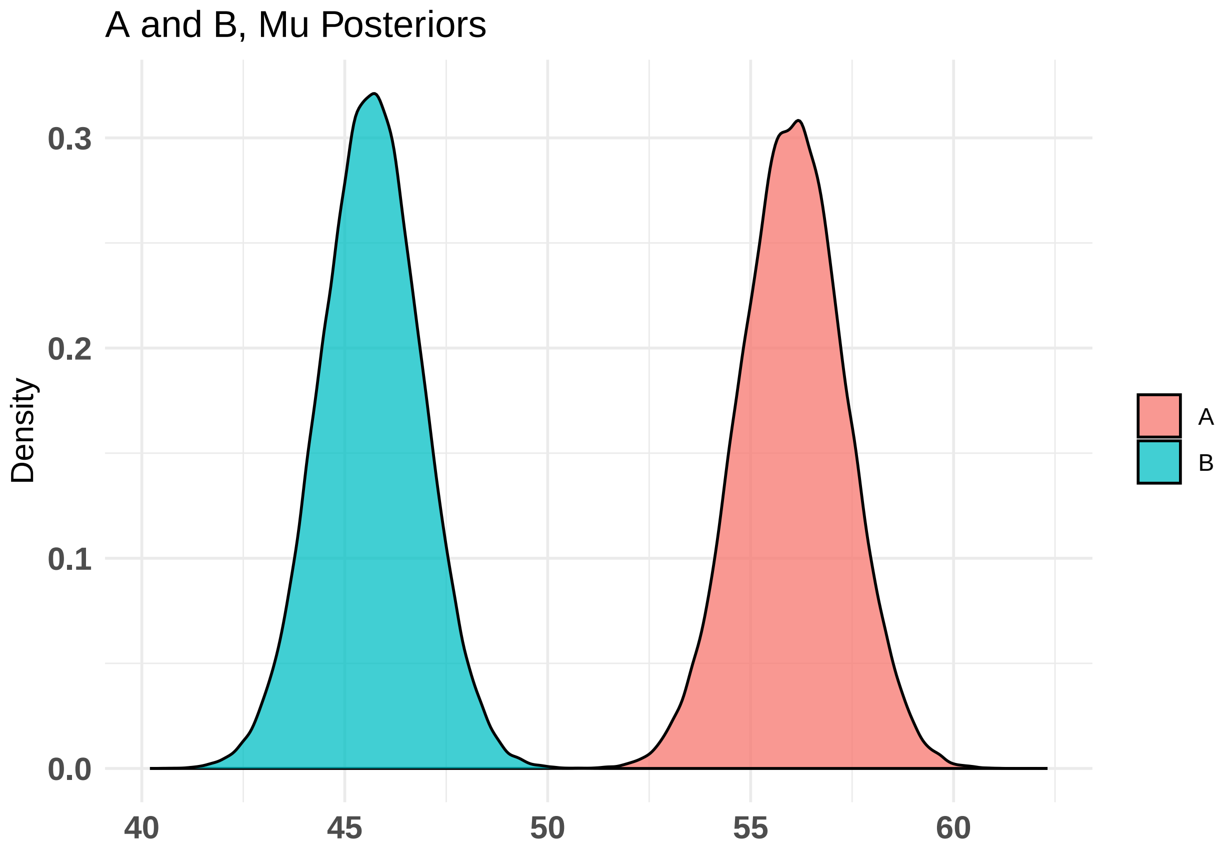
```
## Quantiles of posteriors for A and B:
##
## $Mu
## $Mu$A
##       0%      25%      50%      75%     100%
## 50.19241 55.17110 56.04095 56.89924 62.31463
##
## $Mu$B
##       0%      25%      50%      75%     100%
## 40.20034 44.81890 45.65341 46.48241 50.95095
##
##
## $Sig_Sq
## $Sig_Sq$A
##       0%      25%      50%      75%     100%
## 187.5571 264.2590 283.5563 304.7085 474.1701
##
## $Sig_Sq$B
##       0%      25%      50%      75%     100%
## 185.8658 270.7788 289.6939 310.3342 510.8360
##
##
## --------------------------------------------
##
```

```
## P(A > B) by (0, 0)%:
##
## $Mu
## [1] 1
##
## $Sig_Sq
## [1] 0.44231
##
## --------------------------------------------
##
## Credible Interval on (A - B) / B for interval length(s) (0.9, 0.9) :
##
## $Mu
##        5%       95%
## 0.1584679 0.3014764
##
## $Sig_Sq
##         5%        95%
## -0.2312652  0.2452771
##
## --------------------------------------------
##
## Posterior Expected Loss for choosing A over B:
##
## $Mu
## [1] 0
##
## $Sig_Sq
## [1] 0.07724428
```

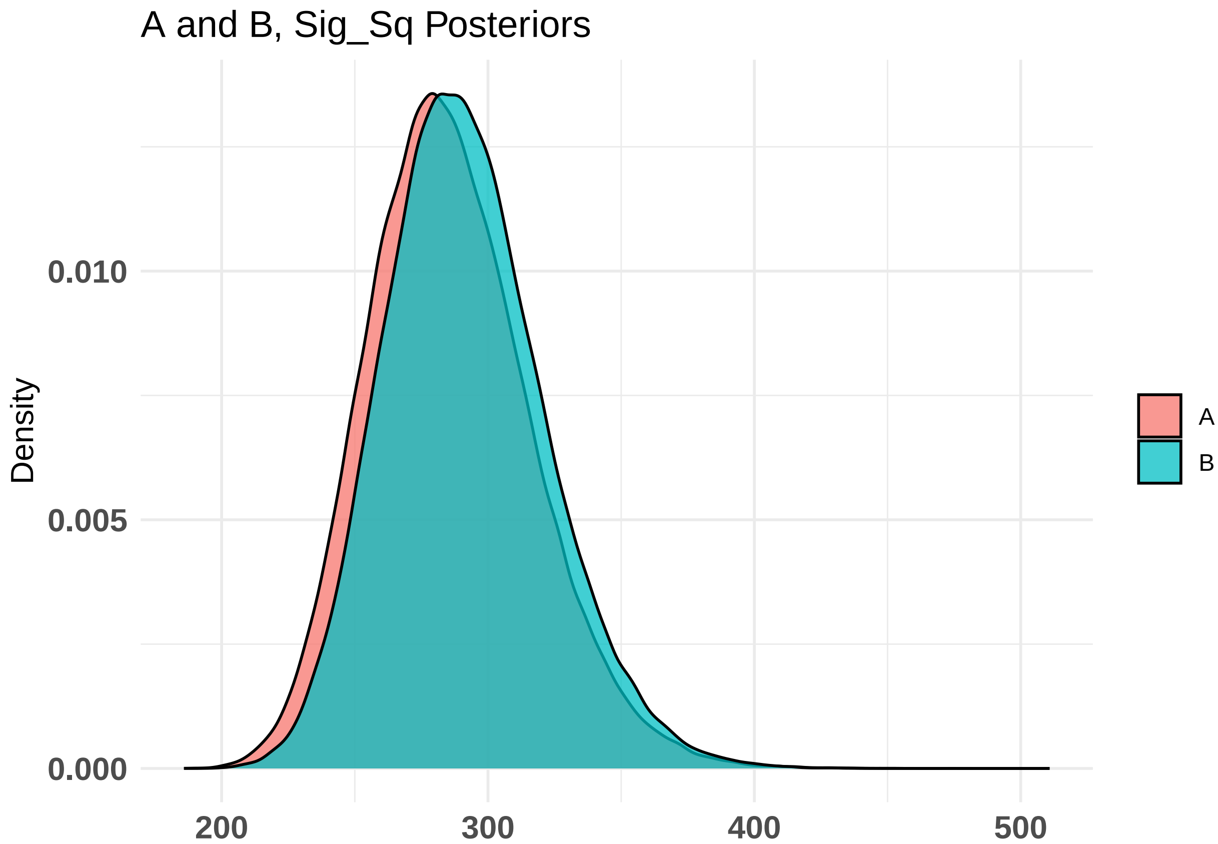
```
plot(pm12)
```

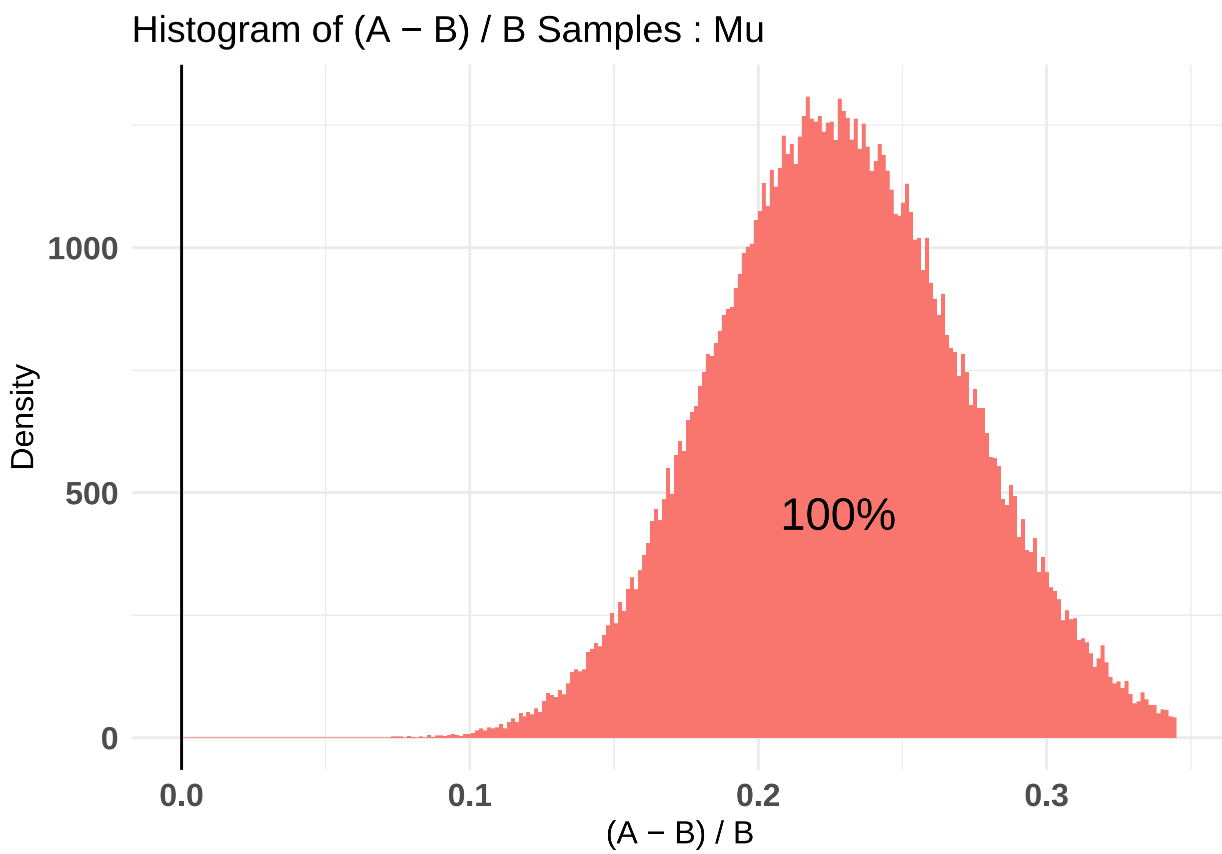
```
## Warning: The dot-dot notation (`..level..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(level)` instead.
## i The deprecated feature was likely used in the bayesAB package.
##   Please report the issue at <https://github.com/FrankPortman/bayesAB/issues>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```


Normal Inverse Gamma PDF for 5, 2, 3, 1
sig_sq
0.6
0.4
0.2
4.5
5.0
5.5
x
Probability Density
0.5
1.0
1.5
2.0
2.5



A and B, Mu Posteriors
Density
0.3
0.2
0.1
0.0
40
45
50
55
60
A
B


A and B, Sig_Sq Posteriors
Density
0.010
0.005
0.000
200
300
400
500
A
B




```r
pm13 <- bayesTest(prom_1, prom_3,
          priors = c('mu' = 5, 'lambda' = 1, 'alpha' = 3, 'beta' = 1),
                 distribution = 'normal')
pm13
```

```
## --------------------------------------------
## Distribution used: normal
## --------------------------------------------
## Using data with the following properties:
##                A         B
## Min.    14.04177  16.57377
## 1st Qu. 46.52201  42.84910
## Median  56.58673  56.01045
## Mean    56.63313  55.75738
## 3rd Qu. 67.51655  67.32566
## Max.    93.10449 102.81068
## --------------------------------------------
## Conjugate Prior Distribution: NormalInvGamma
## Conjugate Prior Parameters:
## $mu
## [1] 5
##
## $lambda
## [1] 1
##
```

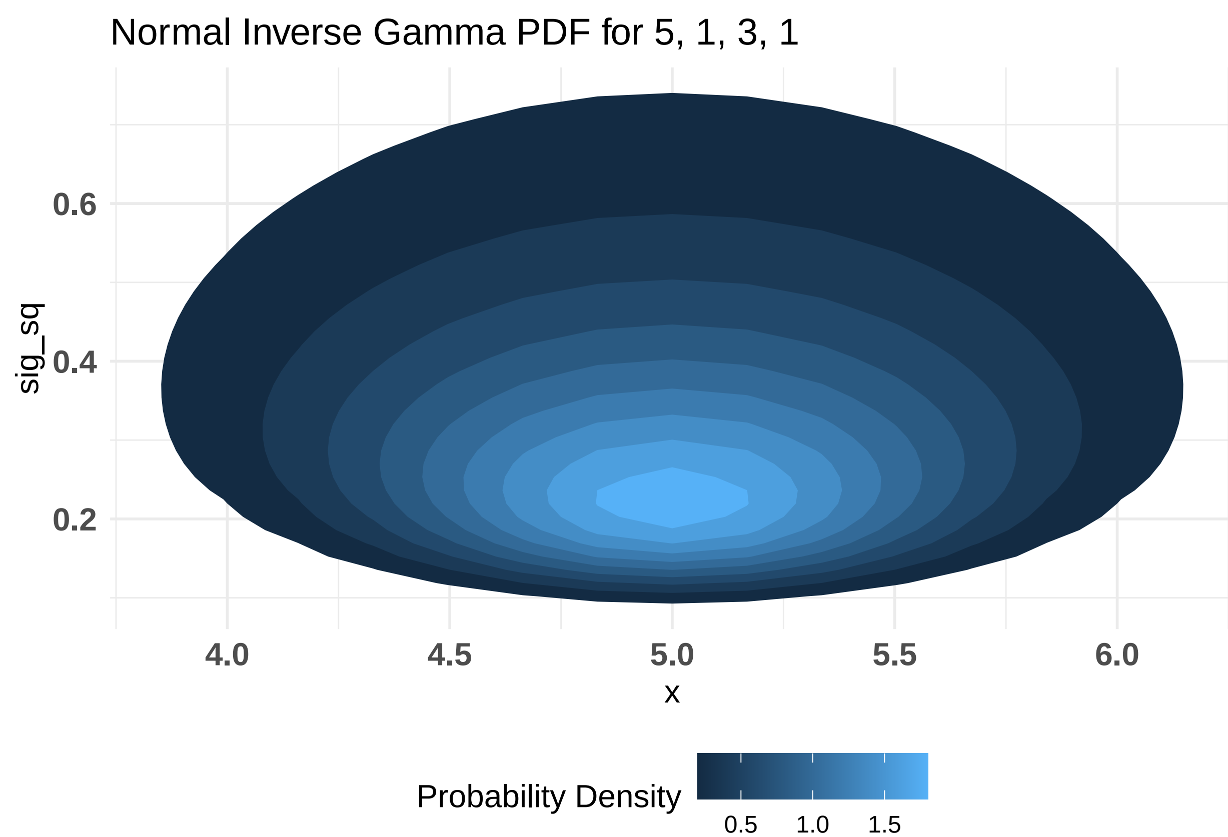
```
## $alpha
## [1] 3
##
## $beta
## [1] 1
##
## --------------------------------------------
## Calculated posteriors for the following parameters:
## Mu, Sig_Sq
## --------------------------------------------
## Monte Carlo samples generated per posterior:
## [1] 1e+05
```

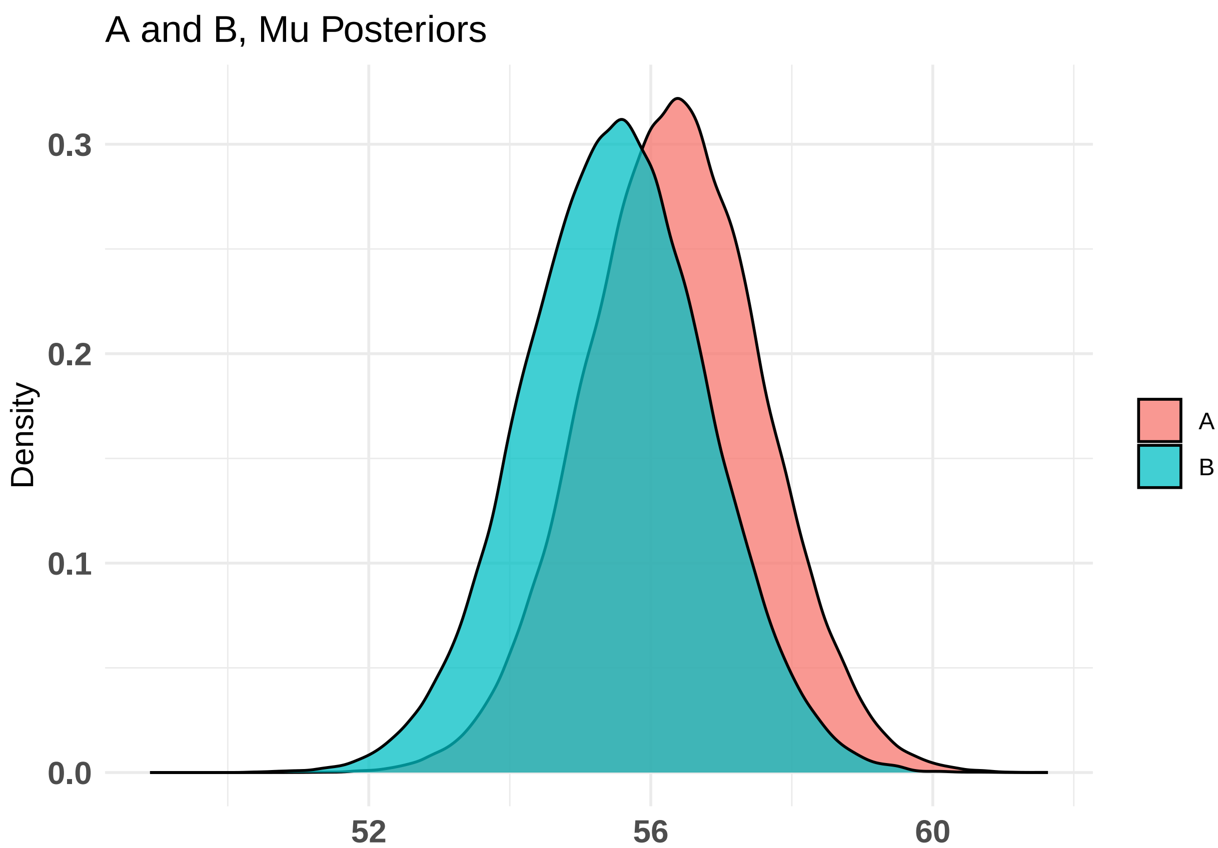
```r
summary(pm13)
```

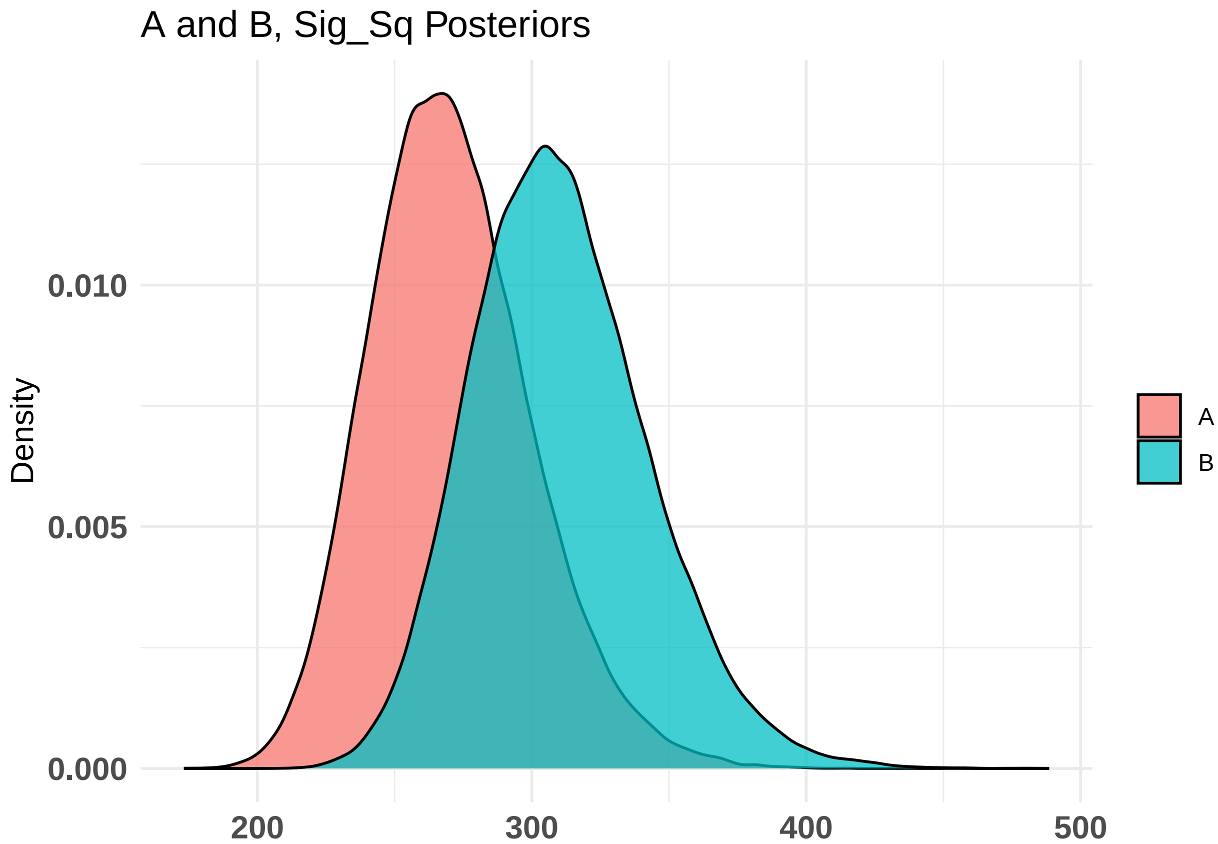
```
## Quantiles of posteriors for A and B:
##
## $Mu
## $Mu$A
##       0%      25%      50%      75%     100%
## 50.95030 55.49015 56.33174 57.17448 61.63357
##
## $Mu$B
##       0%      25%      50%      75%     100%
## 48.89679 54.62247 55.49955 56.36166 61.17065
##
##
## $Sig_Sq
## $Sig_Sq$A
##       0%      25%      50%      75%     100%
## 173.2278 250.2876 268.7051 288.8492 453.6336
##
## $Sig_Sq$B
##       0%      25%      50%      75%     100%
## 208.3234 287.8678 308.1428 330.3772 488.5220
##
##
## --------------------------------------------
##
## P(A > B) by (0, 0)%:
##
## $Mu
## [1] 0.6792
##
## $Sig_Sq
## [1] 0.1751
##
## --------------------------------------------
##
## Credible Interval on (A - B) / B for interval length(s) (0.9, 0.9) :
##
## $Mu
##          5%         95%
## -0.03702531  0.07035827
```

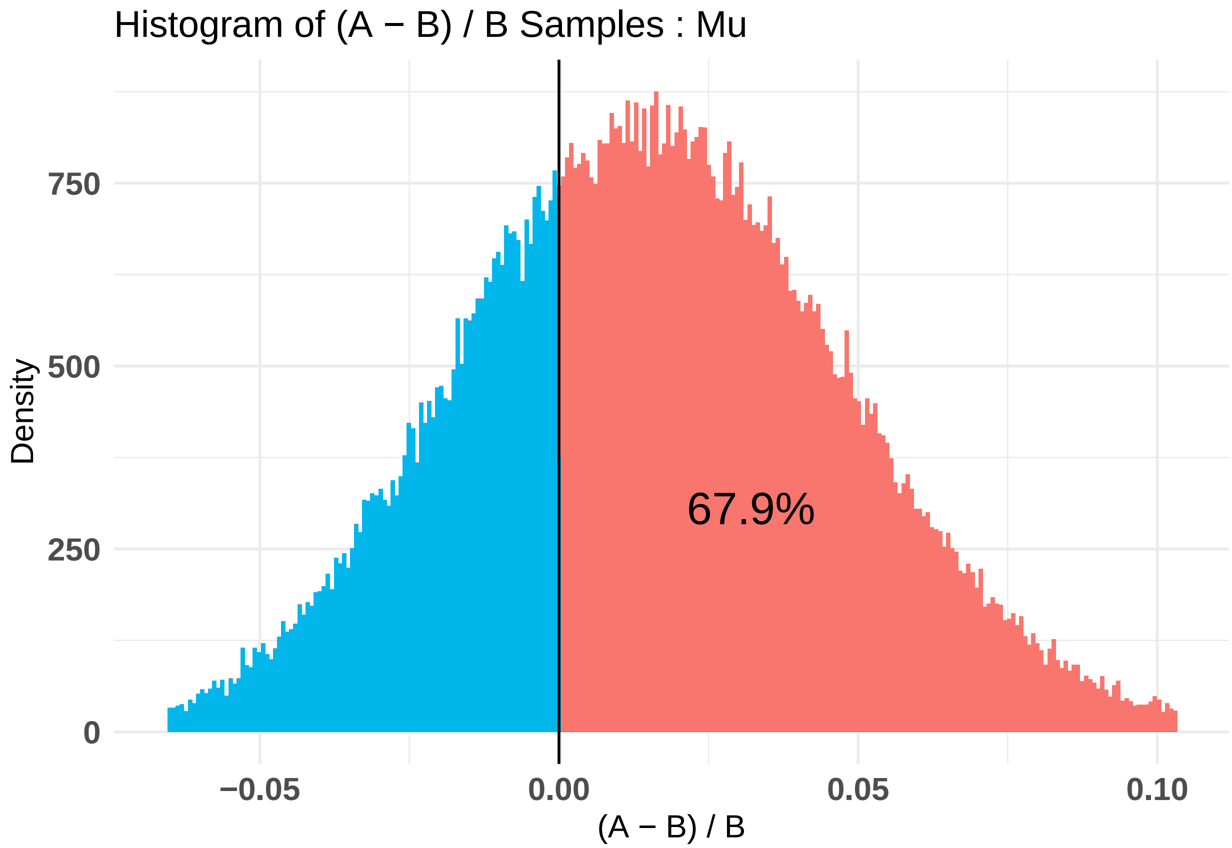
```
##
## $Sig_Sq
##         5%        95%
## -0.3159265  0.1112977
##
## --------------------------------------------
##
## Posterior Expected Loss for choosing A over B:
##
## $Mu
## [1] 0.006795355
##
## $Sig_Sq
## [1] 0.1725381
```

```
plot(pm13)
```




A and B, Mu Posteriors
Density
0.3
0.2
0.1
0.0
52
56
60
A
B


A and B, Sig_Sq Posteriors
Density
0.010
0.005
0.000
200
300
400
500
A
B



Promotion 1 outperform promotion 2, and 3.