

Pricing Strategy Analytics

XG

9/8/2018

Background

At Zillow new construction sales team, we are facing a challenge about pricing: which pricing model to choose for new construction communities?

- **Community model** (labeled as `comm_mdl` in the code): charge a flat rate of \$400 / community / month
- **Lead model** (labeled as `lead_mdl` in the code): charge \$4 / lead / community / month

The following criteria are taken into consideration while making such choice:

- Prioritizing long-term revenue opportunity over short-term.
- Providing the best experience for the consumer possible (More listings is better)

In addition, the evaluation on the future growth rate under two models were given.

```
library(devtools)
library(magrittr)
library(dplyr)
library(reshape)
library(lubridate)
library(ggplot2)
library(ggpmisc)
```

Q1: How many communities does each method result in for December 2018?

- Here we are making such assumption: since the expected value of monthly growth of leads per community and the expected value of monthly growth of number of communities are given, we assume that each month's growth rate was drawn from a normal distribution (distribution mean = expected growth rate, variance = yearly variance / 36)
- The estimation of monthly growth variance is based on central limit theorem, in which sample variance = population variance / sample size; in this case sample size equals to the total number of month from 2016 to 2018.

```
set.seed(100)

## known parameter
cur_leads_per_comm_per_month <- 4.00
leads_per_comm_per_month <- c(5, 4, 1)/100
comm_per_month_lead_mdl <- c(6, 4, 2)/100
comm_per_month_comm_mdl <- comm_per_month_lead_mdl * .9

## leads in january
jan_comm <- 6174
jan_leads <- cur_leads_per_comm_per_month * jan_comm

#### assumption: each month per leads grow follows a distribution
## normal
```

```

std_growth_leads_per_comm_per_month <- sd(leads_per_comm_per_month)/sqrt(36)

## function to calculate yearly rate
yearly_rate <- function(year, myrate){
  if (year == 1){
    new_rate <- rnorm(11, mean = myrate[year], sd = sd(myrate)/sqrt(36))
  } else {new_rate <- rnorm(12, mean = myrate[year], sd = sd(myrate)/sqrt(36))}
  #print(new_rate)
  return(new_rate)
}

## function to calculate the number of communities based on different growth rate
calc_comm <- function(myrate){
  year2016 <- jan_comm * cumprod(1 + c(0, yearly_rate(1, myrate)))
  year2016 <- as.integer(year2016)
  #print('Dec 2018: ')
  #print(year2016[12])
  year2017 <- year2016[12] * cumprod(1 + yearly_rate(2, myrate))
  year2017 <- as.integer(year2017)
  #print( year2017[12])
  year2018 <- year2017[12] * cumprod(1 + yearly_rate(3, myrate))
  year2018 <- as.integer(year2018)
  #print(year2018[12])
  return(c(year2016, year2017, year2018))
}

## this is the number of community each month under lead model (1000 times simulation)
new_comm_monthly_lead_md1 <- replicate(1000, calc_comm(comm_per_month_lead_md1))
new_comm_monthly_lead_md1_df <- new_comm_monthly_lead_md1 %>%
  as.data.frame() %>%
  mutate(ym = make_date(year = rep(c(2016:2018), 1, each = 12), month = rep(c(1:12), 3, each = 1))) %>%
  mutate(md1 = 'lead') %>%
  melt(id = c('ym', 'md1'))

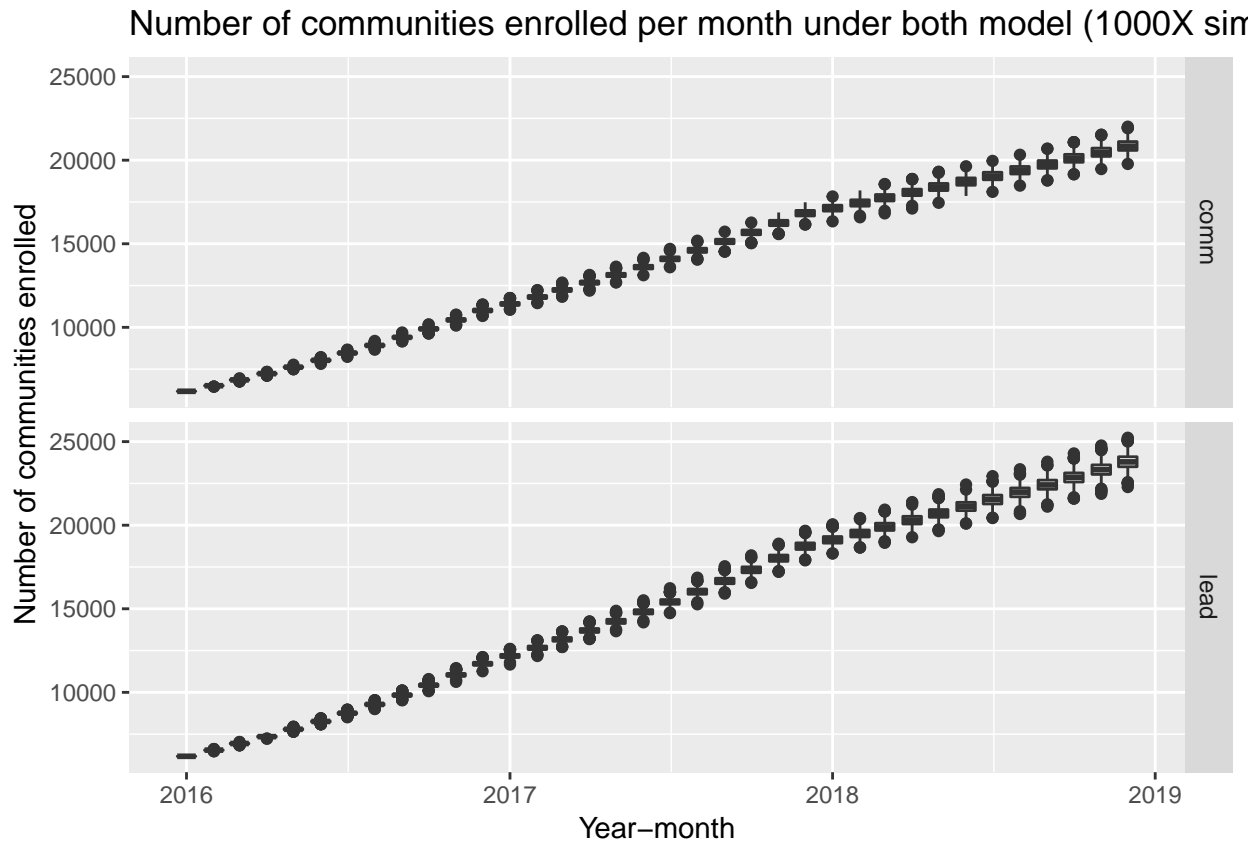
## for per community model

new_comm_monthly_comm_md1 <- replicate(1000, calc_comm(comm_per_month_comm_md1))
new_comm_monthly_comm_md1_df <- new_comm_monthly_comm_md1 %>%
  as.data.frame() %>%
  mutate(ym = make_date(year = rep(c(2016:2018), 1, each = 12), month = rep(c(1:12), 3, each = 1))) %>%
  mutate(md1 = 'comm') %>%
  melt(id = c('ym', 'md1'))

q1_join <- rbind(new_comm_monthly_lead_md1_df, new_comm_monthly_comm_md1_df)

q1_join %>% ggplot(aes(x = ym, y = value, group = ym)) +
  geom_boxplot() +
  ylab('Number of communities enrolled') +
  xlab('Year-month') +
  facet_grid(rows = vars(md1)) +
  ggtitle('Number of communities enrolled per month under both model (1000X simulation)')

```



```
print(paste0("Lead model: number of communities in Dec 2018: ", as.integer(mean(new_comm_monthly_lead_m
## [1] "Lead model: number of communities in Dec 2018: 23914"
print(paste0("Community model: number of communities in Dec 2018: ", as.integer(mean(new_comm_monthly_c
## [1] "Community model: number of communities in Dec 2018: 20931"
```

Q2. What is the overall revenue impact?

- For lead model: revenue = number of communities * number of leads per community per month X 40\$
- For community model: revenue = number of communities * 400\$

To answer this question, let's first take a look at monthly revenue from 2016 to 2018 under two different model.

```
## Per lead model
## calculate lead growth rate per month
cal_lead_rate <- function(myrate){
  year2016 <- cur_leads_per_comm_per_month * cumprod(1 + c(0, yearly_rate(1, myrate)))
  year2017 <- year2016[12] * cumprod(1 + yearly_rate(2, myrate))
  #print( year2017[12])
  year2018 <- year2017[12] * cumprod(1 + yearly_rate(3, myrate))
  #print(year2018)
  return(c(year2016, year2017, year2018))
}

## this is the leads per comm per month for each month under leads model
```

```

leads_monthly_leads_md1 <- replicate(1000, cal_lead_rate(leads_per_comm_per_month))

## use this to multiply community
total_lead_subs_lead_md1 <- leads_monthly_leads_md1 * new_comm_monthly_lead_md1 * 40

total_lead_subs_lead_md1_df <- total_lead_subs_lead_md1 %>% as.data.frame() %>%
  mutate(ym = make_date(year = rep(c(2016:2018), 1, each = 12), month = rep(c(1:12), 3, each = 1))) %>%
  mutate(md1 = 'lead') %>%
  melt(id = c('ym', 'md1'))

#### community model
total_comm_subs_comm_md1 <- new_comm_monthly_comm_md1*400
total_comm_subs_comm_md1_df <- total_comm_subs_comm_md1 %>% as.data.frame() %>%
  mutate(ym = make_date(year = rep(c(2016:2018), 1, each = 12), month = rep(c(1:12), 3, each = 1))) %>%
  mutate(md1 = 'comm') %>%
  melt(id = c('ym', 'md1'))

print(paste0("Lead model: total revenue from Jan 2016 to Dec 2018: ", as.integer(mean(colMeans(total_lead_subs_lead_md1_df)))))

## [1] "Lead model: total revenue from Jan 2016 to Dec 2018: 5825091"

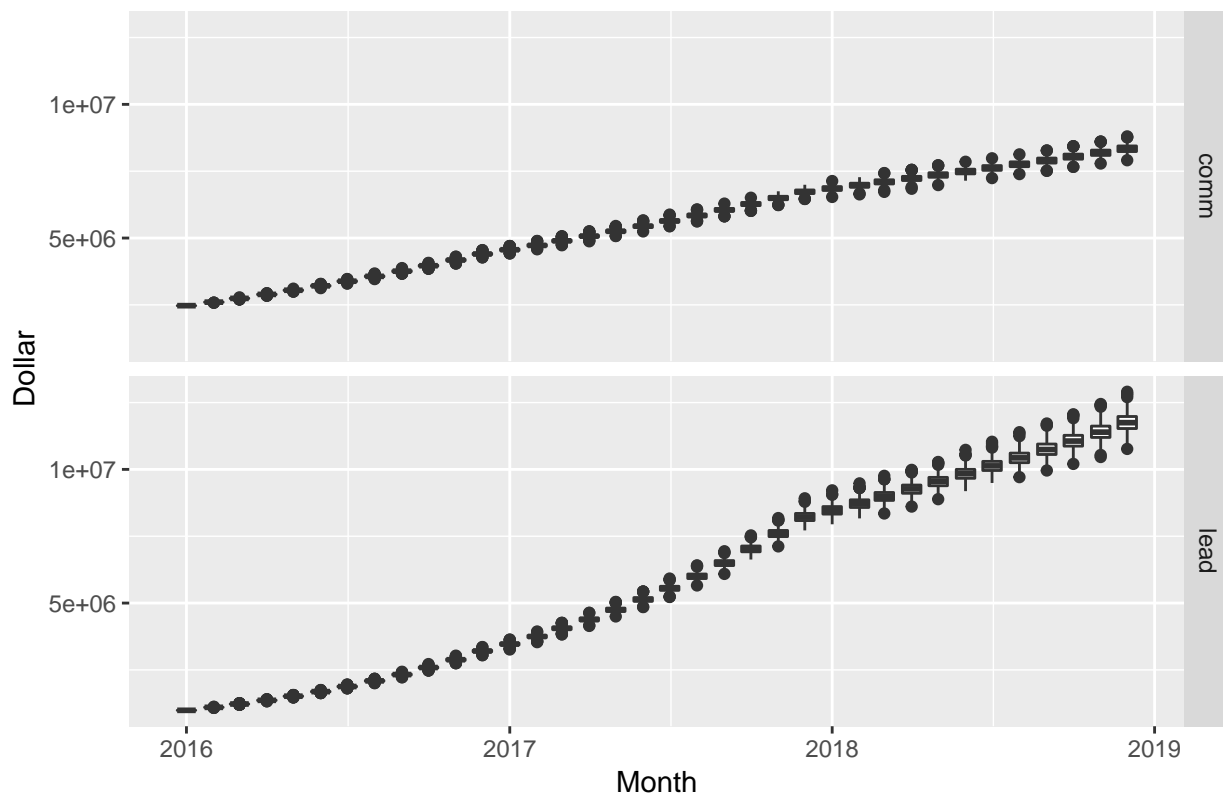
print(paste0("Community model: total revenue from Jan 2016 to Dec 2018: ", as.integer(mean(colMeans(total_comm_subs_comm_md1_df)))))

## [1] "Community model: total revenue from Jan 2016 to Dec 2018: 5502184"

q2_join <- rbind(total_lead_subs_lead_md1_df, total_comm_subs_comm_md1_df)
q2_join %>% ggplot(aes(x = ym, y = value, group = ym)) +
  geom_boxplot() +
  ylab('Dollar') +
  xlab('Month') +
  facet_grid(rows = vars(md1)) +
  ggtitle('Monthly revenue under both model (1000X simulation)')

```

Monthly revenue under both model (1000X simulation)

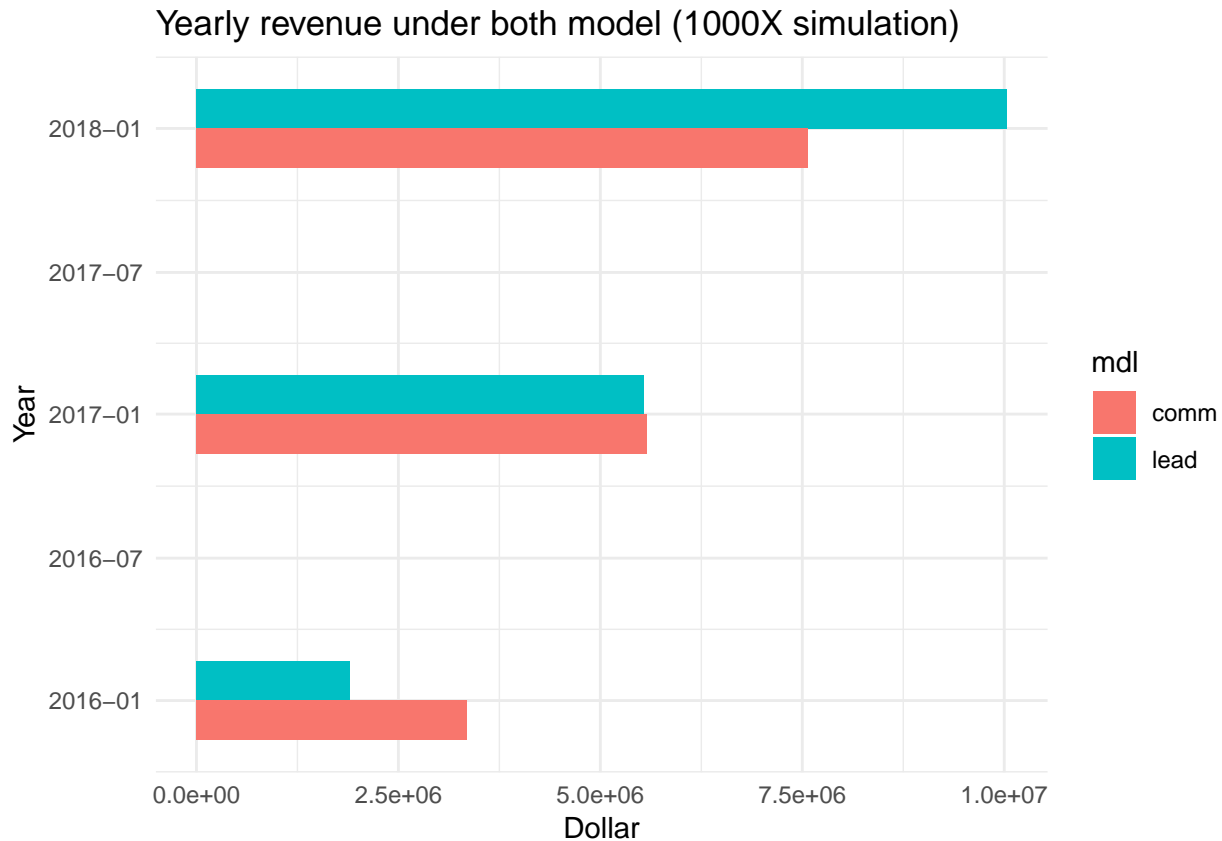


Based on 1000 simulation, the revenue of community model is larger than the lead model, but lead model out-performs community model during 2017. Below is a bar plot showing the yearly total revenue from two different models.

```
total_lead_subs_lead_md1_df <- total_lead_subs_lead_md1 %>% as.data.frame() %>%
  mutate(ym = make_date(year = rep(c(2016:2018), 1, each = 12), month = rep(c(1:12), 3, each = 1))) %>%
  mutate(yr = make_date(year = rep(c(2016:2018), 1, each = 12))) %>%
  melt(id = c('yr', 'ym')) %>%
  group_by(yr) %>%
  summarise(yearly_revenue = mean(value)) %>%
  mutate(md1 = 'lead')

total_comm_subs_comm_md1_df <- total_comm_subs_comm_md1 %>% as.data.frame() %>%
  mutate(ym = make_date(year = rep(c(2016:2018), 1, each = 12), month = rep(c(1:12), 3, each = 1))) %>%
  mutate(yr = make_date(year = rep(c(2016:2018), 1, each = 12))) %>%
  melt(id = c('yr', 'ym')) %>%
  group_by(yr) %>%
  summarise(yearly_revenue = mean(value)) %>%
  mutate(md1 = 'comm')

q2b_join <- rbind(total_lead_subs_lead_md1_df, total_comm_subs_comm_md1_df)
q2b_join %>% ggplot(aes(x = yr, y = yearly_revenue, fill = md1)) +
  geom_bar(position = 'dodge', stat = 'identity', width = 100) +
  ylab('Dollar') +
  xlab('Year') +
  ggtitle('Yearly revenue under both model (1000X simulation)') +
  theme_minimal() + coord_flip()
```



We also want to estimate the year after 2018 to see what the revenue looks like.

Calculate long-term revenue

```
### assume that the growth in the future is a random distribution from the old one
## yearly rate in the future
yearly_rate_future <- function( myrate){
  year <- sample(c(1, 2, 3), 1)
  new_rate <- rnorm(12, mean = myrate[year], sd = sd(myrate)/sqrt(36))
  return(new_rate)
}

cal_lead_rate_future <- function(initial_rate, myrate){
  year2019 <- initial_rate * cumprod(1 + yearly_rate_future( myrate))
  year2020 <- year2019[12] * cumprod(1 + yearly_rate_future(myrate))

  return(c(year2019, year2020))
}

initial_lead_per_comm_per_month <- mean(leads_monthly_leads_md1[c(36), ])

##
future1920_lead_rate <- replicate(100, cal_lead_rate_future(initial_lead_per_comm_per_month, leads_per_

calc_comm_future <- function(initial_num, myrate){
  year2019 <- initial_num * cumprod(1 + yearly_rate_future(myrate))
  year2020 <- year2019[12] * cumprod(1 + yearly_rate_future(myrate))
  #print(year2018[12])
```

```

    return(c(year2019, year2020))
}

dec2018_comm_lead_md1 <- mean(new_comm_monthly_lead_md1[36, ])
dec2018_comm_lead_md1

## [1] 23784.85
##
future1920_comm_lead_md1 <- replicate(100, calc_comm_future(dec2018_comm_lead_md1, comm_per_month_lead_md1))

## community mdl

dec2018_comm_comm_md1 <- mean(new_comm_monthly_comm_md1[36, ])
future1920_comm_comm_md1 <- replicate(100, calc_comm_future(dec2018_comm_comm_md1, comm_per_month_comm_md1))

## future revenue for both model
total_rev_lead_md1 <- future1920_lead_rate * future1920_comm_lead_md1 * 40
total_rev_comm_md1 <- future1920_comm_comm_md1 * 400

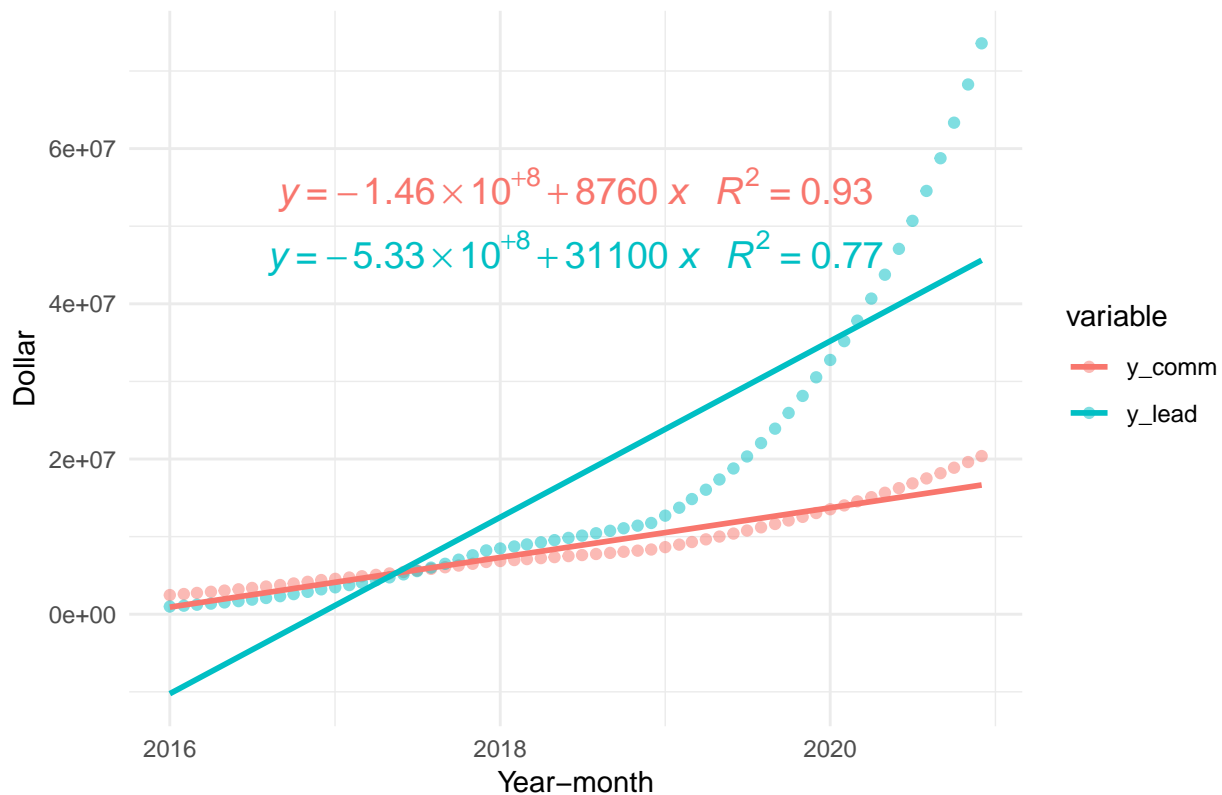
rev_df_lead_md1 <- data.frame(rbind(t(total_lead_subs_lead_md1)), t(total_rev_lead_md1))
rev_df_comm_md1 <- data.frame(rbind(t(total_comm_subs_comm_md1)), t(total_rev_comm_md1))
df_comm <- data.frame( x = c(1:60), y = colMeans(rev_df_comm_md1))
df_lead <- data.frame(x = c(1:60), y = colMeans(rev_df_lead_md1))

df_join <- df_comm %>% left_join(df_lead, by = 'x', suffix = c('_comm', '_lead' ))
df_join <- df_join %>% select('y_comm', 'y_lead') %>%
  mutate(ym = make_date(year = rep(c(2016:2020), 1, each = 12), month = rep(c(1:12), 3, each = 1))) %>%
  melt(id = 'ym')

fml <- y ~ x
df_join %>% ggplot(aes(x = ym, y = value, color = variable)) +
  geom_point(alpha = .5) +
  geom_smooth(method = "lm", formula = fml, se = F) +
  stat_poly_eq(aes(label = paste(..eq.label.., ..rr.label.., sep = "~~~")),
    label.x.npc = .5, label.y.npc = 0.7,
    formula = fml, parse = TRUE, size = 5) +
  ylab('Dollar') +
  xlab('Year-month') +
  ggtitle('Monthly revenue projection from 2016 - 2020 (1000X simulation)') +
  theme_minimal()

```

Monthly revenue projection from 2016 – 2020 (1000X simulation)



In conclusion, the revenue growth under lead model is faster than the growth under community model. For the next two years (2019-2020), this trend is estimated to be continue and the revenue gap will become bigger and bigger.

Q3: What strategic advantages do you believe each plan provides? What are the drawbacks of each?

For lead model

Pros:

- **Enriched content** The number of communities enrolled is larger than the number under community model (see plot below). This lead to more diverse content in the search result because there are more available communities to choose from.
- **Beneficial for small marketers** This pricing model incentivizes marketers for communities with low demand (less than 10 leads per month). For example, the willingness to pay of a marketer for a small community is less than 120\$ per month. This pricing scheme gives more benefits to him/her because the community is not going to get 10 leads per month.
- **Higher revenue** This model generates higher revenue in the long run.

Cons:

- **High fragility, high risk** This model has high fragility. The demand (leads received per community per month) for new constructed houses is volatile and will be affected by many factors such as macroeconomy, loan interest rate, consumer preferences, etc.

For community model

Pros:

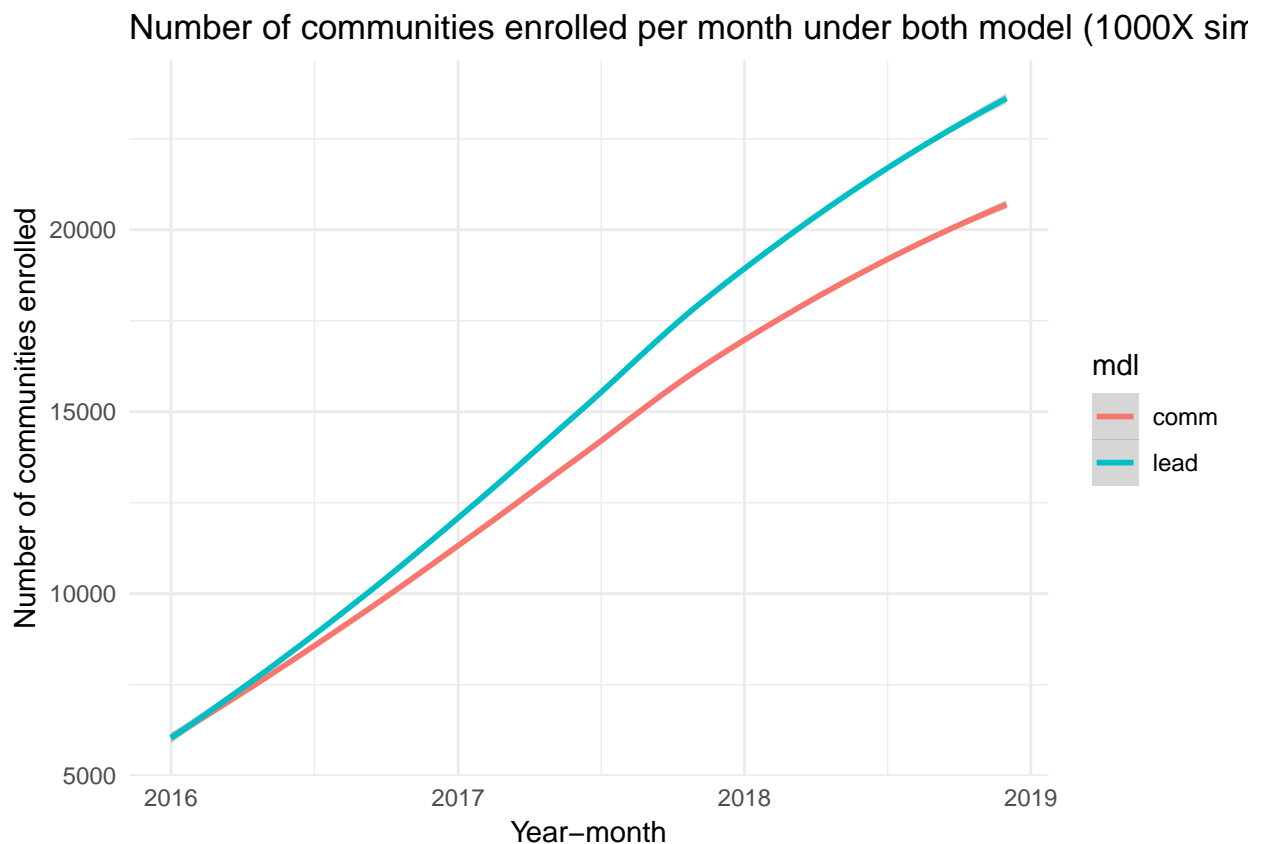
- **Stable revenue gain, low risk** The revenue gain is relatively stable because the pricing is not based on the demand but rather the supply.
- **Beneficial for big marketers** In contrast to the lead model, this model provides benefits for marketers of high-demand communities.

Cons:

- **Low revenue** This model generates less revenue in the long term.
- **Less enriched content** The number of communities enrolled is smaller, thus the search result is less enriched compared to the lead model.

```
## number of communities at the end of each year
q1_join %>% group_by(ym, mdl) %>%
  summarize(monthly_avg = mean(value)) %>%
  ggplot(aes(x = ym, y = monthly_avg, color = mdl)) +
  geom_smooth()+
  ylab('Number of communities enrolled') +
  xlab('Year-month') +
  ggtitle('Number of communities enrolled per month under both model (1000X simulation)') +
  theme_minimal()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Q4: Are there any additional concerns or opportunities about your model based on the information provided? Does anything seem unrealistic?

- 1. Churn rate of both model is not considered. Here we assume the marketers will not churn at any point once signed up, thus the number of communities grows steadily. The estimation of revenue seems to be overly optimistic. Below is an estimation from a consumer market point of view. The estimate revenue is only 27% of the revenue from the lead model in 2017.

data source

```
## 38, 000 number of new construction community in 2016
## Among 38, 000 new community constrcued in 2016, 25% were listed for sale in 2017
(24 + 41 + 154 + 75) * 1000/30 / 38000
```

```
## [1] 0.2578947
```

```
## Among 380000 in 2016, 50% were sold in 2017
(38 + 70 + 338 + 162) * 1000/30 / 38000
```

```
## [1] 0.5333333
```

```
## total new construction community 38000
```

```
## this is the estimated revenue in 2017 from the market
```

```
total_rev <- 38000*.5*.27 * 350000*1/100
```

```
total_rev_lead_model_2017 <- total_lead_subs_lead_md1[c(13:24), ] %>% colSums() %>% mean()
```

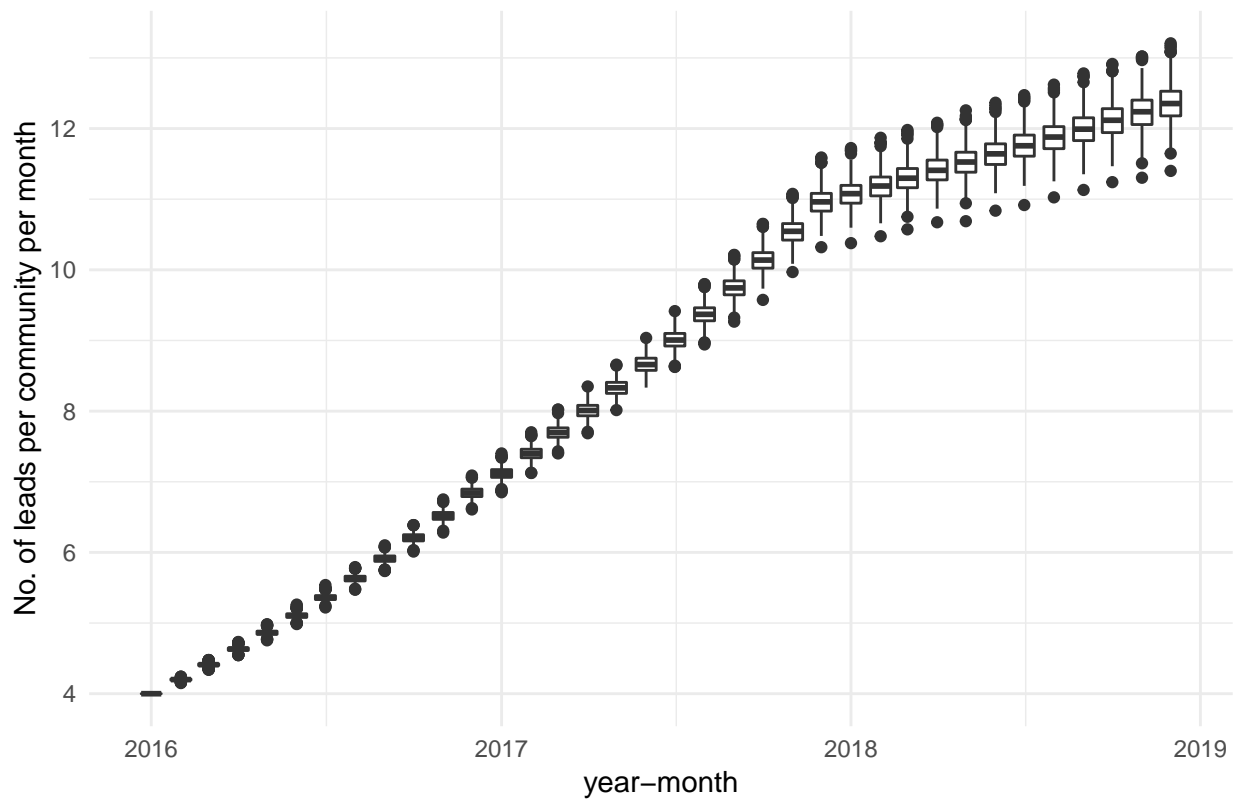
```
print(paste0('the ratio of the estimated revenue vs revenue in lead model for 2017: ', total_rev/total_
```

```
## [1] "the ratio of the estimated revenue vs revenue in lead model for 2017: 0.270217415391484"
```

- 2. Growth rate of lead per community per month may be over estimated. By the end of 2018, the number of lead delivered reaches to 12 per community per month. This seemes unrealistic and there should be some saturation parameter to control the ceiling of the lead number.

```
leads_montly_leads_md1 %>% as.data.frame() %>%
  mutate(ym = make_date(year = rep(c(2016:2018), 1, each = 12), month = rep(c(1:12), 3, each = 1))) %>%
  melt(id = c('ym')) %>% ggplot(aes(x = ym, y = value, group = ym)) + geom_boxplot() + theme_minimal() +
  xlab('year-month') +
  ylab('No. of leads per community per month') +
  ggtitle('Trend of no. of leads per community per month under lead model')
```

Trend of no. of leads per community per month under lead model



- 3. The pricing model for new construction community doesn't make to be a clean-cut solution. It can be a mixture of both community mode or provide both options for marketers.
 - For mixture model, we can provide such solution: charge each community less than 400 per month for leads under X per month, and charge extra Y amount of dollars for leads that exceed X.
 - Pricing option can be available based on some criteria of marketers. For example, depending on the duration of previous subscription, we can offer a lower rate or a different model.
- 4. External factors such as inflation rate, economic trend, price elasticity, price sensitivity are not taken into consider.