

# Симуляцийн аргууд хичээлийн бие даалтын ажил

## IIE/RA Contest Problem 3: Sally Model's SM Pizza Shop

Г.Махгал

МУИС – ХШУИС – Хэрэглээний Математикийн Тэнхим

᠑᠒ 2017/1/3



## Удиртгал

- ▶ IIE/RA Contest Problem 3: Sally Model's SM Pizza Shop бодлогыг авч үзэх болно.
- ▶ Бодлогыг програмчлалын R хэл<sup>1</sup> дээр бодсон.
- ▶ Хугацааны нэгжийг минутаар авсан.
- ▶ Энэхүү слайд болон бодлогын өгүүлбэр, холбогдох өгөгдөл, R хэл дээрх код зэргийг  
<https://github.com/galaamn/source-code-on-statistics/tree/master/Simulation/Example/Pizza%20Shop>  
хаягаар интернэтэд байрлуулсан.
- ▶ Цаашид зуухны хэсгийн загварчлалыг сайжруулах шаардлагатай.

---

<sup>1</sup>[https://en.wikipedia.org/wiki/R\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/R_(programming_language))



# Бодлого

Пицца хийж борлуулдаг дэлгүүр

## ▶ Өгөгдөхүүн

- ▶ Ихдээ 3 хүн ажиллах боломжтой пицца бэлдэх ширээ
- ▶ Гурван төрлийн сонголттой<sup>2</sup> нэг зуух
- ▶ Хүргэлтийн ажилчид

## ▶ Бусад

- ▶ Захиалгын эрчим
- ▶ Захиалга дахь пиццаны тоо болон хэмжээ тус бүрийн тархалтууд<sup>3</sup>
- ▶ Пицца бэлдэх ажлын шат дамжлагууд, тэдгээрийн үргэлжлэх хугацааны тархалт
- ▶ Пиццаг хэрчиж савлахад зарцуулагдах хугацааны тархалт
- ▶ Хүргэлтийн хугацааны тархалт

## ▶ Олох зүйл

- ▶ Пицца бэлдэх ширээнд ажиллуулах хүний тоо
- ▶ Зуухны төрөл
- ▶ Пицца хүргэлтэнд ажиллах хүний тоо

---

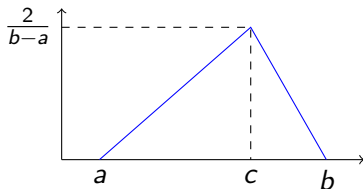
<sup>2</sup> оруулах хэсгийн багтаамжаараа ялгагдана

<sup>3</sup> эдгээр хувьсагчид дээр хамаарлын талаар ямар нэг зүйл дурдагдаагүй байна



# Санамсаргүй хувьсагчийн загварчлал

Гурвалжин тархалт<sup>4</sup>



$$F(x) = \begin{cases} 0 & x \leq a \\ \frac{(x-a)^2}{(b-a)(c-a)} & a < x \leq c \\ 1 - \frac{(b-x)^2}{(b-a)(b-c)} & c < x < b \\ 1 & b \leq x \end{cases}$$

Урвуу хувиргалтын аргаар

```
rtriangular = function (n, a, c, b) {  
  ifelse(  
    (u = runif(n)) < (c-a)/(b-a),  
    a + sqrt((b-a)*(c-a)*u),  
    b - sqrt((b-a)*(b-c)*(1-u))  
  )  
}
```

<sup>4</sup>[https://en.wikipedia.org/wiki/Triangular\\_distribution](https://en.wikipedia.org/wiki/Triangular_distribution)



# Санамсаргүй хувьсагчийн загварчлал

## Бусад

- ▶ Жигд тархалт буюу санамсаргүй тоо  
R програмын stats багц дахь `runif()` функц
- ▶ Тархалтуудын холимог<sup>5</sup>  
Хоёр ширхэг гурвалжин тархалтын холимог<sup>6</sup>
- ▶ Илтгэгч тархалт  
R програмын stats багц дахь `rexp()` функц<sup>7</sup>

---

<sup>5</sup> composition method

<sup>6</sup> хэрчиж савлах хэсэгт

<sup>7</sup> захиалга хоорондын хугацааг загварчлахад ашиглагдана



# Загварчлалын үе шатууд

1. Захиалга орж ирэх хугацааны эгшинг загварчлах  
захиалгын эрчимд тохируулан захиалга хоорондын хугацааг тодорхойлно
2. Захиалгын зарим мэдээлэл  
пиццаны тоо ба хэмжээ, захиалгын төрөл (delivery эсвэл carry-out), хүргэлтийн хугацаа
3. Пиццаны зарим мэдээлэл  
пицца нэг бүрийн хэмжээ болон хийхэд зарцуулагдах хугацаа (3 үе шат нэг бүрчлэн)
4. Пиццаг бэлдэх ажлыг загварчлах  
пицца хийх ширээнд ажиллах хүний тооноос (1-ээс 3 хүртэл) хамаарна
5. Зуухны хэсгийг загварчлах  
пицца нэг бүрийн зуухны оруулах хэсэгт нэвтрэх эгшин болон зуухны гаргах хэсэгт шилжих эгшин
6. Хэрчиж савлах, захиалгыг гаргах хэсгийг загварчлах  
IIE\_SM\_3.dat файл дахь өгөгдлийг шинжилж улмаар зохих тархалтын тусламжтайгаар энэ хэсэгт зарцуулагдах хугацааг тодорхойлно
7. Хүргэлтийн хэсгийг загварчлах  
захиалгыг хүргэхэд зарцуулагдах хугацааг тодорхойлно, хүргэлтийн ажилчдын тоо хувьсах боломжтой байх шаардлагатай
8. Захиалгыг гүйцэтгэхэд зарцуулсан нийт хугацааг олох



### ► Дундаж

```
the.number.of.demands = c()  
for (i in 1:1000) {  
  the.number.of.demands = c(the.number.of.demands,  
    length(simulate.demands()))  
}  
mean(the.number.of.demands)
```

```
120.303
```

энд `simulate.demands()` – захиалгыг загварчлах функц

### ► Стандарт хазайлт

```
sd(the.number.of.demands)
```

```
10.86505
```



# Үр дүн

Захиалгыг бүрэн гүйцэтгэх хугацааны зарим үзүүлэлт, параметрийн зарим утганд

## Ашиглагдах код

```
elapsed.time = demand.type = c()
for (i in 1:1000) {
  demands = simulate.demands()
  demands = simulate.count.and.size(demands)
  pizzas = simulate.pizza.making(demands)
  pizzas = switch(the.number.of.person.at.make.table,
    pizza.making.one.person(), pizza.making.two.person(),
    pizza.making.three.person())
  oven(); cut.and.box(); delivery()
  demands$elapsed.time = demands[["delivery.time"]] -
    demands[["time"]]
  elapsed.time = c(elapsed.time, demands[["elapsed.time"]])
  demand.type = c(demand.type, demands[["type"]])
}
```








# Үр дүн

Захиалгыг бүрэн гүйцэтгэх хугацааны зарим үзүүлэлт, параметрийн зарим утганд

## Ашиглагдах код

```
tapply(elapsed.time, demand.type, mean)
tapply(elapsed.time, demand.type, range)
```

 <sup>8</sup>	 <sup>9</sup>	 <sup>10</sup>	delivery			carry-out		
			mean	min	max	mean	min	max
1	3	7	115.02	16.31	299.16	108.42	12.59	289.28
2	3	7	39.57	15.97	112.57	32.06	12.35	101.92
3	1	5	46.95	15.97	149.52	23.36	12.40	74.67
3	1	6	37.09	16.16	113.10	23.56	12.38	75.37
3	1	7	32.82	15.91	86.42	23.43	12.37	71.54
3	3	7	31.12	16.37	79.40	21.47	12.40	69.97
3	3	8	28.96	16.04	67.95	21.24	12.38	63.03

<sup>8</sup>Ширээн дэх ажилчдын тоо

<sup>9</sup>Зуухны төрөл

<sup>10</sup>Хүргэгчдийн тоо



## Үр дүн

Захиалгыг бүрэн гүйцэтгэх хугацааны зарим үзүүлэлт, параметрийн 3, 1, 7 утганд

### Ашиглагдах код

```
tapply(elapsed.time, demand.type, quantile, probs = 0.9)  
boxplot(formula = elapsed.time ~ demand.type)
```

### Үр дүн

```
47.17 # delivery  
35.12 # carry-out
```

