MKTG776 HW1

Jordan Farrer

2017-01-23

Contents

1 Question 1

1.1 PMF of Unshifted Beta-Geometric

To derive the unshifted beta-geometric expression for $P(T = t)$, we begin with the statement that since Θ is unobserved, $P(T = t)$ for a randomly chosen individual is the expected value of the unshifted geometric distribution probability of dropping in period t, conditional on $\Theta = \theta$. However, we weight each $P(T = t|\Theta = \theta)$ by the probability of Θ occurring via the beta distribution.

The pmf of the unshifted geometric distribution is defined as

$$P(T = t|\theta) = \theta(1 - \theta)^t, \ t = 0, 1, 2, ... \tag{1}$$

Thus,

$$P(T = t|\alpha, \beta) = \int_0^1 P(T = t|\Theta = \theta) f(\theta|\alpha, \beta) d\theta \tag{2}$$

$$= \int_0^1 \theta(1 - \theta)^t \frac{\theta^{\alpha-1}(1-\theta)^{\beta-1}}{B(\alpha, \beta)} d\theta \tag{3}$$

$$= \frac{1}{B(\alpha, \beta)} \int_0^1 \theta^\alpha (1-\theta)^{\beta+t-1} d\theta \tag{4}$$

We note that the integral above is similar to the integral expression of the beta function

$$B(\alpha, \beta) = \int_0^1 \theta^{\alpha-1}(1-\theta)^{\beta-1} d\theta, \ \alpha, \beta > 0 \tag{5}$$

with parameters $\alpha + 1$ and $\beta + t$. Therefore,

1

$$P(T = t|\alpha, \beta) = \frac{B(\alpha + 1, \beta + t)}{B(\alpha, \beta)} \quad t = 0, 1, 3, ... \tag{6}$$

1.2 Survival Function of Unshifted Beta-Geometric

For the survival function, $S(t)$, we perform similar steps. However, we need to first determine the survival function of an unshifted geometric distribution. Using (1) and the fact that the sum of an infinite geometric series is equal to $\frac{a}{1-r}$, by

$$a + ar + ar^2 + ar^3 + \cdots = \frac{a}{1 - r} \tag{7}$$

we find

$$P(T > t) = P(T \geq t + 1) = \sum_{x=t+1}^{\infty} \theta(1 - \theta)^x = \theta[(1 - \theta)^{t+1} + (1 - \theta)^{t+2} + \cdots] \tag{8}$$

$$= \theta[\frac{(1 - \theta)^{t+1}}{1 - (1 - \theta)}] \tag{9}$$

$$= \theta\frac{(1 - \theta)^{t+1}}{\theta} \tag{10}$$

$$= (1 - \theta)^{t+1} \tag{11}$$

Thus,

$$S(t) = P(T > t|\alpha, \beta) = \int_0^1 P(T > t|\Theta = \theta) f(\theta|\alpha, \beta) d\theta \tag{12}$$

$$= \int_0^1 (1 - \theta)^{t+1} \frac{\theta^{\alpha-1}(1 - \theta)^{\beta-1}}{B(\alpha, \beta)} d\theta \tag{13}$$

$$= \frac{1}{B(\alpha, \beta)} \int_0^1 \theta^{\alpha-1}(1 - \theta)^{\beta+t} d\theta \tag{14}$$

Like in the derivation of $P(T = t|\alpha, \beta)$, we note that the integral above (14) is the beta function with parameters α and $\beta + t + 1$. Therefore,

$$S(t) = P(T > t|\alpha, \beta) = \frac{B(\alpha, \beta + t + 1)}{B(\alpha, \beta)}, \quad t = 0, 1, 3, ... \tag{15}$$

1.3 Forward-Recursion Algorithm for $P(T = t)$

To derive the forward-recursion algorithm needed to compute unshifted beta-geometric probabilities, we first note that

$$P(T = 0|\alpha, \beta) = \frac{B(\alpha + 1, \beta)}{B(\alpha, \beta)} \tag{16}$$

We know that the beta function can be expressed in terms of gamma functions:

$$B(\alpha, \beta) = \frac{\Gamma(\alpha), \Gamma(\beta)}{\Gamma(\alpha + \beta)} \tag{17}$$

Thus, we can rewrite (16) as

$$= \frac{\frac{\Gamma(\alpha+1)\Gamma(\beta)}{\Gamma(\alpha+\beta+1)}}{\frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}} \tag{18}$$

and simplify

$$= \frac{\Gamma(\alpha + 1)\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\alpha + \beta + 1)} \tag{19}$$

We recall the recursive nature of the gamma function, i.e.

$$\frac{\Gamma(x + 1)}{\Gamma(x)} = x \tag{20}$$

So, we can rewrite (19) as

$$= \frac{\alpha\Gamma(\alpha)\Gamma(\alpha + \beta)}{\Gamma(\alpha)(\alpha + \beta)\Gamma(\alpha + \beta)} \tag{21}$$

$$= \frac{\alpha}{\alpha + \beta} \tag{22}$$

This is the same formula as $P(T = 1)$ for the shifted beta-geometric distribution. However, the ratio of $\frac{P(T=t)}{P(T=t-1)}$ will be different. We seek to find this ratio because of the identity

$$P(T = t) = \frac{P(T = t)}{P(T = t - 1)} P(T = t - 1) \tag{23}$$

As we can find the value of $P(T = 0)$, we can find each subsequent value of $P(T = t)$ by using the previous. We express this ratio first as beta and then as gamma functions:

$$\frac{P(T = t)}{P(T = t - 1)} = \frac{\frac{B(\alpha+1,\beta+t)}{B(\alpha,\beta)}}{\frac{B(\alpha+1,\beta+t-1)}{B(\alpha,\beta)}} \tag{24}$$

$$= \frac{B(\alpha + 1, \beta + t)}{B(\alpha + 1, \beta + t - 1)} \tag{25}$$

$$= \frac{\frac{\Gamma(\alpha+1)\Gamma(\beta+t)}{\Gamma(\alpha+\beta+t+1)}}{\frac{\Gamma(\alpha+1)\Gamma(\beta+t-1)}{\Gamma(\alpha+\beta+t)}} \tag{26}$$

$$= \frac{\Gamma(\beta + t)\Gamma(\alpha + \beta + t)}{\Gamma(\alpha + \beta + t + 1)\Gamma(\beta + t + 1)} \tag{27}$$

$$= \frac{(\beta + t - 1)\Gamma(\beta + t - 1)\Gamma(\alpha + \beta + t)}{(\alpha + \beta + t)\Gamma(\alpha + \beta + t)\Gamma(\beta + t - 1)} \tag{28}$$

$$= \frac{\beta + t - 1}{\alpha + \beta + t} \tag{29}$$

Thus,

$$P(T = t) = \begin{cases} \frac{\alpha}{\alpha+\beta} & t = 0 \\ \frac{\beta+t-1}{\alpha+\beta+t} P(T = t-1) & t = 1, 2, 3, ... \end{cases} \tag{30}$$

1.4 Forward-Recursion Algorithm for r_t

We can derive an expression for the retention rate in period t implied by the sBG model by substituting the expression for the sBG survivor function (15) into the definition:

$$r_t = \frac{S(t)}{S(t-1)} \tag{31}$$

So,

$$r_t = \frac{S(t)}{S(t-1)} = \frac{\frac{B(\alpha,\beta+t+1)}{B(\alpha,\beta)}}{\frac{B(\alpha,\beta+t)}{B(\alpha,\beta)}} \tag{32}$$

$$= \frac{B(\alpha, \beta+t+1)}{B(\alpha, \beta+t)} \tag{33}$$

$$= \frac{\frac{\Gamma(\alpha)\Gamma(\beta+t+1)}{\Gamma(\alpha+\beta+t+1)}}{\frac{\Gamma(\alpha)\Gamma(\beta+t)}{\Gamma(\alpha+\beta+t)}} \tag{34}$$

$$= \frac{\Gamma(\beta+t+1)\Gamma(\alpha+\beta+t)}{\Gamma(\alpha+\beta+t+1)\Gamma(\beta+t)} \tag{35}$$

$$= \frac{(\beta+t)\Gamma(\beta+t)\Gamma(\alpha+\beta+t)}{(\alpha+\beta+t)\Gamma(\alpha+\beta+t)\Gamma(\beta+t)} \tag{36}$$

$$= \frac{\beta+t}{\alpha+\beta+t} \tag{37}$$

2 Question 2

We first load the provided customer data.

```
pacman::p_load(tidyverse, forcats, pander)
panderOptions('round', 2)
panderOptions('keep.trailing.zeros', TRUE)

customer_data <-
  readxl::read_excel("Homework data.xlsx",
                col_names = c("Year","Regular","High End","Empty"), skip = 1) %>%
  select(1:3) %>%
  filter(row_number() <= 13)

customer_data
```

Year	Regular	High End
0	1000	1000
1	631	869
2	468	743

Year	Regular	High End
3	382	653
4	326	593
5	289	551
6	262	517
7	241	491
8	223	468
9	207	445
10	194	427
11	183	409
12	173	394

Next we define functions that implement the shifted beta-geometric (sBG) algorithm.

```r
# Calculates P(T=t) using recursion
fn_p_t <- function(alpha, beta, t) {
  if (t == 1) {
    p_t <- alpha / (alpha + beta)
    return(p_t)
  } else {
    p_t <- (beta + t - 2) / (alpha + beta + t - 1) * fn_p_t(alpha, beta, t -1) # Recursive
    return(p_t)
  }
}


# Calculates the log-likelihood of initial alpha and beta
# given data (in this case the percent customers alive from t = 1)
fn_max_ll <- function(par, percent_alive) {
  alpha <- par[1]
  beta <- par[2]
  len_alive <- length(percent_alive)

  percent_dead <- abs(diff(c(1, percent_alive)))
  p_t <- sapply(1:len_alive, function(x) {fn_p_t(alpha, beta, x)})
  s_t <- 1 - sum(p_t)

  ll <- sum((percent_dead * log(p_t)))  + percent_alive[length(percent_alive)] * log(s_t)

  return(-ll)
}


# Using calculated alpha and beta, predict the customers
# alive in each period
fn_sGB_predict <- function(alpha, beta, last_period) {
  p_t <- sapply(1:last_period, function(x) {fn_p_t(alpha, beta, x)})
  pred_surival <- 1 - cumsum(p_t)
  return(pred_surival)
}
```

The for both Regular and High End customer types and both 7- and 10-period calibrations, we run the sBG model and put the results in a table.

```r
results <-
  customer_data %>%
  gather(customer_type, number_alive, -Year) %>%
  rename(year = Year) %>%
  mutate(calibration = 'Actual') %>%
  select(customer_type, calibration, year, number_alive)
```

```r
param_history <- data_frame()

for(customer_type in c('Regular','High End')) {
  for(calibration in c(7, 10)) {

    mutate_call <- lazyeval::interp(~ a /1000, a = as.name(customer_type))

    percent_alive <-
      (customer_data %>%
         mutate_(.dots = setNames(list(mutate_call), "percent_alive")) %>%
         filter(Year >= 1 & Year <= calibration))$percent_alive

    params <- optim(par = c(1,1), fn_max_ll, gr = NULL, percent_alive)
    predicted <- fn_sGB_predict(alpha = params$par[1], beta = params$par[2], last_period = 12)

    param_history <-
      param_history %>%
      bind_rows(
        data_frame(
          customer_type = customer_type
            , calibration = paste0(calibration,"-period")
            , alpha = params$par[1]
            , beta = params$par[2]
        )
      )

    results <-
      results %>%
      bind_rows(
        data_frame(
              customer_type = customer_type
            , calibration = paste0(calibration,"-period")
            , year = 0
            , number_alive = 1000
          )
      ) %>%
      bind_rows(
          data_frame(
            customer_type = customer_type
            , calibration = paste0(calibration,"-period")
            , year = 1:12
            , number_alive = predicted * 1000
          )
      )
  }
}
```

Next, we review the results for regular and then High End customers.

```r
results %>%
  filter(customer_type == "Regular") %>%
  spread(calibration, number_alive) %>%
  select(Year = year, Actual, `7-period`, `10-period`) %>%
  pander(caption = "Regular Customers", justify = c('center', rep('right',3)))
```

Year	Actual	7-period	10-period

Table 2: Regular Customers

Year	Actual	7-period	10-period
0	1000	1000.0	1000.0
1	631	626.7	626.5
2	468	473.8	474.0
3	382	388.0	388.4
4	326	332.1	332.7
5	289	292.4	293.1
6	262	262.5	263.3
7	241	239.0	239.9
8	223	220.1	221.0
9	207	204.4	205.4
10	194	191.2	192.2
11	183	179.9	180.9
12	173	170.1	171.1

```
results %>%
  filter(customer_type == "High End") %>%
  spread(calibration, number_alive) %>%
  select(Year = year, Actual, `7-period`, `10-period`) %>%
  pander(caption = "High End Customers", justify = c('center', rep('right',3)))
```

Table 3: High End Customers

Year	Actual	7-period	10-period
0	1000	1000.0	1000.0
1	869	850.6	848.5
2	743	746.8	745.7
3	653	669.7	670.4
4	593	609.9	612.4
5	551	561.8	566.0
6	517	522.2	528.0
7	491	488.9	496.0
8	468	460.4	468.7
9	445	435.8	445.1
10	427	414.2	424.3
11	409	395.1	406.0
12	394	378.0	389.6

Let's also look at a plot the estimates and the actuals.

```
results %>%
  ggplot(aes(x = year, y = number_alive,
             colour = fct_rev(calibration), linetype = fct_rev(calibration))) +
  geom_line() +
  facet_grid(. ~ forcats::fct_rev(customer_type)) +
  scale_x_continuous(breaks = scales::pretty_breaks()) +
  labs(title = 'Shifted Beta-Geometric Model', y = 'Number of Customers Alive', x = 'Year') +
  theme(legend.position = "top", legend.title = element_blank())
```

Shifted Beta–Geometric Model
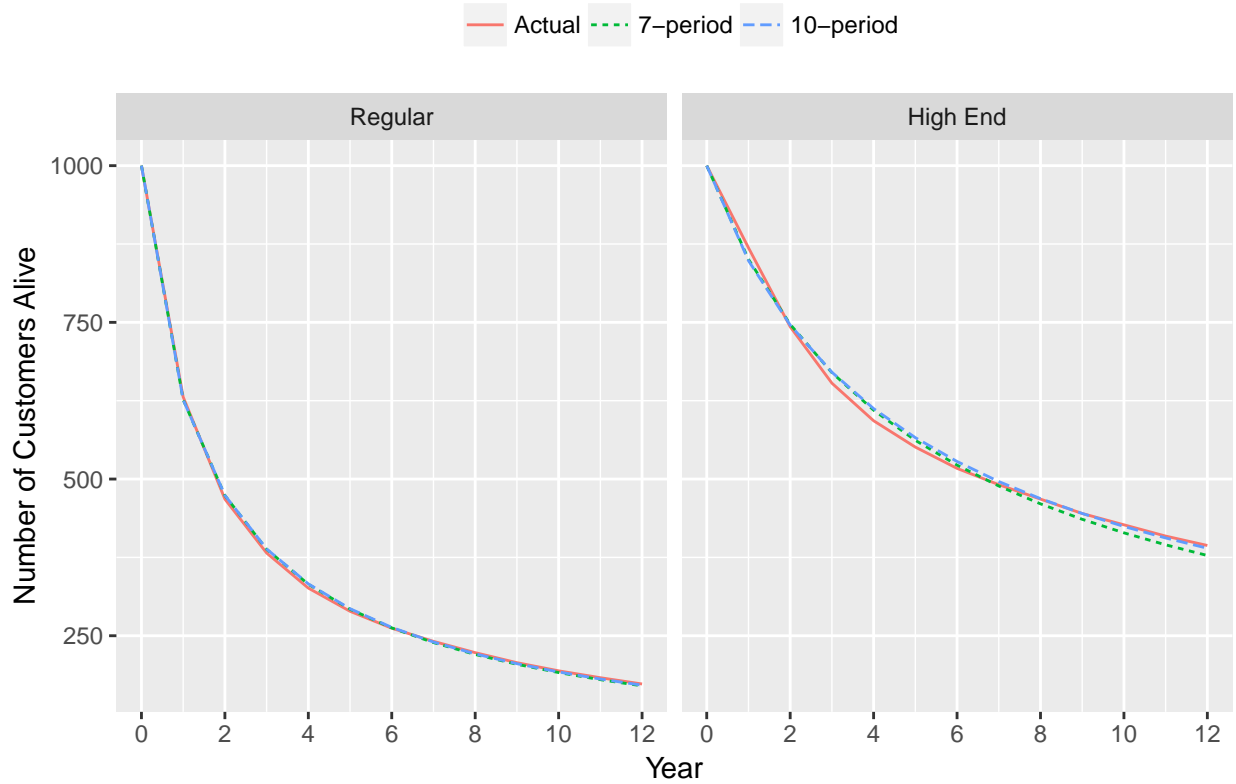
The lines are extremely close together, so let's look at the percent error to get a better idea. First the Regular Customers:

```
results %>%
  filter(customer_type == "Regular") %>%
  spread(calibration, number_alive) %>%
  mutate(
    `7-period % Error` = paste0(formatC(abs((Actual - `7-period`) / Actual) * 100,
                                    digits = 2, format = "f"),'%')
    , `10-period % Error` = paste0(formatC(abs((Actual - `10-period`) / Actual) * 100,
                                    digits = 2, format = "f"),'%')
  ) %>%
  select(Year = year, `7-period % Error`, `10-period % Error`) %>%
  pander(align = c('center','right','right'),
         caption = "Regular Customers % Error Between Actual and Estimate")
```

Table 4: Regular Customers % Error Between Actual and Estimate

Year	7-period % Error	10-period % Error
0	0.00%	0.00%
1	0.68%	0.71%
2	1.25%	1.29%
3	1.57%	1.69%
4	1.87%	2.06%
5	1.17%	1.43%
6	0.18%	0.50%
7	0.81%	0.44%
8	1.30%	0.88%
9	1.24%	0.77%

Year	7-period % Error	10-period % Error
10	1.44%	0.93%
11	1.70%	1.16%
12	1.70%	1.12%

Then, the High End customers:

```
results %>%
  filter(customer_type == "High End") %>%
  spread(calibration, number_alive) %>%
  mutate(
    `7-period % Error` = paste0(formatC(abs((Actual - `7-period`) / Actual) * 100,
                                 digits = 2, format = "f"),'%')
    , `10-period % Error` = paste0(formatC(abs((Actual - `10-period`) / Actual) * 100,
                                 digits = 2, format = "f"),'%')
  ) %>%
  select(Year = year, `7-period % Error`, `10-period % Error`) %>%
  pander(align = c('center','right','right'),
         caption = "High End Customers % Error Between Actual and Estimate")
```

Table 5: High End Customers % Error Between Actual and Estimate

Year	7-period % Error	10-period % Error
0	0.00%	0.00%
1	2.11%	2.36%
2	0.51%	0.36%
3	2.56%	2.67%
4	2.84%	3.27%
5	1.96%	2.73%
6	1.00%	2.12%
7	0.43%	1.02%
8	1.62%	0.15%
9	2.07%	0.01%
10	3.01%	0.62%
11	3.41%	0.74%
12	4.06%	1.12%

We find that for both calibrations, the estimate for the Regular customers is better (lower MAPE) than the estimate for High End Customers. For both customer types, the 7-period calibration performs worse in later periods than the 10-period calibration. This is expected because the 10-period calibration only needed to estimate two periods in the future whereas the 7-period needed to estimate five periods in the future. Interestingly, both both Regular and High End customers, the 7-period calibration performs better than the 10-period calibration in periods 3 and 4.

Let's also look at the different parameters of the sBG model:

```
param_history %>%
  pander(caption = "sBG Parameters")
```

Table 6: sBG Parameters

customer_type	calibration	alpha	beta
Regular	7-period	0.70	1.18
Regular	10-period	0.70	1.17
High End	7-period	0.67	3.80
High End	10-period	0.61	3.39

Recognizing that alpha and beta are parameters for the beta distribution which weights the geometric distribution, we can look at the distribution of the unobserved parameter θ. θ is the probability a customer will leave in period t. By sampling from the beta distribution for the 4 sets of parameters alpha and beta, we can show the estimated distributions of the churn probabilities. There is high density of low theta's for the High End customers as their propensity to churn is lower than Regular customers. While the shape of the Regular customers' distribution is similar to the High End - they are both J-shaped as for all four estimates have alpha < 1 and beta > 1 - there are more customers with a higher propensity to churn among Regular customers.

```
estimation <-
  param_history %>%
  mutate(
    density = map2(alpha, beta, function(.x, .y) {rbeta(10000, .x, .y)})
  ) %>%
  mutate(
    label1 = paste0("atop(alpha == ", formatC(alpha, digits = 2, format = 'f'),
                    ',beta == ', formatC(beta, digits = 2, format = 'f'),')')
    , label2 = paste0('E(theta) == ', formatC(alpha / (alpha + beta), digits = 2, format = 'f'))
  )

estimation %>%
  unnest() %>%
  ggplot(aes(x = density, colour = calibration, fill = calibration)) +
  geom_density(alpha = 1/3) +
  facet_grid( forcats::fct_rev(calibration) ~ forcats::fct_rev(customer_type)) +
  theme(legend.position = 'none') +
  geom_text(data = estimation, aes(x = .75, y = 4, label = label1), parse = TRUE) +
  geom_text(data = estimation, aes(x = .75, y = 3, label = label2), parse = TRUE) +
  labs(x = expression(theta), y = expression(f(theta)))
```
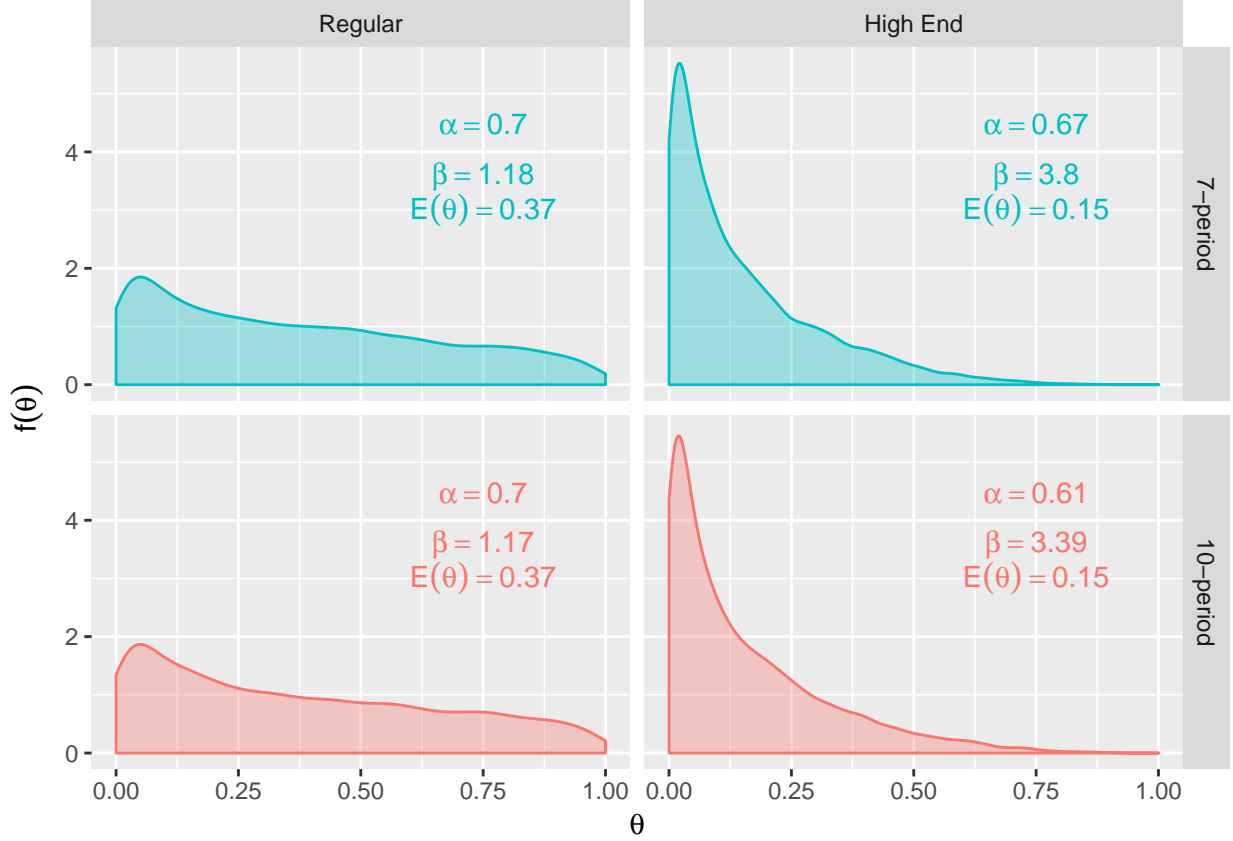
3 Question 3

The fact that $P(T = 1)$ for the sBG equals the mean of the sBG is not surprising because the shifted geometric when $t = 1$ is just θ. Thus, $P(T = 1)$ is simply θ times the beta distribution, which by definition is the expectation.

Let $X \sim Beta(\alpha, \beta)$, then

$$E[X] = \int_{-\infty}^{\infty} x f(x) dx = \int_0^1 \theta \frac{\theta^{\alpha-1}(1-\theta)^{\beta-1}}{B(\alpha, \beta}d\theta \tag{38}$$

$$= P(T = 1|\alpha, \beta) = \int_0^1 \theta(1-\theta)^{1-1}\frac{\theta^{\alpha-1}(1-\theta)^{\beta-1}}{B(\alpha, \beta}d\theta \tag{39}$$

$$= \int_0^1 \theta \frac{\theta^{\alpha-1}(1-\theta)^{\beta-1}}{B(\alpha, \beta}d\theta \tag{40}$$

4 Question 4

The implied shape of the retention curve under the beta-geometric is an increasing function of the length of a customer's relationship with the firm. The increasing retention rates in later periods is *not* a result of customers becoming "more loyal" (an common explanation), but rather is a result of a sorting effect in the heterogeneous population (customers with high-churn probabilities drop out earlier and the remaining customers have lower churn rates).

The figure below shows how the retention rate curve changes with different model parameters:

```r
fn_retention <- function(alpha, beta, t) {
  return((beta + t - 1) / (alpha + beta + t - 1))
}

expand.grid(alpha = c(.5, 1.5), beta = c(.5, 1.5), t = 1:12) %>%
  as_tibble() %>%
  arrange(desc(beta), alpha) %>%
  mutate(
    retention_rate = fn_retention(alpha, beta, t)
    , alpha_beta = paste0("alpha == ", alpha, "~~ beta == ", beta)
    , alpha_beta = fct_inorder(alpha_beta)
  ) %>%
  ggplot(aes(x = t, y = retention_rate)) +
  geom_line() +
  facet_wrap(~alpha_beta, labeller= label_parsed) +
  scale_x_continuous(breaks = scales::pretty_breaks()) +
  labs(x = "Period", y = "Retention Rate", title = "Retention Rates for Different Parameters") +
  scale_y_continuous(labels = scales::percent)
```
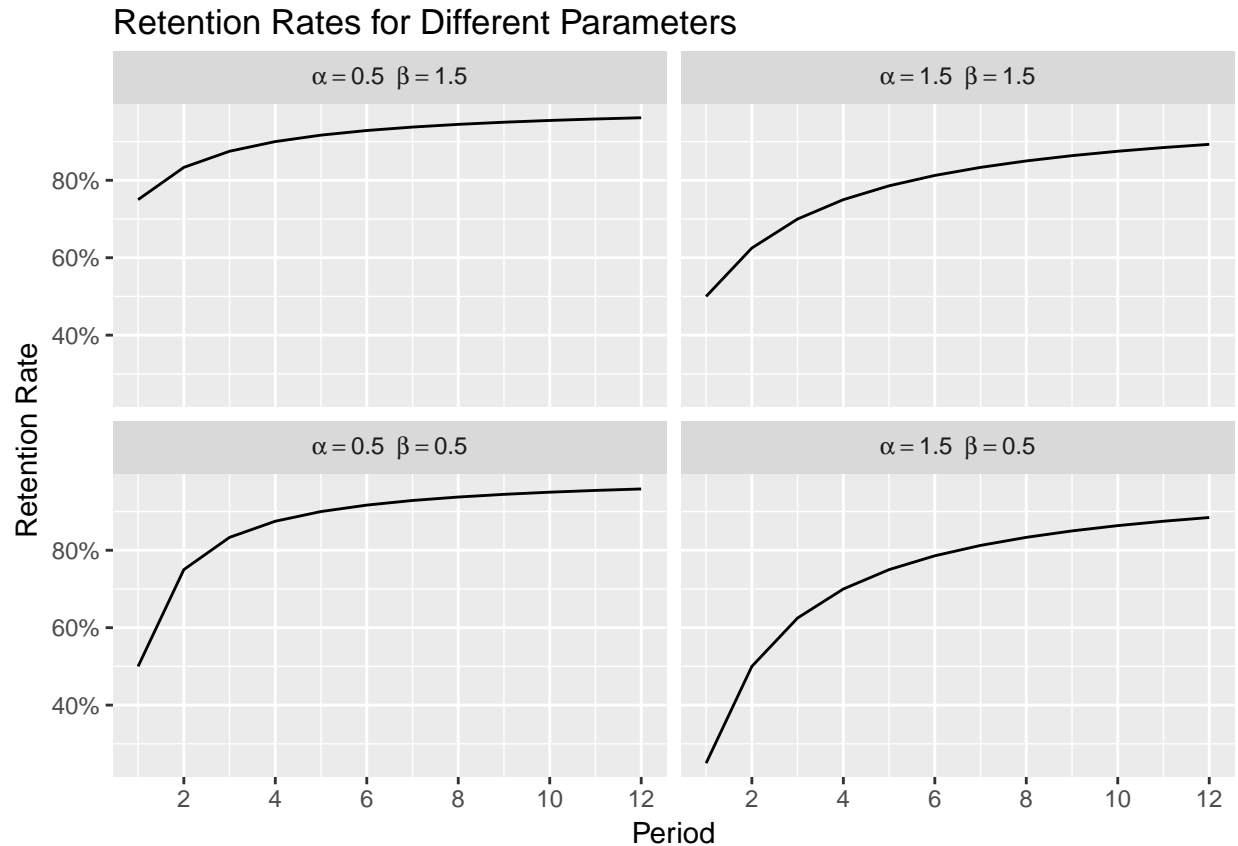

We see that an optimal place for the firm is in the upper left-hand corner where alpha is less than 1 and beta is greater than 1. Under these parameters, retention is higher in earlier periods. A sub-optimal place for the firm is the lower right-hand corner where alpha is greater than 1 and beta is less than 1. Under these parameters, a large percentage of the customers leave in initial periods. However, for all four values of alpha and beta, the retention rates reaches an asymptote between 88% and 96% in period 12.

The differences are explained by the heterogeneity of the customers: a beta distribution with alpha less than 1 and beta greater than 1 has more customers with lower churn probabilities, θ, and a beta distribution

with alpha greater than 1 and beta less than 1 has more customers with higher churn probabilities. This relationship is shown in the figure below where each correspond to one of the four retention rate plots above.

```r
expand.grid(alpha = c(.5, 1.5), beta = c(.5, 1.5)) %>%
  as_tibble() %>%
  arrange(desc(beta), alpha) %>%
  mutate(
    theta = map2(alpha, beta, function(.x, .y) {rbeta(10000, .x, .y)})
  ) %>%
  mutate(
      alpha_beta = paste0("alpha == ", alpha, "~~ beta == ", beta)
    , alpha_beta = fct_inorder(alpha_beta)
  ) %>%
  unnest() %>%
  ggplot(aes(x = theta)) +
  geom_line(stat = "density") +
  facet_wrap(~alpha_beta, labeller= label_parsed) +
  scale_x_continuous(breaks = scales::pretty_breaks()) +
  labs(x = expression(theta), y = expression(f(theta)),
       title = "Estimated Distributions of Churn Probabilities")
```
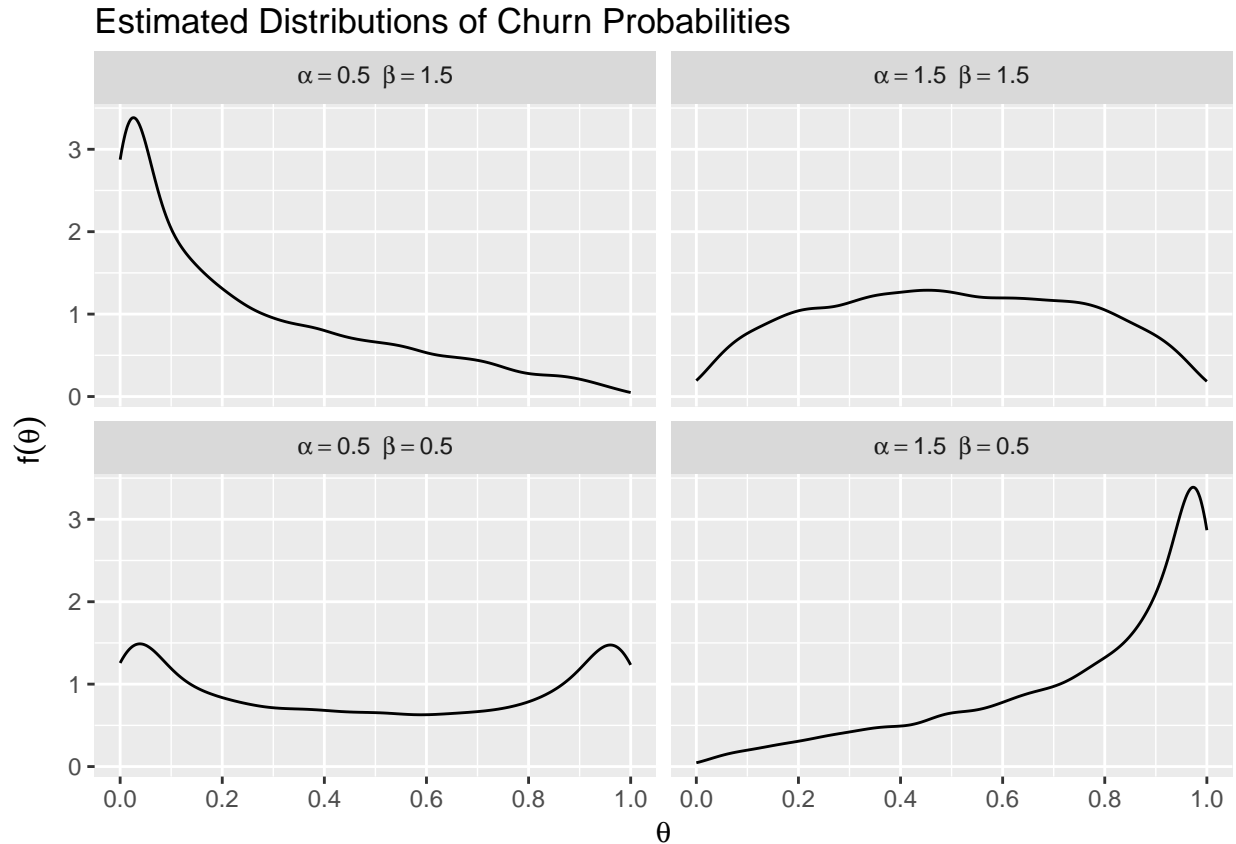

4.0.1 References

Derivation from "How to Project Customer Retention"