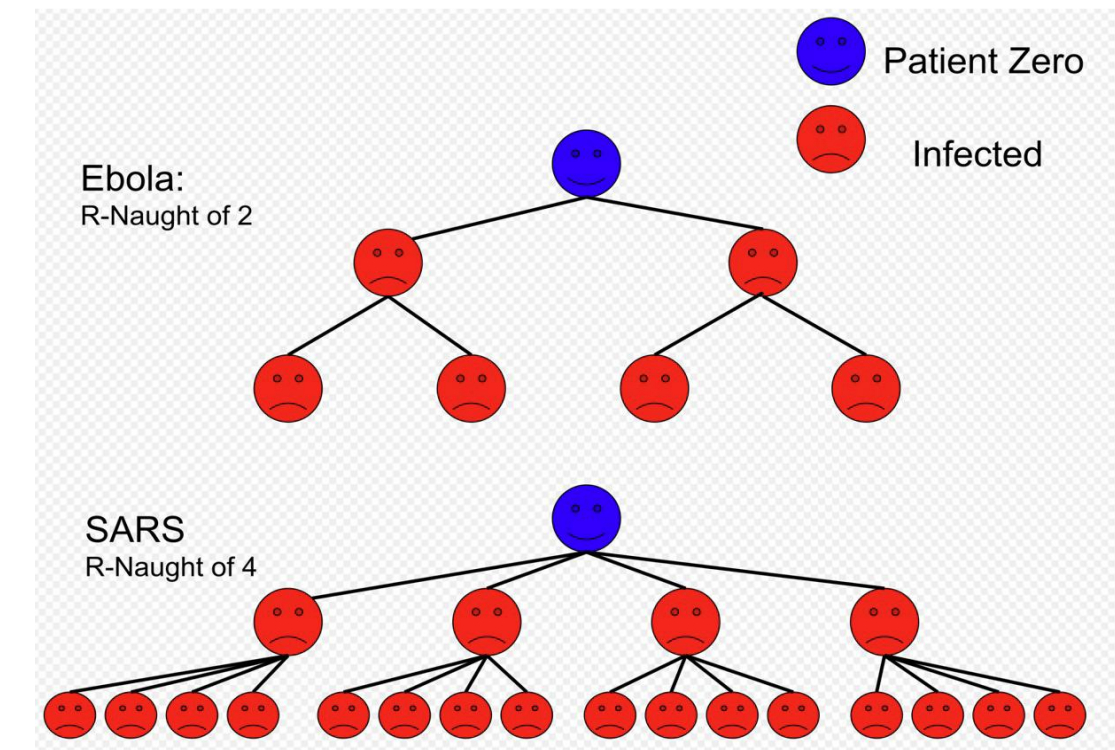


감염역학 지표 추정

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

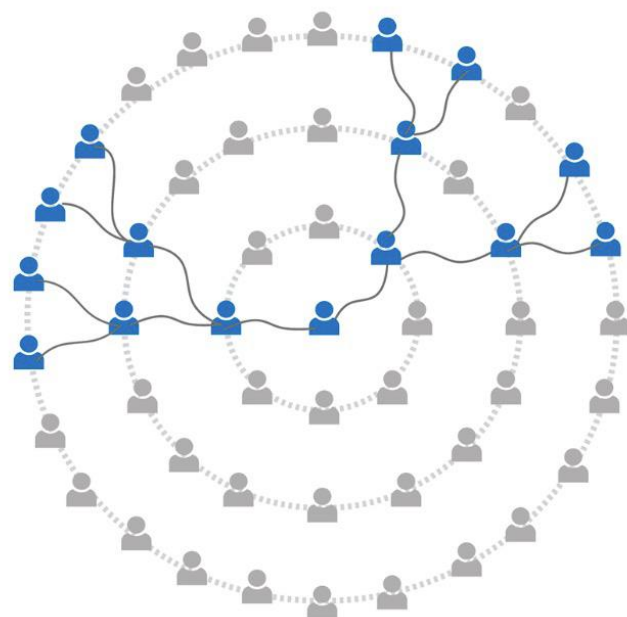
- 감염병 전염 가능성의 개인간 변동을 수치화하는 지표

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

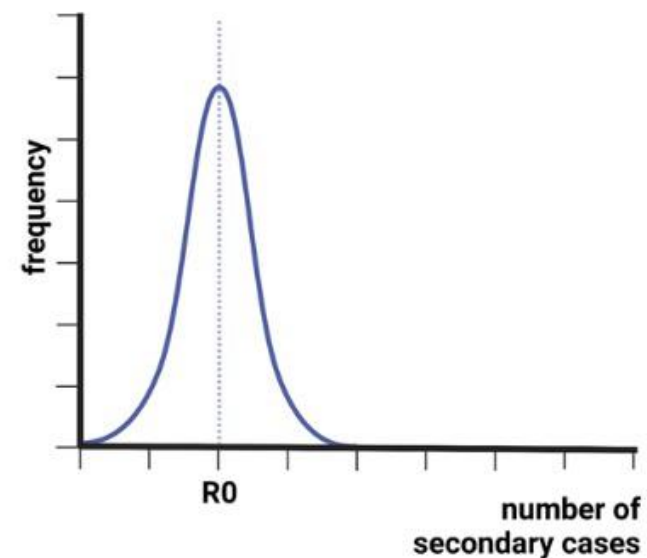
- : $k \downarrow$ 이면 전염성의 변동성 \uparrow (SSE 출현 가능성 \uparrow , 2차 감염자의 분포가 **음이항 분포**를 따름)

A More homogeneous spread

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

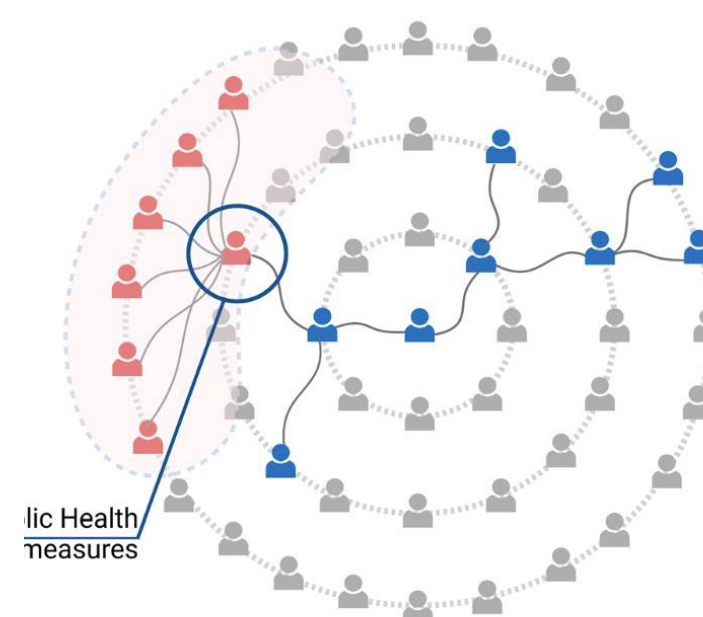


k approaching infinity: Poisson distribution with mean and variance equal to R_0

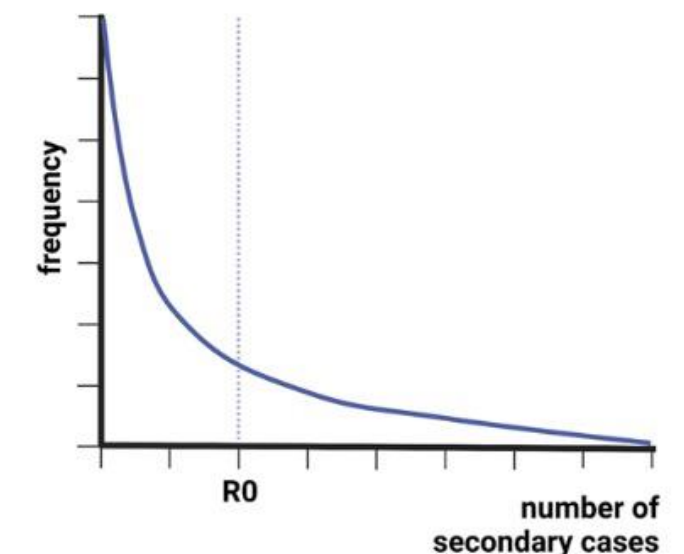


B More heterogeneous spread

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



$0 < k < 1$: 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

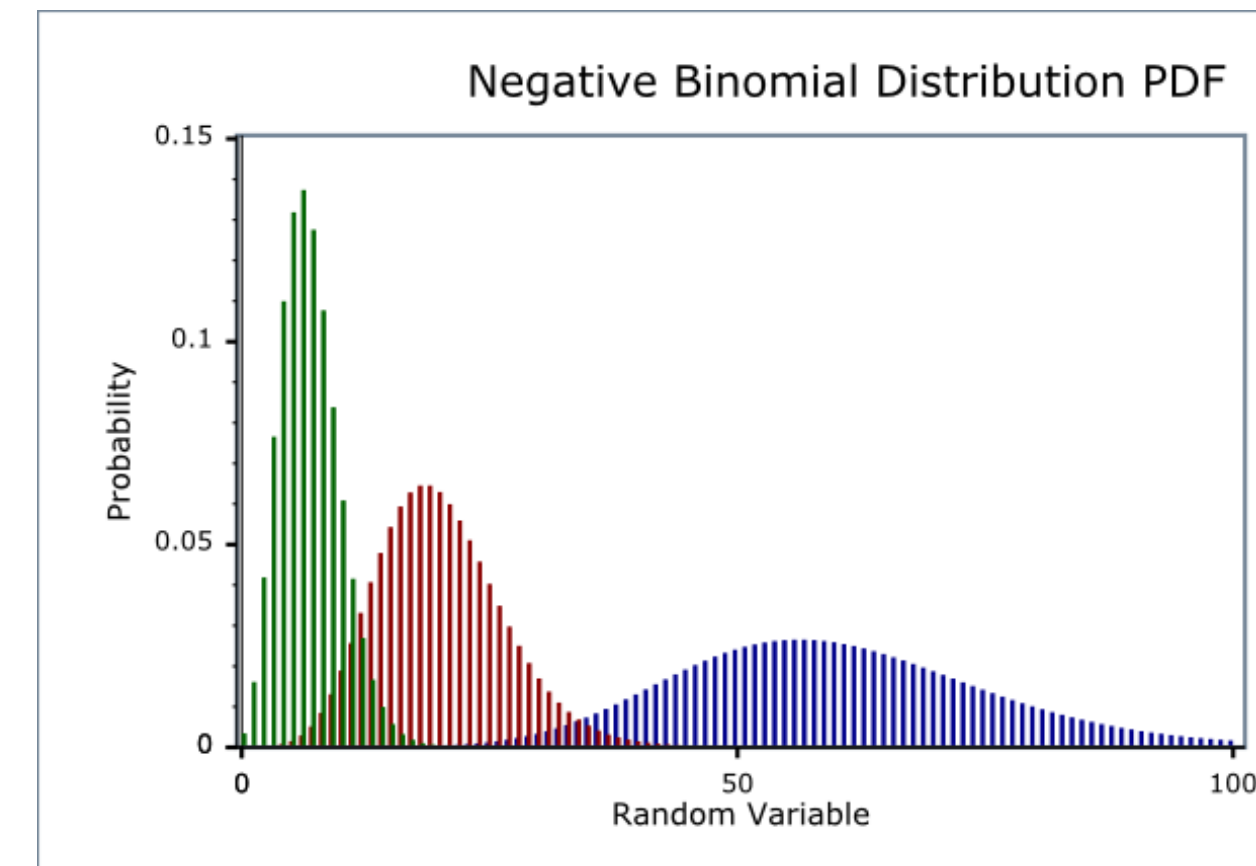
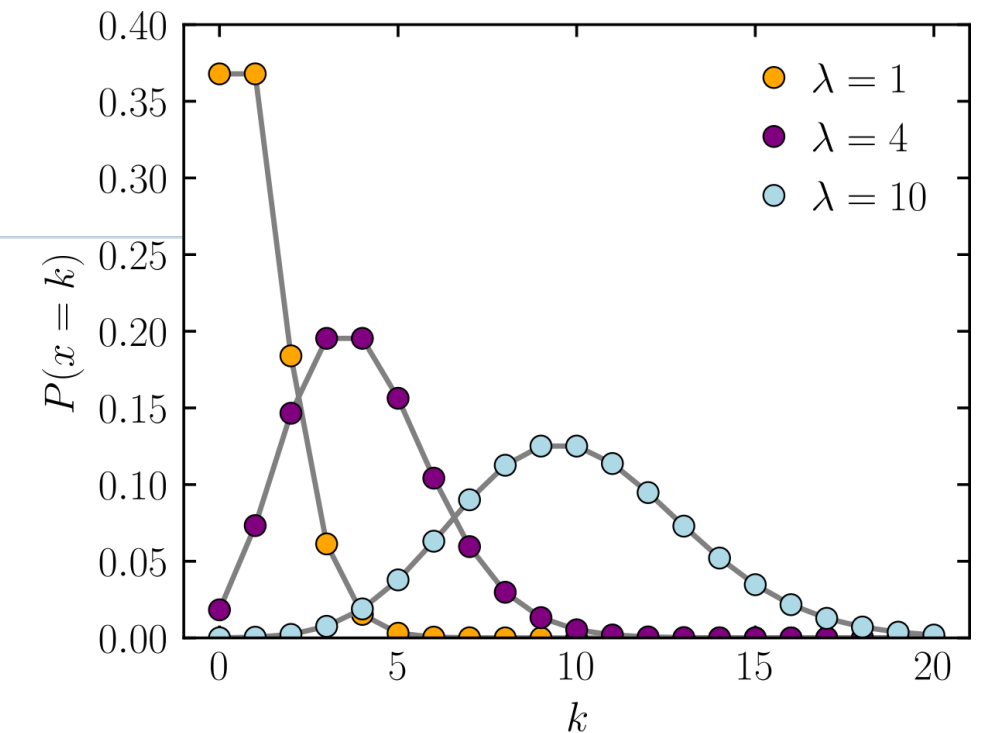
- 포아송 분포 $Z \sim \text{Poisson}(R)$

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

$$P(Z = z) = \frac{R_0^z e^{-R_0}}{z!}, z = 0, 1, 2, \dots$$

- 음이항분포 $Z \sim \text{NB}(R, k)$

$$P(Z = z) = \frac{\Gamma(k + z)}{z! \Gamma(k)} \left(\frac{R_0}{R_0 + k} \right)^z \left(1 + \frac{R_0}{k} \right)^{-k}, z = 0, 1, 2, \dots$$



실제 데이터 적용

- 2020년 발생한 wild type COVID-19 데이터 (n= 277)
- 데이터를 포아송분포와 음이항 분포를 적합시킨 후 각 모형의 감염재생산지수와 과대산포모수(R_0, k)를 추정

id	sex	date of onset	date of confirmation	date of quarantine	Category	cluster name	전파관계	source of infection	infectee
1	M	2020-05-02	2020-05-06	모름	이태원유흥시설	이태원	#1	#1	
2	M	무증상	2020-05-07	모름	이태원유흥시설	이태원	#2	#2	
3	M	무증상	2020-05-08	모름	지역사회전파	이태원관련	#1→#3	#1	#3
4	M	2020-05-06	2020-05-08	모름	이태원유흥시설	이태원	#4	#4	
5	M	2020-05-02	2020-05-08	모름	이태원유흥시설	이태원	#5	#5	
6	M	2020-05-07	2020-05-08	모름	이태원유흥시설	이태원	#6	#6	
7	M	2020-05-07	2020-05-08	모름	이태원유흥시설	이태원	#7	#7	
8	M	2020-05-05	2020-05-08	모름	이태원유흥시설	이태원	#8	#8	
9	M	2020-05-05	2020-05-08	모름	이태원유흥시설	이태원	#9	#9	
10	M	2020-05-06	2020-05-08	모름	이태원유흥시설	이태원	#10	#10	
11	M	#N/A	2020-05-08	모름	이태원유흥시설	이태원	#11	#11	
12	M	#N/A	2020-05-08	모름	이태원유흥시설	이태원	#12	#12	
13	M	#N/A	2020-05-08	모름	이태원유흥시설	이태원	#13	#13	

데이터 불러오기 및 전처리

id	sex	date of onset	date of confirmation	date of quarantine	Category	cluster name	전파관계	source of infection	infectee
1	M	2020-05-02	2020-05-06	모름	이태원유희시설	이태원	#1	#1	
2	M	무증상	2020-05-07	모름	이태원유희시설	이태원	#2	#2	
3	M	무증상	2020-05-08	모름	지역사회전파	이태원관련	#1→#3	#1	#3
4	M	2020-05-06	2020-05-08	모름	이태원유희시설	이태원	#4	#4	
5	M	2020-05-02	2020-05-08	모름	이태원유희시설	이태원	#5	#5	



```
# A tibble: 277 × 2
  infector.case infectee.case
  <chr>         <chr>
1 1            NA
2 2            NA
3 1            3
4 4            NA
5 5            NA
6 6            NA
7 7            NA
8 8            NA
9 9            NA
10 10          NA
```

```
#전체 데이터 불러오기
linelist <- read_excel("2_SSE_data_wildtype_KimH_JIPH2025.xlsx")

#source of infection, infectee 변수 선택 및 변수명 변경
overall_data <- linelist %>% dplyr::select("source of infection", "infectee") %>%
  rename(infector.case = "source of infection", infectee.case = "infectee")

#변수를 숫자 타입으로 바꾸기
overall_data <- overall_data %>%
  mutate(
    infector.case = gsub("#", "", infector.case),
    infectee.case = gsub("#", "", infectee.case)
  )

print(overall_data)
```

데이터를 count data로 변환

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

```
#감염된 사람들의 정보를 추출
infectee <- overall_data %>%
  dplyr::select(infector.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()
```

```
#최종적으로 감염된 사람들의 수 계산
nterminal_infectees <- infectee %>%
  dplyr::select(value) %>%
  filter(!value %in% duplicate$value) %>%
  transmute(case.no = as.numeric(value)) %>%
  nrow()
```

```
complete_offspringd <- enframe(c(offspring$n, rep(0, nterminal_infectees)))
```

```
> print(overall_data)
# A tibble: 277 × 2
  infector.case infectee.case
  <chr>         <chr>
1 1             NA
2 2             NA
3 1             3
4 4             NA
5 5             NA
6 6             NA
7 7             NA
8 8             NA
9 9             NA
10 10            NA
# i 267 more rows
```



```
> complete_offspringd
# A tibble: 372 × 2
  name value
  <int> <dbl>
1 1      15
2 2      12
3 3      12
4 4       9
5 5       9
6 6       8
7 7       6
8 8       6
9 9       5
10 10      5
# i 362 more rows
```


포아송, 음이항 분포 적합

```
# offspring count 데이터 가져오기
offspring_counts <- complete_offspringd %>% pull(value)

#POI
pfit <- offspring_counts %>% fitdist(., distr = 'pois')

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

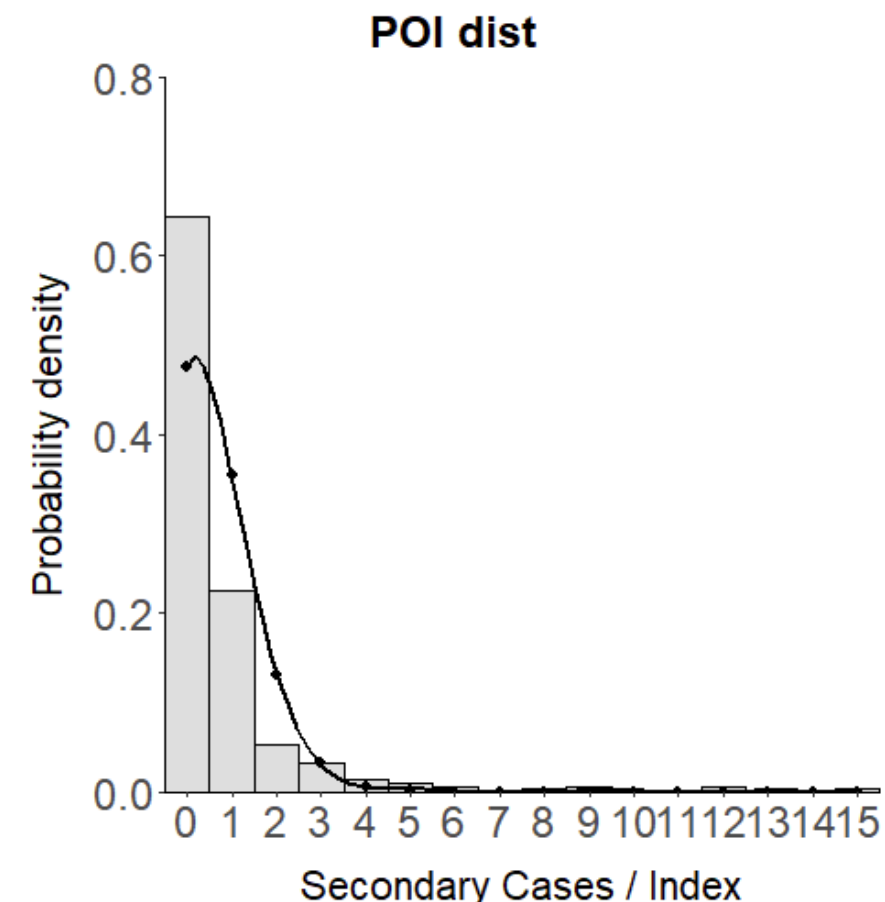
#결과 확인
summary(pfit)
summary(nbfit)
```

```
> summary(pfit)
Fitting of the distribution ' pois ' by maximum likelihood
Parameters :
      estimate Std. Error
lambda 0.7446237 0.04474002
Loglikelihood: -546.3255   AIC: 1094.651   BIC: 1098.57
> summary(nbfit)
Fitting of the distribution ' nbinom ' by maximum likelihood
Parameters :
      estimate Std. Error
size 0.4045195 0.06330407
mu    0.7447938 0.07543558
Loglikelihood: -428.4823   AIC: 860.9647   BIC: 868.8025
Correlation matrix:
      size      mu
size 1.0000000000 -0.0002280915
mu   -0.0002280915 1.0000000000
```

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

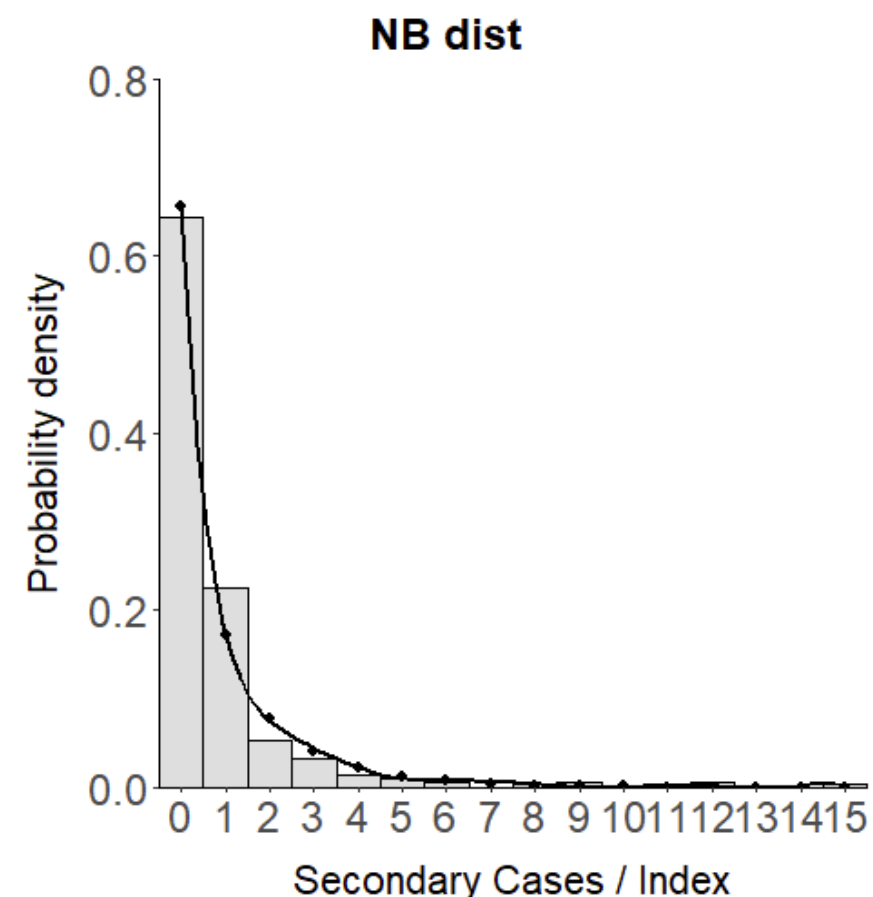
적합 플롯 만들기

```
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.