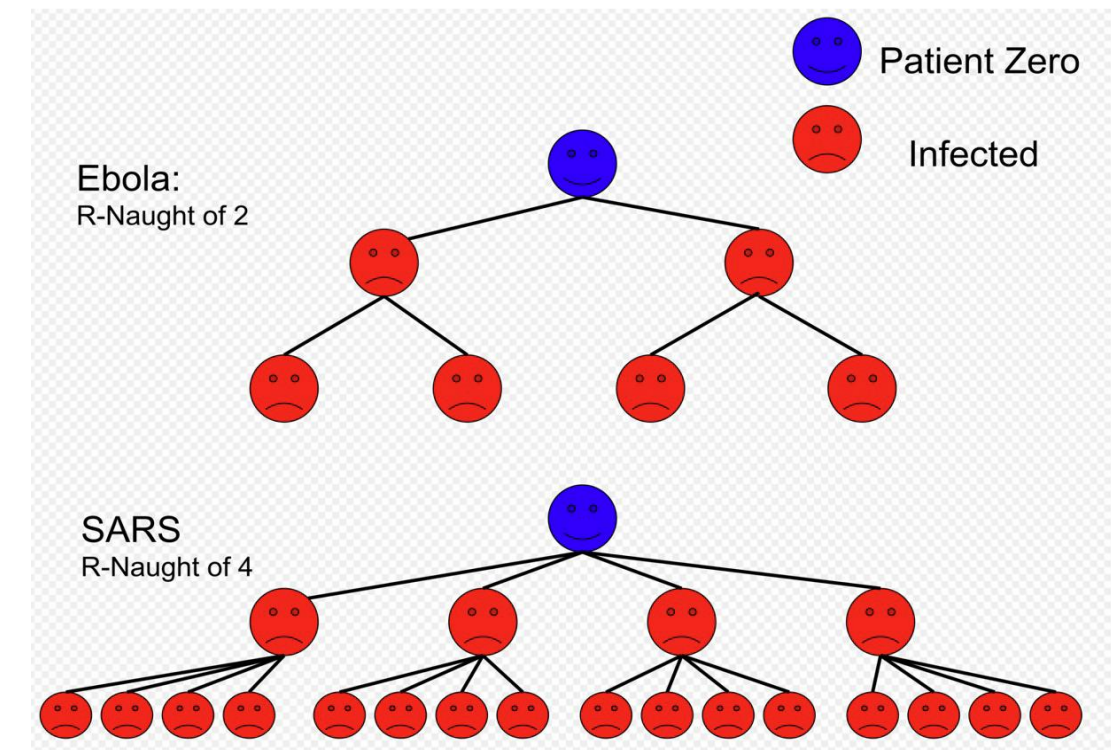


감염역학 지표 추정

가톨릭대학교 예방의학과
남유경

감염재생산지수 R

- 감염병 전파 양상을 나타내는 척도로, **감염재생산지수 R**과 **과대산포모수 K**를 주로 사용
- **감염재생산지수 (R)**
: 감염된 환자 1명이 질병을 전염시킬 수 있는 **평균 사람 수**
- **감염재생산지수의 한계점 (“평균의 함정”)**
: 점염의 이질성과 변동성을 나타내지 못함



=> 감염재생산지수 외에도 개인간 변동을 나타내는 지표(과대산포모수 K)가 필요

과대산포모수 k

- 과대산포모수 (Overdispersion parameter, k 값)

- : 전염의 이질성과 변동성을 나타내는 지표

- : k 값을 통해 수퍼전파능(SSE)의 가능성을 탐색할 수 있음

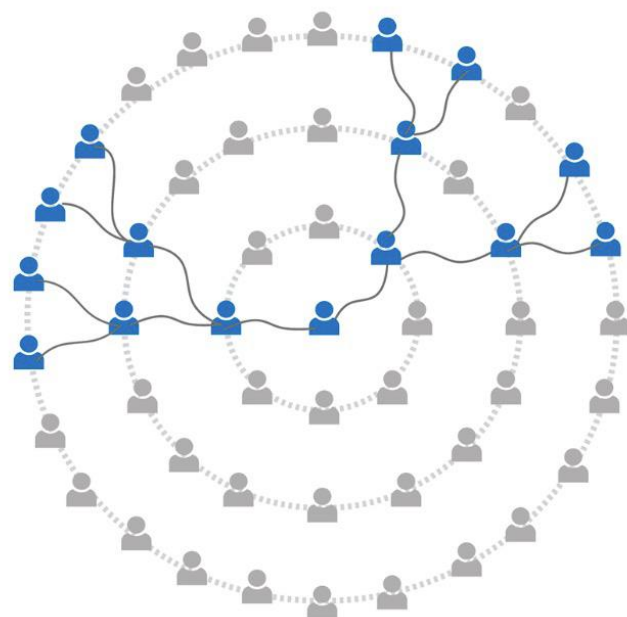
- 감염병 전염 가능성의 개인간 변동을 수치화하는 지표

- : $k \uparrow$ 이면 전염성의 변동성 \downarrow (SSE 출현 가능성 \downarrow , 2차 감염자의 분포가 **포아송 분포**를 따름)

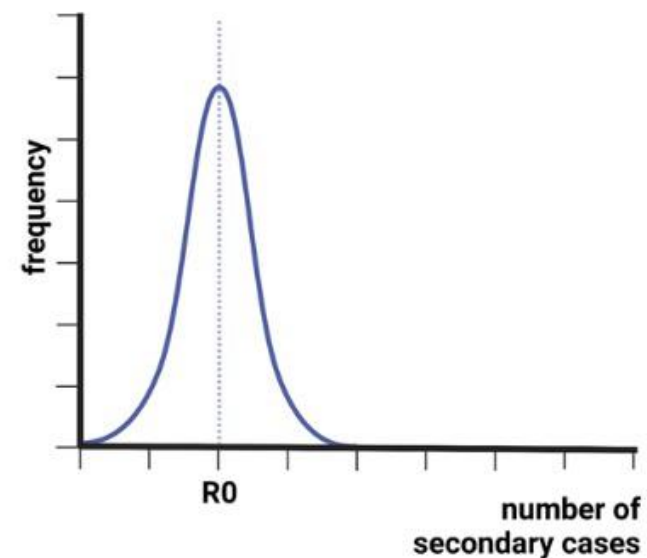
- : $k \downarrow$ 이면 전염성의 변동성 \uparrow (SSE 출현 가능성 \uparrow , 2차 감염자의 분포가 **음이항 분포**를 따름)

A More homogeneous spread

$R=2.0$; $1 < k < \infty$

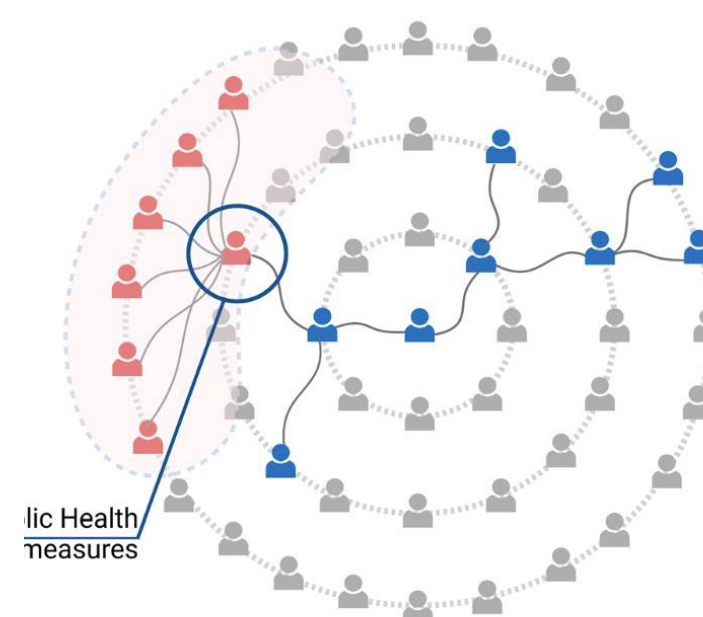


k approaching infinity: Poisson distribution with mean and variance equal to R_0

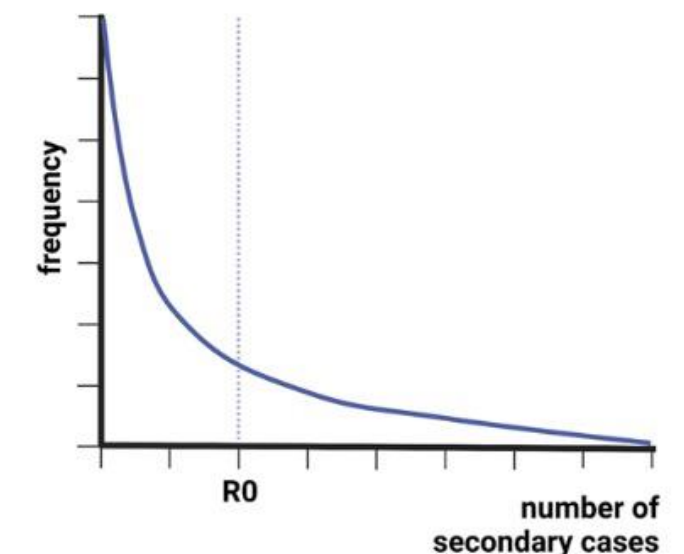


B More heterogeneous spread

$R=2.0$ $0 < k < 1$



$0 < k < 1$: Negative Binomial distribution with mean R_0 and fat tailed-end



생산분포 (Offspring distribution)

- 생산분포 (Offspring distribution)

: 각 감염자로부터 발생된 2차 감염자 수(Z)를 나타내는 확률분포

: 대표적인 생산분포는 포아송분포와 음이항분포

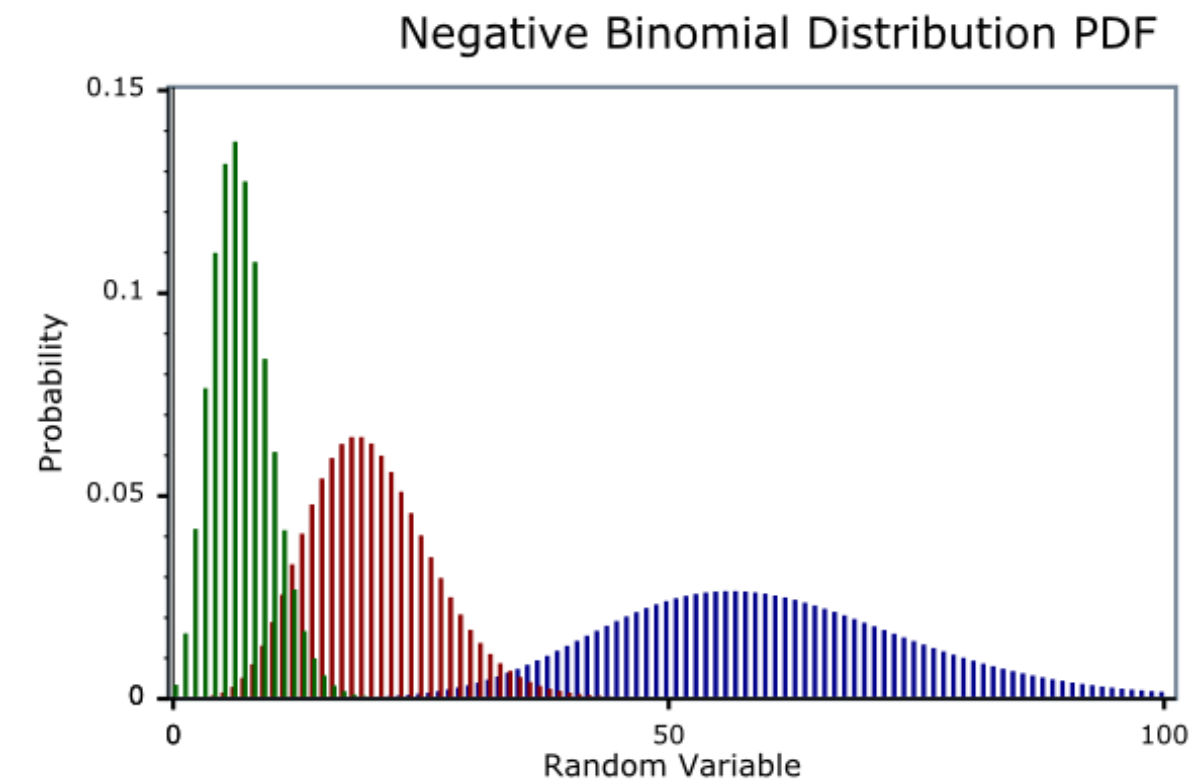
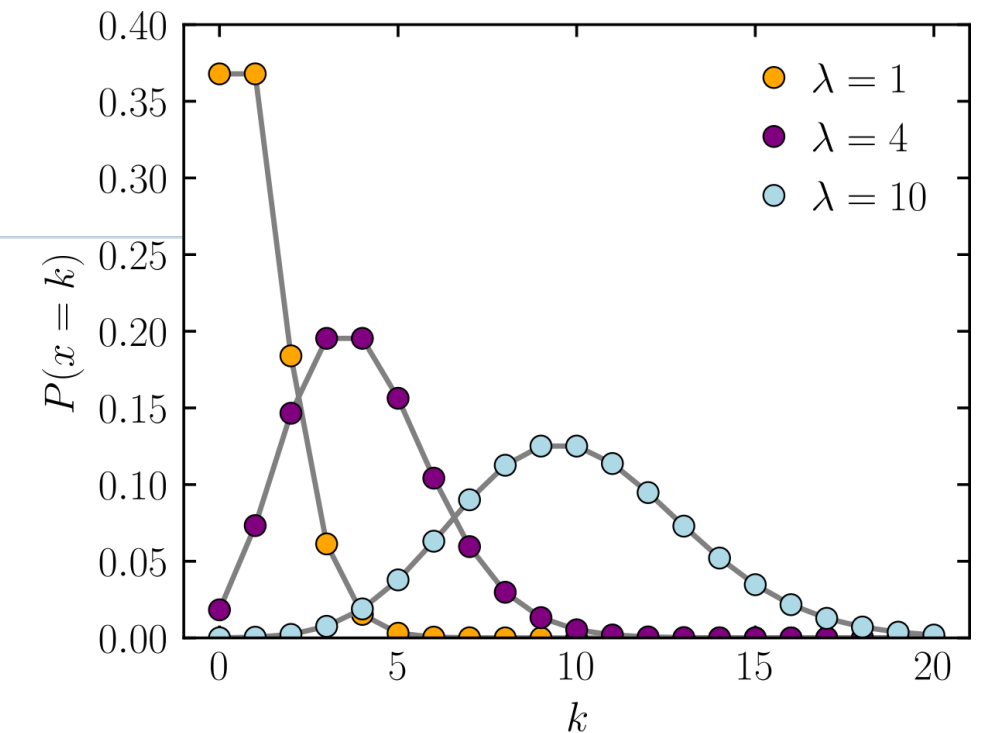
- 포아송 분포 $Z \sim \text{Poisson}(R)$

: 한 감염자로부터 발생된 2차 감염자 수가 z 일 확률은 다음과 같음

$$P(Z = z) = \frac{R^z e^{-R}}{z!}, \quad z = 0, 1, 2, \dots$$

- 음이항분포 $Z \sim \text{NB}(R, k)$

$$P(Z = z) = \frac{\Gamma(k + z)}{z! \Gamma(k)} \left(\frac{R}{R + k} \right)^z \left(1 + \frac{R}{k} \right)^{-k}, \quad z = 0, 1, 2, \dots$$



실제 데이터 적용

- 2024년 4월에 부산 영동구 체중고에서 발생한 백일해 데이터 ($n = 48$)
- 데이터에 포아송 분포 및 음이항 분포를 적합
- 각 모형의 감염재생산지수 및 과대산포모수 (R, k) 추정

데이터 설명

	A	B
1	key	SOI
2	1	7
3	2	19
4	3	1
5	4	1
6	5	1
7	6	1
8	7	
9	8	1
10	9	19
11	10	2
12	11	1
13	12	1
14	13	10
15	14	18

- 백일해 데이터 (n= 48)
- Key : 고유식별 아이디, 개별 환자 식별에 사용
- SOI : Source of Infection, 감염원의 고유식별 아이디


데이터 불러오기 및 전처리

```
1
2 # 라이브러리 불러오기
3 library(readxl)
4 library(dplyr)
5 library(tidyr)
6 library(fitdistrplus)
7 library(ggplot2)
8
9
10 # 패키지 설치
11 # install.packages("readxl")
12 # install.packages("dplyr")
13 # install.packages("tidyr")
14 # install.packages("fitdistrplus")
15 # install.packages("ggplot2")
16
17
18
19 #전체 데이터 불러오기
20 linelist <- read_excel("백일해데이터.xlsx")
21
22
23 #source of infection, infectee 변수 선택 및 변수명 변경
24 overall_data <- linelist %>% rename(infectior.case = "SOI", infectee.case = "key")
25
26
```

```
> overall_data
# A tibble: 48 × 2
  infectee.case infectior.case
      <dbl>         <dbl>
1           1             7
2           2            19
3           3             1
4           4             1
5           5             1
6           6             1
7           7            NA
8           8             1
9           9            19
10          10             2
# i 38 more rows
# i Use `print(n = ...)` to see more rows
```

데이터를 count data로 변환

```
> overall_data
# A tibble: 48 × 2
  infectee.case infector.case
    <dbl>         <dbl>
1         1         7
2         2        19
3         3         1
4         4         1
5         5         1
6         6         1
7         7        NA
8         8         1
9         9        19
10        10         2
```



```
> complete_offspringd
# A tibble: 49 × 2
  name value
  <int> <dbl>
1     1    10
2     2     9
3     3     6
4     4     5
5     5     4
6     6     3
7     7     2
8     8     2
9     9     1
10    10     1
```

- 특정 환자(infector.case)가 누구(infectee.case)에게 감염을 시켰는지에 대한 관계를 보여주는 데이터
- 특정환자(name)가 몇 명(value)에게 감염을 시켰는지에 대한 데이터로 변환

데이터를 count data로 변환

- 특정 환자(infector.case)가 몇 명(n)에게 감염을 전파했는지에 대한 데이터 생성

```
> overall_data
# A tibble: 48 × 2
  infectee.case infector.case
      <dbl>         <dbl>
1           1             7
2           2            19
3           3             1
4           4             1
5           5             1
6           6             1
7           7            NA
8           8             1
9           9            19
10          10             2
```



```
> offspring
# A tibble: 14 × 2
# Groups:   infector.case [14]
  infector.case     n
      <dbl> <int>
1           19     10
2            1      9
3            2      6
4           10      5
5           18      3
6            7      2
7           14      2
8           12      1
9           15      1
10          17      1
11          20      1
12          40      1
13          43      1
14          44      1
```

```
# infector.case 별로 감염시킨 사람의 수를 계산
# infector.case가 몇번 등장하는지
offspring <- overall_data %>%
  dplyr::select(infector.case) %>%
  drop_na() %>%
  group_by(infector.case) %>%
  count() %>%
  arrange(desc(n))
```

데이터를 count data로 변환

- 아무에게도 감염시키지 않은 환자수 계산

```
#감염된 사람들의 정보를 추출
infectee <- overall_data %>%
  dplyr::select(infectee.case, infectee.case) %>%
  gather() %>%
  filter(key == 'infectee.case')
```

```
#감염시킨 사람들만 추출
infector <- overall_data %>%
  dplyr::select(infector.case, infectee.case) %>%
  gather() %>%
  filter(key == 'infector.case')
```

```
#감염시킨 사람과 감염된 사람의 중복되는 정보
duplicate <- infector %>%
  left_join(., infectee, by = 'value', relationship = "many-to-many") %>%
  filter(!is.na(key.y)) %>%
  dplyr::select(value) %>%
  distinct()
```

```
#아무에게도 감염시키지 않은 환자수 계산
nterminal_infectees <- infectee %>%
  dplyr::select(value) %>%
  filter(!value %in% duplicate$value) %>%
  transmute(case.no = as.numeric(value)) %>%
  nrow()
```

```
> infectee
# A tibble: 48 × 2
  key      value
  <chr>    <dbl>
1 infectee.case    1
2 infectee.case    2
3 infectee.case    3
4 infectee.case    4
5 infectee.case    5
6 infectee.case    6
7 infectee.case    7
8 infectee.case    8
9 infectee.case    9
10 infectee.case   10
# i 38 more rows
# i Use `print(n = ...)` to see more rows
```

```
> infector
# A tibble: 48 × 2
  key      value
  <chr>    <dbl>
1 infector.case    7
2 infector.case   19
3 infector.case    1
4 infector.case    1
5 infector.case    1
6 infector.case    1
7 infector.case   NA
8 infector.case    1
9 infector.case   19
10 infector.case    2
# i 38 more rows
# i Use `print(n = ...)` to see more rows
```

```
> duplicate
# A tibble: 14 × 1
  value
  <dbl>
1     7
2    19
3     1
4     2
5    10
6    18
7    14
8    20
9    17
10    12
11    15
12    40
13    44
14    43
```

```
> nterminal_infectees
[1] 34
```

데이터를 count data로 변환

- 특정 환자(infector.case)가 몇 명(n)에게 감염을 전파했는지에 대한 데이터와 아무에게도 감염시키지 않은 환자수 데이터 병합

```
offspring_counts <- enframe(c(offspring$n, rep(0, nterminal_infectees))) %>% pull(value)
```

```
> offspring
# A tibble: 14 × 2
# Groups:   infector.case [14]
  infector.case     n
    <dbl> <int>
1         19     10
2          1      9
3          2      6
4         10      5
5         18      3
6          7      2
7         14      2
8         12      1
9         15      1
10        17      1
11        20      1
12        40      1
13        43      1
14        44      1
```

```
> nterminal_infectees
[1] 34
```

```
> offspring_counts
[1] 10  9  6  5  3  2  2  1  1  1  1  1  1  1  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
[36]  0  0  0  0  0  0  0  0  0  0  0  0  0  0
```

포아송 분포 적합

```
#POI 분포 적합
pfit <- offspring_counts %>% fitdist(., distr = 'pois')
#결과 확인
summary(pfit)
```

```
> summary(pfit)
Fitting of the distribution ' pois ' by maximum likelihood
Parameters :
      estimate Std. Error
lambda 0.9166667 0.1381925
Loglikelihood: -90.27954 AIC: 182.5591 BIC: 184.4303
```

- 포아송 분포의 λ = 감염재생산수 R
- 음이항분포의 μ = 감염재생산수 R,
size = 과대산포모수 K

음이항 분포 적합

```
#NB 분포 적합
nbfit <- offspring_counts %>% fitdist(., distr = 'nbinom')

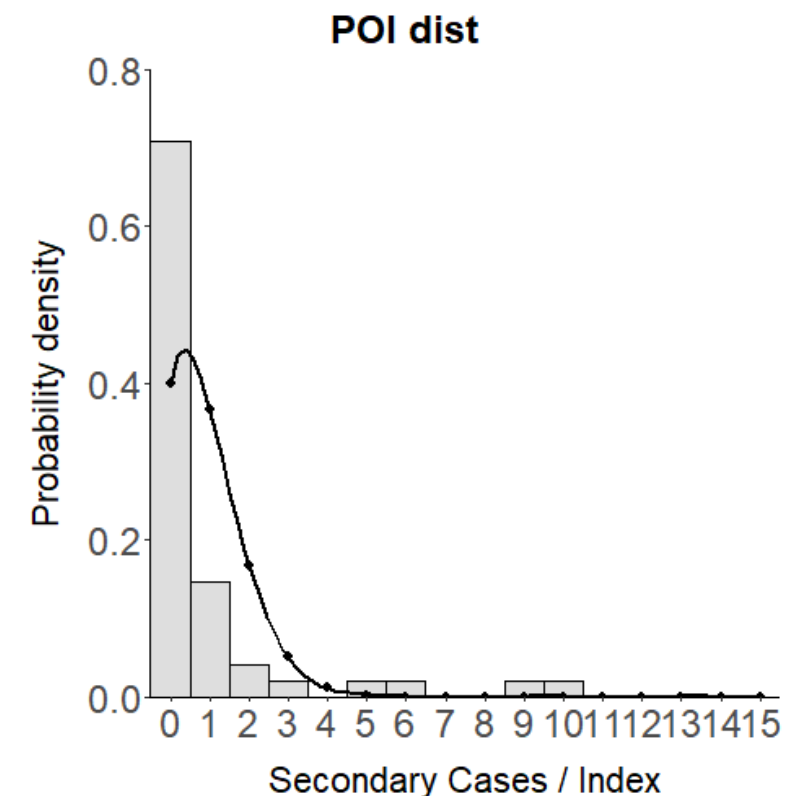
#결과 확인
summary(nbfit)
```

```
> summary(nbfit)
Fitting of the distribution ' nbinom ' by maximum likelihood
Parameters :
      estimate Std. Error
size 0.1957303 0.07666012
mu    0.9168151 0.32954369
Loglikelihood: -55.59319   AIC:  115.1864   BIC:  118.9288
Correlation matrix:
              size              mu
size 1.0000000000 -0.0001448425
mu   -0.0001448425 1.0000000000
```

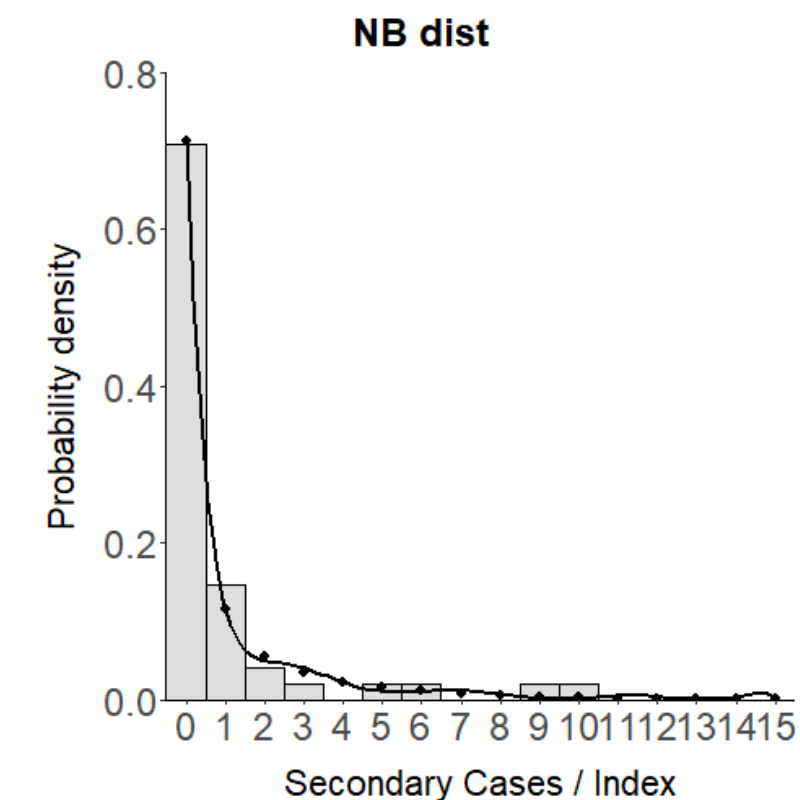
- 음이항분포의 μ = 감염재생산수 R ,
size = 과대산포모수 k
- $R = 0.92$: 한 명의 환자가 평균적으로 0.92명에게 전파
- $K = 0.2$: k 값이 작을수록 변동성이 높은것으로 해석

적합 플롯 만들기

```
ggplot() +
  geom_histogram(aes(x = complete_offspringd$value, y = ..density..),
    fill = "gray", colour = "black", binwidth = 1, alpha = 0.5) +
  geom_point(aes(x = 0:15, y = dpois(x = 0:15, lambda = pfit$estimate[[1]])), size = 2) +
  stat_smooth(aes(x = 0:15, y = dpois(x = 0:15, lambda = pfit$estimate[[1]])),
    method = 'lm', formula = y ~ poly(x, 9), se = FALSE, size = 0.8, colour = 'black') +
  expand_limits(x = 0, y = 0) +
  scale_x_continuous("Secondary Cases / Index", expand = c(0, 0), breaks = 0:15) +
  scale_y_continuous("Probability density", limits = c(0, 0.8), expand = c(0, 0)) +
  ggtitle("POI dist") +
  theme_classic() +
  theme(
    axis.line = element_line(size = 0.7),
    axis.text = element_text(size = 20),
    axis.title = element_text(size = 18),
    aspect.ratio = 1,
    axis.title.y = element_text(margin = margin(r = 12)),
    axis.title.x = element_text(margin = margin(t = 10)),
    plot.margin = margin(t = 50, r = 10, b = 10, l = 10),
    plot.title = element_text(size = 20, face = "bold", hjust = 0.5, vjust = 2), # 제목 크기 키우고 중앙 정렬
    plot.title.position = "plot"
  )
```



```
#음 이항분포
ggplot() +
  geom_histogram(aes(x = complete_offspringd$value, y = ..density..), fill = "gray", colour = "black", binwidth = 1, alpha = 0.5) +
  geom_point(aes(x = 0:15, y = dnbinom(x = 0:15, size = nbfit$estimate[[1]], mu = nbfit$estimate[[2]])), size = 2) +
  stat_smooth(aes(x = 0:15, y = dnbinom(x = 0:15, size = nbfit$estimate[[1]], mu = nbfit$estimate[[2]])),
    method = 'lm', formula = y ~ poly(x, 9), se = FALSE, size = 0.8, colour = 'black') +
  expand_limits(x = 0, y = 0) +
  scale_x_continuous("Secondary Cases / Index", expand = c(0, 0), breaks = 0:15) +
  scale_y_continuous("Probability density", limits = c(0, 0.8), expand = c(0, 0)) +
  ggtitle("NB dist") +
  theme_classic() +
  theme(
    axis.line = element_line(size = 0.7), # 축의 두께 조절
    axis.text = element_text(size = 20), # 축 텍스트 크기 조절
    axis.title = element_text(size = 18), # 축 제목 크기 조절
    aspect.ratio = 1,
    axis.title.y = element_text(margin = margin(r = 12)),
    axis.title.x = element_text(margin = margin(t = 10)),
    plot.margin = margin(t = 50, r = 10, b = 10, l = 10), # 상단 여백 조정
    plot.title = element_text(size = 20, face = "bold", hjust = 0.5, vjust = 2), # 제목 크기 키우고 중앙 정렬
    plot.title.position = "plot" # 제목을 플롯의 맨 위에 배치
  )
```



감사합니다.