Chicago 311 Service Requests of Rodent Baiting

14조 분석 보고서
김소희 마시현 임윤지 한진희 홍지은

*각 조원들의 역할

임윤지:『Ⅱ-1』, 「Ⅰ-1」, 「Ⅰ-2」, 『Ⅲ 결론』
마시현:『Ⅱ-2』, 「Ⅰ-2」
김소희:『Ⅱ-3』, 「Ⅰ-2」
한진희:『Ⅱ-4』, 「Ⅰ-2」
홍지은:『Ⅱ-5』, 「Ⅰ-2」

Ⅰ. 서론

「Ⅰ-1」 주제 설명

시카고 311 서비스 센터에 접수된 쥐덫 요청과 쥐 불만에 대한 자료
모든 요청은 2011 년 1 월 1 일 이후 완료됨.
골목상태를 검사하고, 손상된 카트가 발견되면, 카트를 배포하는 Sanitation Ward Offices에 통보하고, 쥐약은 둥지를 박멸하기 위해 쥐 굴에 놓임.
311에서는 때때로 중복된 쥐 불만과 쥐덫에 대한 요청을 받음. 중복으로 분류된 요청은 동일한 지리적 영역에 있으며 이전 요청과 거의 동시에 311의 고객서비스요청(CSR) 시스템에 입력되었음.

「Ⅰ-2」변수 설명

변수 총 20개, 자료 개수 316,154개 (결측값 25,343개 제외하면 290811개)

	변수	변수 형태	변수 설명
1	Creation Date	범주형	불만이 제기 된 날짜
2	Status	범주형	요청 상태 (총 4가지) *(Completed/ Completed-Dup/ Open/ Open-Dup)*
3	Completion Date	범주형	요청이 완료된 날짜 (완료되지 않은 경우는 비워둠)
4	Service Request Number	범주형	각 서비스 요청에 대한 고유 식별자
5	Type of Service Number	범주형	서비스 요청 유형 (총 1가지) *(Rodent Baiting, Rat Complaint)*
6	Number of Premises Baited	연속형	쥐덫이 설치된 구내(건물 부지)의 수
7	Number of Premises with Garbage	연속형	쓰레기가 있는 구내(건물 부지)의 수
8	Number of Premises with Rats	연속형	쥐가 있는 구내(건물 부지)의 수
9	Current Activity	범주형	가장 최근에 한 활동내역(총 4가지) *(Dispatch Crew/ FVI - Outcome/* *Inspect for Violation/ Request Sanitation Inspector)*
10	Most Recent Action	범주형	지역 검사를 통해 제일 최근에 한 일 (총 11가지) *(Area Baited/ Area inspected, no cause and no baiting/* *Backyard serviced, contact made/ Completed/* *Create Work Order/ Inspected and baited/* *No contact, left door hanger/* *No contact, gate locked; left door hanger./* *Refer to Sanitation for Inspection)*
11	Street Address	범주형	도로까지 나온 주소
12	Zip Code	범주형	미국의 우편 번호
13	X Coordinate	연속형	주소의 x좌표
14	Y Coordinate	연속형	주소의 y좌표
15	Ward	범주형	주소에 따른 시의원의 구(1-50)
16	Police District	범주형	주소에 따른 경찰 관할 구역(25개의 구역)
17	Community Area	범주형	커뮤니티 지역을 0~77까지의 수로 분류한 변수
18	Latitude	Geography Data Type	지역의 위도를 나타내는 변수
19	Longitude	Geography Data Type	지역의 경도를 나타내는 변수
20	Location	Geography Data Type	지역의 위도 값과 경도 값을 함께 나타낸 변수

Ⅱ 분석 및 시각화

『Ⅱ-1』 각 경찰관할구역 별 평균 요청해결기간은?

분석방향:
요청이 완료되는데 평균적으로 며칠이 걸리는지를 알아보고, 각 경찰관할구역 별 평균 요청해결기간의 분포를 파악함.
→ 요청이 완료된 날짜(completion date), 불만이 제기 된 날짜(creation date), 경찰관할구역 변수를 이용함. Status를 보면 Completed/ Completed-Dup/ Open/ Open-Dup 네 가지가 있음을 알 수 있는데, 이 중 completed만 뽑아내어 분석함. 요청완료날짜에서 불만이 제기된 날짜를 빼서 해결기간을 파악.
→ geom histogram/ geom freqpoly를 이용하여 분포를 시각화함
분석과정
↓먼저, Status가 "completed" 인 자료들만 분석할 것이므로 필터링. 관심변수들만 따로 뽑아냄.

```
> new<-rodent%>%filter(Status=="Completed")%>%
+ select(`Completion Date`,`Creation Date`,`Police District`)
> new
# A tibble: 295,305 x 3
   `Completion Date`   `Creation Date`     `Police District`
   <dttm>              <dttm>                          <int>
 1 2011-01-05 00:00:00 2011-01-01 00:00:00                 9
 2 2011-01-05 00:00:00 2011-01-01 00:00:00                15
 3 2011-01-05 00:00:00 2011-01-01 00:00:00                24
 4 2011-01-05 00:00:00 2011-01-01 00:00:00                 7
 5 2011-01-05 00:00:00 2011-01-01 00:00:00                 4
 6 2011-01-05 00:00:00 2011-01-01 00:00:00                 8
 7 2011-01-05 00:00:00 2011-01-01 00:00:00                13
 8 2011-01-05 00:00:00 2011-01-01 00:00:00                 7
 9 2011-01-05 00:00:00 2011-01-01 00:00:00                25
10 2011-01-05 00:00:00 2011-01-01 00:00:00                13
# ... with 295,295 more rows
```

↓날짜형식으로 나와있으므로 separate함수를 써서 분리해줌.

```
> new_sep<-new%>%
+ separate(`Completion Date`,into=c("year_comp","month_comp","day_comp"),sep="-",convert = TRUE)%>%
+ separate(`Creation Date`,into=c("year_create","month_create","day_create"),sep="-",convert=TRUE)
> new_sep
# A tibble: 295,305 x 7
   year_comp month_comp day_comp year_create month_create day_create `Police District`
       <int>      <int>    <int>       <int>        <int>      <int>             <int>
 1      2011          1        5        2011            1          1                 9
 2      2011          1        5        2011            1          1                15
 3      2011          1        5        2011            1          1                24
 4      2011          1        5        2011            1          1                 7
 5      2011          1        5        2011            1          1                 4
 6      2011          1        5        2011            1          1                 8
 7      2011          1        5        2011            1          1                13
 8      2011          1        5        2011            1          1                 7
 9      2011          1        5        2011            1          1                25
10      2011          1        5        2011            1          1                13
# ... with 295,295 more rows
```

↓해결기간을 구하기 위해 month*30+day를 계산하여 일수로 형태를 바꾸어 줌.
또한 (요청완료일수-요청발생일수)를 계산하여 "term"이라는 변수에 저장함.

```
> new_dat<-new_sep%>%select(month_create,day_create,month_comp,day_comp,`Police District`)%>%
+ transmute(creationday=month_create*30+day_create,
+ completeday=month_comp*30+day_comp,police_district=`Police District`)
> new_dat
# A tibble: 295,305 x 3
   creationday completeday police_district
         <dbl>       <dbl>           <int>
 1          31          35               9
 2          31          35              15
 3          31          35              24
 4          31          35               7
 5          31          35               4
 6          31          35               8
 7          31          35              13
 8          31          35               7
 9          31          35              25
10          31          35              13
# ... with 295,295 more rows
> new_cal<-new_dat%>%mutate(term=completeday-creationday)
> new_cal
# A tibble: 295,305 x 4
   creationday completeday police_district  term
         <dbl>       <dbl>           <int> <dbl>
 1          31          35               9     4
 2          31          35              15     4
 3          31          35              24     4
 4          31          35               7     4
 5          31          35               4     4
 6          31          35               8     4
 7          31          35              13     4
 8          31          35               7     4
 9          31          35              25     4
10          31          35              13     4
# ... with 295,295 more rows
```

↓경찰관할 구역이 원래 1-25 밖에 없는데, 분포를 살펴보니 0과 31인 것도 나와있는 것을 확인.

```
> table(new_cal$police_district)

    0     1     2     3     4     5     6     7     8     9    10    11    12    13    14
   14  2903  4865  4390  5854  5031  8552 13319 30389 17196 10413 11754 12992  2070 21415
   15    16    17    18    19    20    21    22    23    24    25    31
 6424 18536 21562 10329 29475  8783   326  7472   510 19591 21118     2
```

↓따라서 구역의 결측치를 제거하고, 해결기간이 0보다 큰 관측치만 필터링 함.

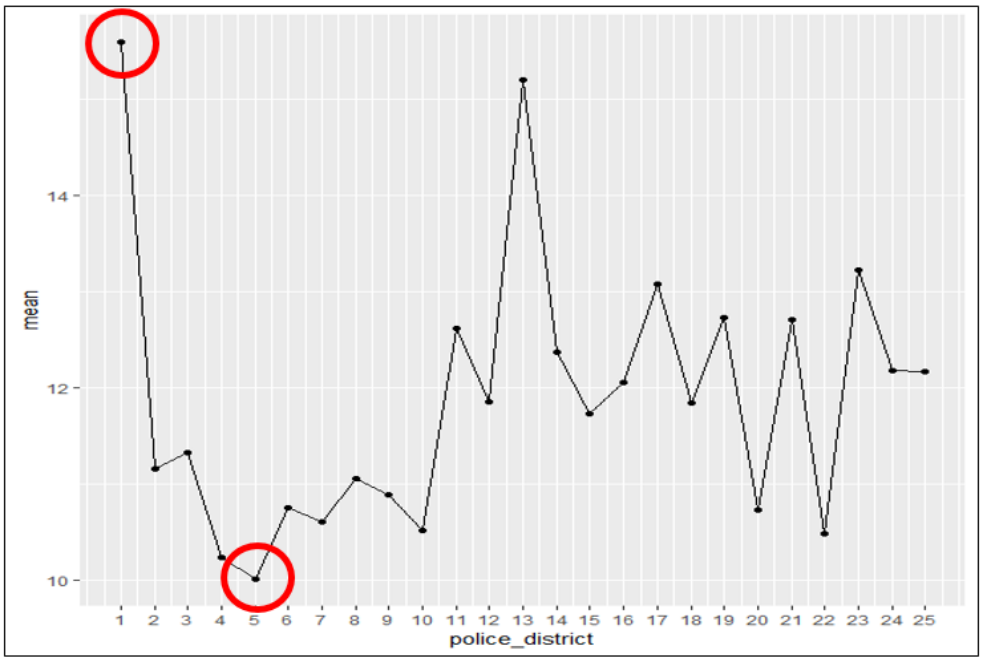
```
> new_filter<-new_cal%>%filter(!is.na(police_district),police_district%in%c(1:25),term>=0)
> new_filter
# A tibble: 293,471 x 4
   creationday completeday police_district  term
         <dbl>       <dbl>           <int> <dbl>
 1          31          35               9     4
 2          31          35              15     4
 3          31          35              24     4
 4          31          35               7     4
 5          31          35               4     4
 6          31          35               8     4
 7          31          35              13     4
 8          31          35               7     4
 9          31          35              25     4
10          31          35              13     4
# ... with 293,461 more rows
```

↓경찰 관할 구역별로 평균 요청해결기간을 구하고 평균의 오름차순으로 자료 정리함.

```
> new_filter%>%group_by(police_district)%>%summarise(mean=mean(term))%>%arrange(mean)
# A tibble: 25 x 2
   police_district  mean
             <int> <dbl>
 1               5  10.0
 2               4  10.2
 3              22  10.5
 4              10  10.5
 5               7  10.6
 6              20  10.7
 7               6  10.8
 8               9  10.9
 9               8  11.1
10               2  11.2
# ... with 15 more rows
```

↓ 이를 그래프로 시각화함.

```
> new_filter%>%group_by(police_district)%>%summarise(mean=mean(term))%>%
+ ggplot(aes(police_district,mean))+geom_freqpoly(stat="identity")+
+ geom_point()+scale_x_continuous(breaks=1:25)
```



결론: 평균 요청해결기간이 가장 오래 걸린 구역은 1구역으로 평균 15.6일, 가장 빨리 해결된 구역은 5구역으로 평균 10일이 걸림. 또한 그래프를 보면, 1구역을 제외한 낮은 숫자의 구역이 대체적으로 해결기간이 짧고, 높은 숫자의 구역이 대체적으로 해결기간이 긴 것을 볼 수 있음. 후의 분석에서 경찰관할구역 별 쓰레기, 쥐 수와 해결기간의 연관성을 볼 것임.

『Ⅱ-2』 각 관할 구역 별 특징 분석

관할 구역은 1에서 25의 수로 분류되어 있다. 하지만 원 데이터에는 아래와 같이 1에서 25의 수를 벗어난 0, 31과 필드 값이 없는 관측치가 존재한다. 관할 구역이 0, 31일 경우와 필드 값이 없는 경우의 관측치를 제거한 rodent_baiting_temp의 데이터셋을 생성한다. rodent_baiting_temp을 이용하여 각 관할 구역 별 특징을 분석한다.

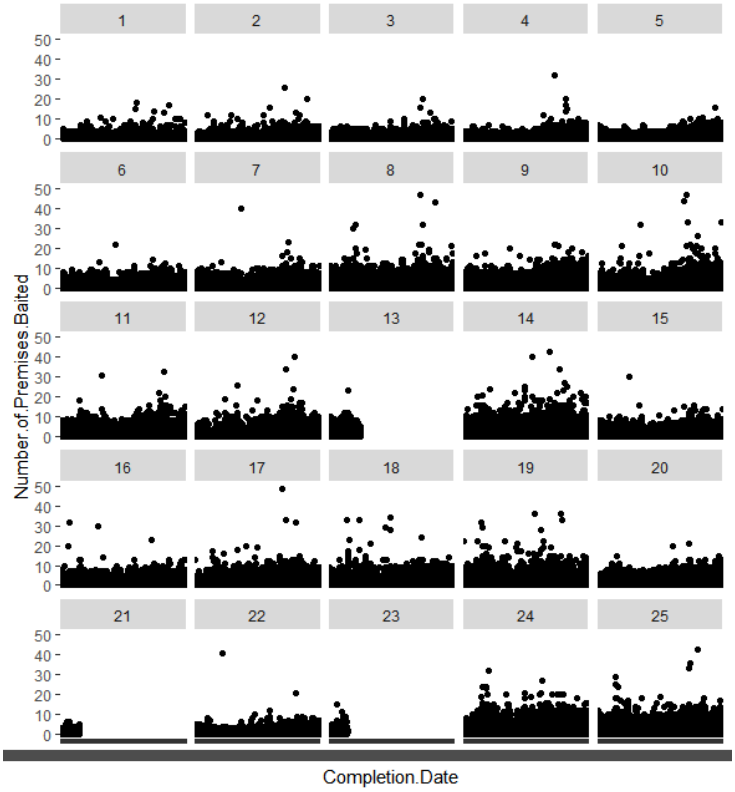
```
> which(!(rodent_baiting$Police.District %in% c(1:25)))
  [1]    2246    5288    6959   14322   16856   21986   36442   40955   41038   44496   50205   70640
 [13]   76776   88985   88988  106022  106023  134516  143600  151600  158796  158797  192130  194868
 [25]  196215  199334  234995  236760  237946  237947  237948  237949  260186  271165  271211  277333
 [37]  277418  307093  307094  307625  307628  315909
```

```
> rodent_baiting_temp<-rodent_baiting[-which(!(rodent_baiting$Police.District %in% c(1:25))),]
> which(!(rodent_baiting_temp$Police.District %in% c(1:25)))
integer(0)
```

위의 코드를 캡쳐한 것 중 which(!(rodent_baiting_temp$Police.District %in% c(1:25)))의 결과가 integer(0)이므로, rodent_baiting_temp 데이터셋은 관할 구역이 1~25인 관측치만 존재한다는 것을 알 수 있다.

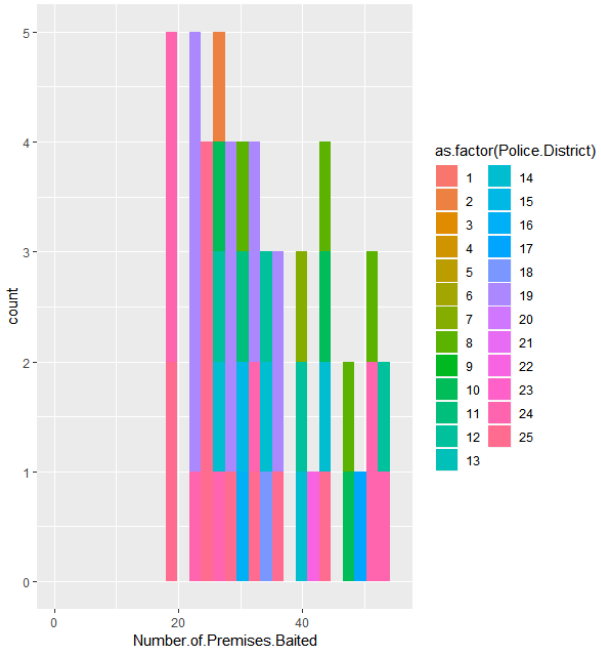
각 관할 구역 별 쥐덫이 설치된 건물 부지의 수를 분석하기 위해 25개의 관할 구역의 개별 창을 생성하여 점을 찍어 보았다. 그런데 오히려 25개의 개별 창을 생성하니 더욱 비교하기가 힘들었다.

```
> ggplot(rodent_baiting_temp,aes(Completion.Date,Number.of.Premises.Baited))+
+   geom_point()+facet_wrap(~Police.District)+ylim(0,50)
```



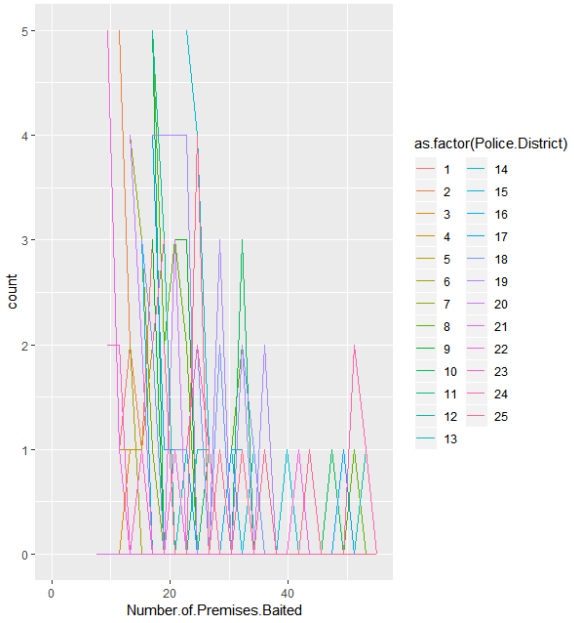
또한 점 플랏의 경우 점이 뭉겨져서 몇 개의 이상치만 파악할 수 있을 뿐, 대다수의 점들은 알아보기 힘들었다. 따라서 freqpoly와 histogram을 이용하여 각 관할 구역 별 건물 부지의 수에 관한 빈도를 한 창에 그려보았다.

```
> ggplot(rodent_baiting_temp,aes(Number.of.Premises.Baited))+
+   geom_histogram(aes(fill=as.factor(Police.District)))+ylim(0,5)+xlim(0,55)
```



geom_histogram을 사용하니 각 관할 구역 별 색깔을 다르게 설정해도 그래프가 겹쳐져서 알아보기 힘들었다.

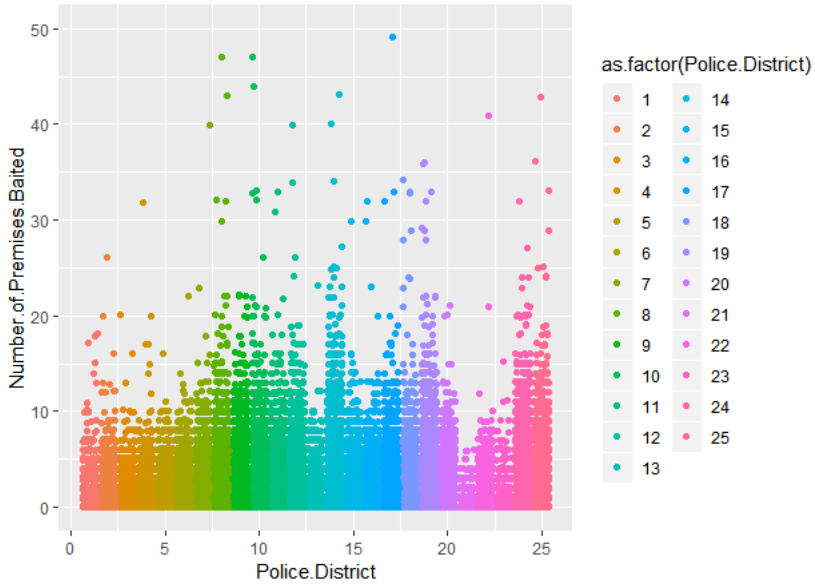
```
> ggplot(rodent_baiting_temp,aes(Number.of.Premises.Baited))+
+    geom_freqpoly(aes(color=as.factor(Police.District)))+ylim(0,5)+xlim(0,55)
```



geom_freqpoly를 사용한 경우에도 관할 구역이 25개나 되어서 25개의 선이 한 플랏에 그려져서 전체적으로 난잡한 그림이 나왔다. 관할 구역 별로 선의 색깔을 달리해도 선이 겹쳐져서 색깔을 판별하기 어려워 어떤 관할 구역인지 알아보기 힘든 경우도 발생했다.

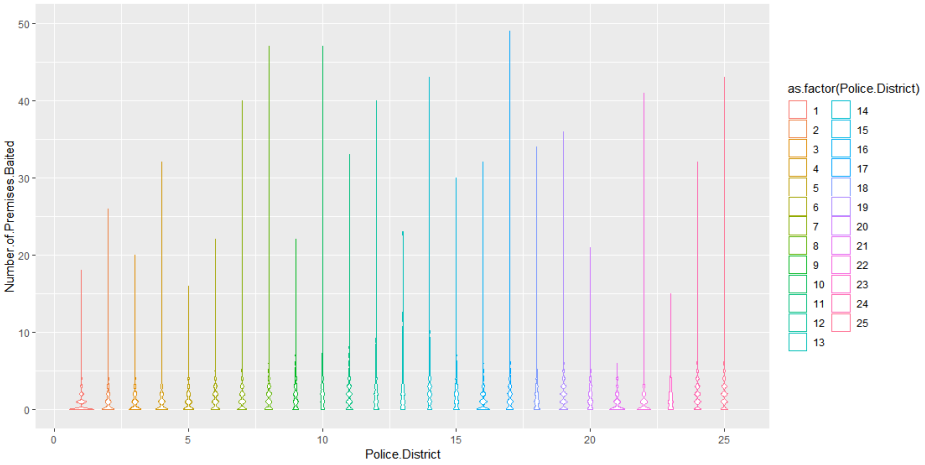
따라서 x축에 각 관할 구역이 **개별적으로 위치**하며 y축에는 건물 부지의 수의 빈도에 따라 점이 찍히는 플랏이 해당 데이터를 나타내기에 더 적합한 플랏이다. x축에 각 관할 구역이 위치라며 y축에는 건물 부지의 수가 빈도에 따라 점이 찍히는 플랏에는 geom_jitter · geom_violin · geom_boxplot이 있다. geom_jitter · geom_violin · geom_boxplot을 그려보고 데이터를 가장 잘 나타내는 플랏을 채택한다.

```
> ggplot(rodent_baiting_temp,aes(Police.District,Number.of.Premises.Baited))+
+   geom_jitter(aes(color=as.factor(Police.District)))+ylim(0,50)
```



geom_jitter의 경우, 빈도수가 너무 많은 곳은 점이 뭉쳐져서 빈도수가 많다는 정도만 짐작할 뿐, 중위수와 같이 의미 있는 값이 어디에 존재하는지 파악하기 힘들다.

```
> ggplot(rodent_baiting_temp,aes(Police.District,Number.of.Premises.Baited))+
+   geom_violin(aes(color=as.factor(Police.District)))+ylim(0,50)
```

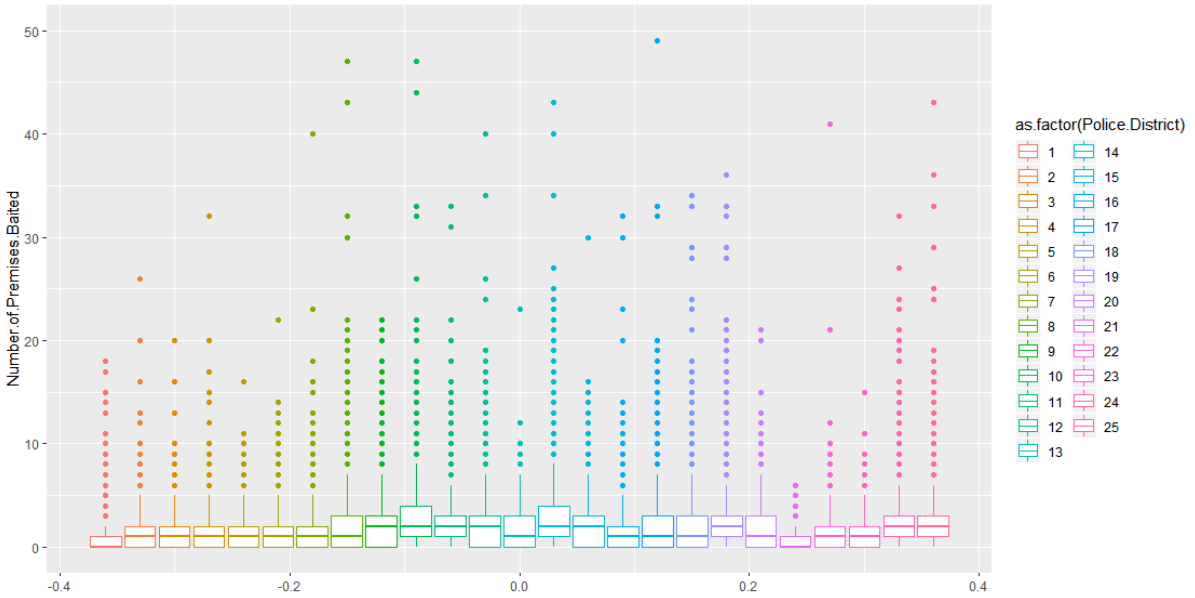


geom_violin으로 생성된 플랏의 경우를 보면, 두꺼운 부분을 보고 부지의 수로 어떤 값을 많이 가지는지 정도만 짐작할 뿐, 중위수와 같이 의미 있는 값을 파악할 수 없다. 따라서 중위수도 쉽게 파악할 수 있는 geom_boxplot을 사용하여 플랏을 그린다.

geom_boxplot을 이용하여 생성한 플랏이 여러 관할 구역을 비교하기 위해 그린 geom_point · geom_histogram · geom_freqpoly · geom_jitter · geom_violin · geom_boxplot 중 가장 25개의 구역의 차이점을 비교하기 쉬웠고, box plot을 이용함으로써 각 구역 별 쥐덫이 설치된 부지의 수의 중위수와 분포되어 있는 값의 범위를 한눈에 파악할 수 있어 25개의 구역을 비교하는 데에 탁월했다.

「Ⅱ-2」 - (1) 각 관할 구역 별 쥐덫이 설치된 건물 부지의 수

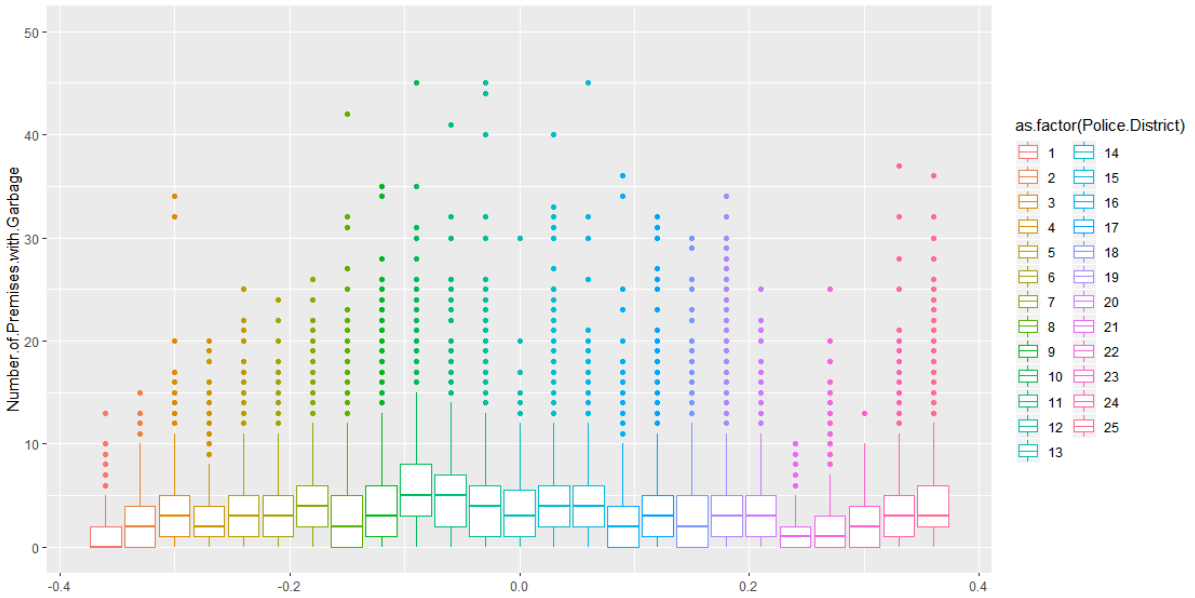
```
> ggplot(rodent_baiting_temp,aes(group=as.factor(Police.District),y=Number.of.Premises.Baited))+
+   geom_boxplot(aes(color=as.factor(Police.District)))+ylim(0,50)
```



- 관할 구역 8·10·11·12·14·16·17·18·22·24·25 쥐덫이 설치된 건물의 수가 상대적으로 넓은 범위의 값을 가진다.
- 관할 구역 9·10·11·12·14·15·19·24·25가 다른 구역보다 쥐덫이 설치된 건물의 수에 있어서 높은 중위 값을 가진다.
- 관할 구역 16에 포함된 지역 내 쥐덫이 설치된 건물 부지의 수는 6을 넘어가지 않았다.
- 관할 구역 22는 쥐덫이 설치된 건물 부지의 수가 50에 육박한 곳도 존재했다.
- 각 관할 구역 별 쥐덫이 설치된 건물 부지의 수가 비슷한 중위값을 가진다는 것에서 쥐덫이 설치되는 것에 예산이 정해져 있어서 쥐덫을 설치하는 건물 부지의 수가 특정 값으로 제한되어 있다는 것을 짐작해볼 수 있다.

「Ⅱ-2」 - (2) 각 관할 구역 별 쓰레기가 있는 건물 부지의 수

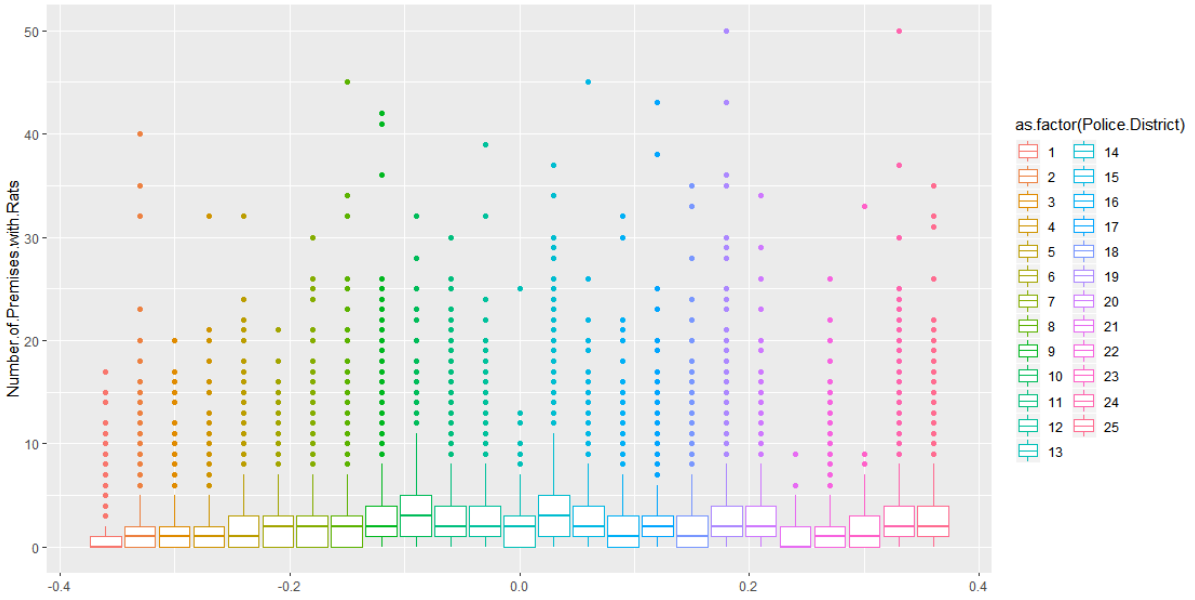
```
> ggplot(rodent_baiting_temp,aes(group=(Police.District),y=Number.of.Premises.with.Garbage))+
+   geom_boxplot(aes(color=as.factor(Police.District)))+ylim(0,50)
```



- 관할 구역 별 쓰레기가 있는 건물 부지의 수를 나타낸 boxplot은 관할 구역 별 쥐덫이 설치된 건물 부지의 수를 나타낸 boxplot과 형태가 매우 유사하다. 쓰레기가 있는 건물 부지의 수가 큰 값을 갖는 관할 구역에서 대체로 쥐덫이 설치된 건물 부지의 수도 큰 값을 보였다. 또한 쓰레기가 있는 건물 부지의 수가 작은 값을 갖는 관할 구역에서 쥐덫이 설치된 건물 부지의 수도 작았다. 즉, 쥐덫과 쓰레기의 상관관계가 있음을 짐작해볼 수 있었다.
- 앞의 관할 구역 별 쥐덫이 설치된 건물 부지의 수와는 다르게 쓰레기가 있는 건물 부지의 수를 나타낸 그래프는 관할 구역 별 다양한 값의 중위수를 가진다는 것을 알 수 있다.
- 관할 지역 중 10·12·15은 쓰레기가 있는 건물 부지의 수가 45의 값을 가졌다.
- 관할 지역 1은 쓰레기가 있는 건물 부지의 수가 13이하의 상대적으로 작은 값을 가졌다.

「Ⅱ-2」 - (3) 각 관할 구역 별 쥐가 있는 건물 부지의 수

```
> ggplot(rodent_baiting_temp,aes(group=(Police.District),y=Number.of.Premises.with.Rats))+
+    geom_boxplot(aes(color=as.factor(Police.District)))+ylim(0,50)
```



- 각 관할 구역 별 쥐가 있는 건물 부지의 수도 앞서 보인 쓰레기가 있는 건물 부지의 수와 쥐덫이 설치된 건물 부지의 수와 비슷한 양상을 보였다. 각 관할 구역의 중위수를 보면, 관할 구역 10은 타 구역에 비해 높은 중위수를 가지는데, 관할 지역 10은 쥐덫 · 쓰레기가 있는 건물 부지의 수가 타 구역에 비해 높은 값을 가진다. 쥐덫, 쓰레기의 존재와 쥐의 존재 사이에 상관관계가 있음을 짐작할 수 있다.
- 관할 구역 19·24는 쥐가 있는 건물 부지의 수가 50인 매우 높은 값이 존재한다.
- 관할 구역 1·21은 쥐가 있는 건물 부지의 수로 상대적으로 작은 값들을 가진다.

(단, 『Ⅱ-2』 의 결과를 보면 각 관할구역의 쥐덫이 존재하는 건물부지의 수, 각 관할구역의 쓰레기가 존재하는 건물부지의 수, 각 관할구역의 쥐가 존재하는 건물부지의 수를 분석했을 때, 쥐덫, 쓰레기 존재여부, 쥐의 존재여부에 상관관계가 있는 듯 하였지만, 『Ⅱ-3』 에서 관할구역으로 conditioned된 것을 제거하고 세 변수 사이의 상관관계에 주목해 보았을 때 세 변수 사이에는 뚜렷한 관계가 존재하지 않았다.)

『Ⅱ-3』

「Ⅱ-3」 -(1) 쓰레기가 있는 구내(건물부지)의 수가 많으면 쥐가 있는 구내의 수가 많은가? (쓰레기와 주의 관련성 탐색)

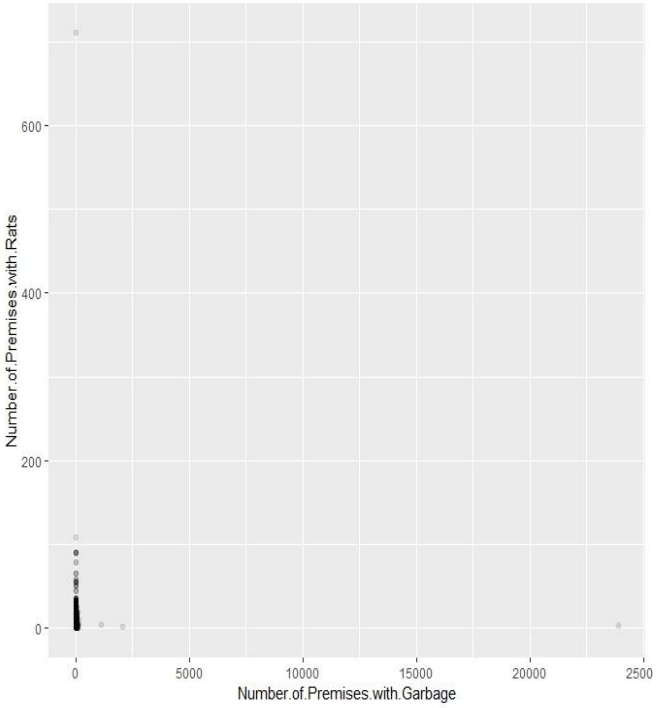
쓰레기가 많으면 쥐가 많아지는지 관련성을 탐색하고자 하였다. 하지만 쓰레기 수와 쥐의 수를 직접적으로 나타내는 변수가 없었기 때문에 쓰레기가 있는 구내의 수와 쥐가 있는 구내의 수를 선택해서 관련성을 알아보고자 했다.

```
> data1<-select(data, 'Creation.Date',Status,'Completion.Date','Number.of.Premises
.Baited','Number.of.Premises.with.Garbage','Number.of.Premises.with.Rats')
```

데이터에서 select를 이용하여 'Creation.Date', Status, 'Completion.Date', 'Number.of.Premises.Baited', 'Number.of.Premises.with.Garbage', 'Number.of.Premises.with.Rats'만 선택하여data1에 저장하였다..

```
> ggplot(data1, aes(`Number.of.Premises.with.Garbage`,`Number.of.Premises.with.Rats`))+geom_point(alpha=1/10
Warning message:
Removed 23979 rows containing missing values (geom_point).
```

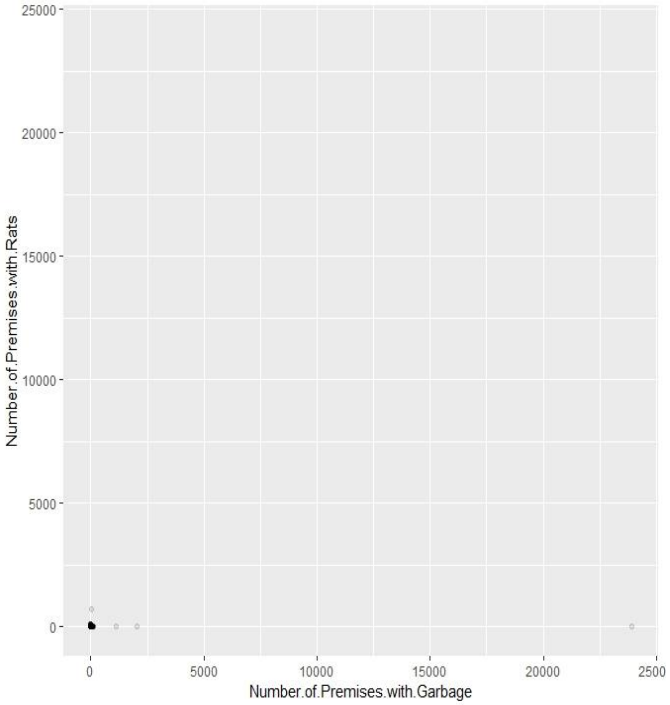
쓰레기가 있는 구내의 수 (Number of premises with garbage)를 x 변수로, 쥐가 있는구내의 수 (number of premises with rats)를 y 변수로로 하는 geom_point 그래프를 그려보면, missing value를 제거하고 그렸다는 warning message가 뜬다.



이대로 진행하여 알아볼 수 있지만, Warning message가 뜨지 않고 자체적으로 missing data를 제거 하고 싶어서 na.omit()을 이용하여 다시 그래프를 그려보았다.

```
> ggplot(data=na.omit(data1), aes(`Number.of.Premises.with.Garbage`,`Nu
mber.of.Premises.with.Rats`))+geom_point(alpha=1/10)
```

아까와는 다르게 warning message가 뜨지 않는 것을 알 수 있었다.



그래프를 보면 데이터가 한쪽에 몰려 있어서 알아보기 어려웠고, 대게 쓰레기가 있는 구내 수는 2500이하, 쥐는 200이하임을 알 수 있다.
확인해 보기 위해 filter를 사용하여 쓰레기가 있는 부지의 수가 2500인 데이터의 개수를 확인해보니 한 개만 나왔음을 알 수 있었다.

```
> data1%>% filter(`Number.of.Premises.with.Garbage`>=2500)
        Creation.Date    Status     Completion.Date Number.of.Premises.Baited
1 2015-05-07T00:00:00 Completed 2015-05-12T00:00:00                         3
  Number.of.Premises.with.Garbage Number.of.Premises.with.Rats
1                           23908                            3
```

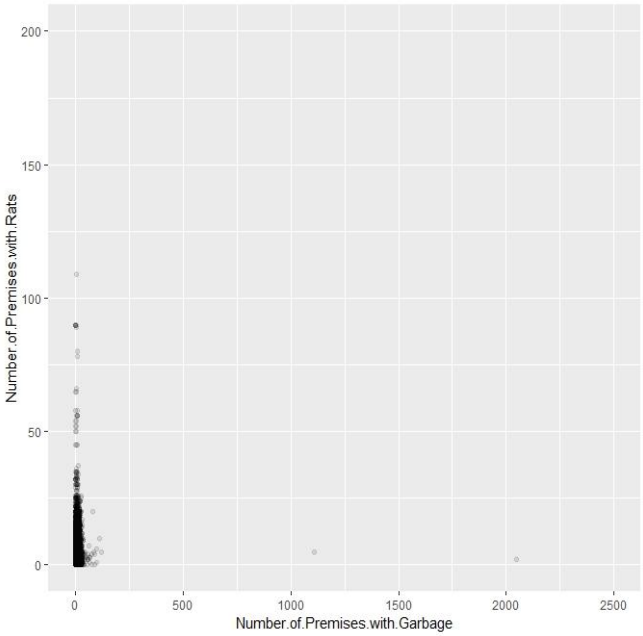
이는 한 개 데이터 빼고 나머지는 2500 이하임을 나타낸다. 쥐가 있는 부지의 수 중 200 이상인 것의 개수를 알아보기 위해 filter를 이용하였더니 여덟 개가 있었다.

```
> data1%>% filter(`Number.of.Premises.with.Rats`>=200)
       Creation.Date    Status     Completion.Date Number.of.Premises.Baited
1 2011-11-22T00:00:00 Completed 2011-11-25T00:00:00                         0
2 2013-08-20T00:00:00 Completed 2013-09-06T00:00:00                       807
3 2016-03-30T00:00:00 Completed 2016-04-04T00:00:00                        NA
4 2016-06-14T00:00:00 Completed 2016-06-22T00:00:00                         1
5 2017-05-30T00:00:00 Completed 2017-05-31T00:00:00                        NA
6 2017-07-13T00:00:00 Completed 2017-07-17T00:00:00                        NA
7 2017-07-28T00:00:00 Completed 2017-07-31T00:00:00                         4
8 2017-08-07T00:00:00 Completed 2017-08-15T00:00:00                        NA
  Number.of.Premises.with.Garbage Number.of.Premises.with.Rats
1                              NA                          912
2                              NA                          937
3                              NA                          932
4                              NA                          303
5                              NA                        23957
6                              NA                         4928
7                              10                          710
8                              NA                          708
```

이는 쥐가 있는 구내 수가 200이상이 8개 빼고 200이하라는 것을 나타낸다. .
(쥐덫과 쓰레기가 있는 구내의 수의 missing value 포함하여 8개)

이는 많은 자료 중 적게 차지하기 때문에 관련성을 알아보기에 빼도 큰 무리가 없을 것이라 판단하여 coord_cartesian()의 xlim, ylim 옵션을 이용하여 zooming하여 살펴보았다.

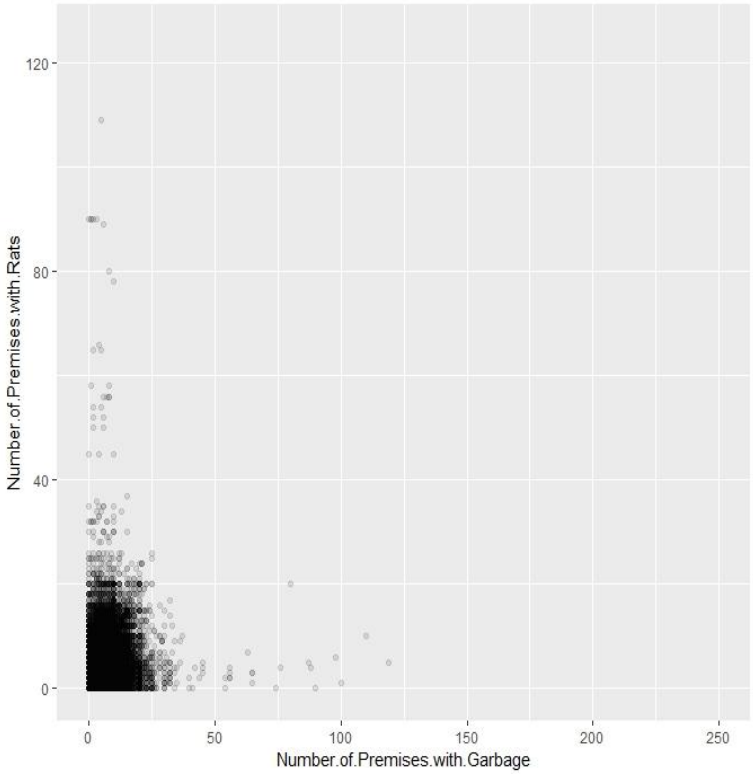
```
> ggplot(data=na.omit(data1), aes(`Number.of.Premises.with.Garbage`,`Number.of.Premises.with.Rats`))+geom_poi
nt(alpha=1/10)+coord_cartesian(xlim=c(0,2500),ylim=c(0,200))
```



여전히 자료가 몰려있는 것을 알 수 있다. 따라서 이들의 상관관계를 알아보기 위하여 자료가 모여있는 부분을 보기로 한다. Xlim, ylim을 더 줄여보았다.

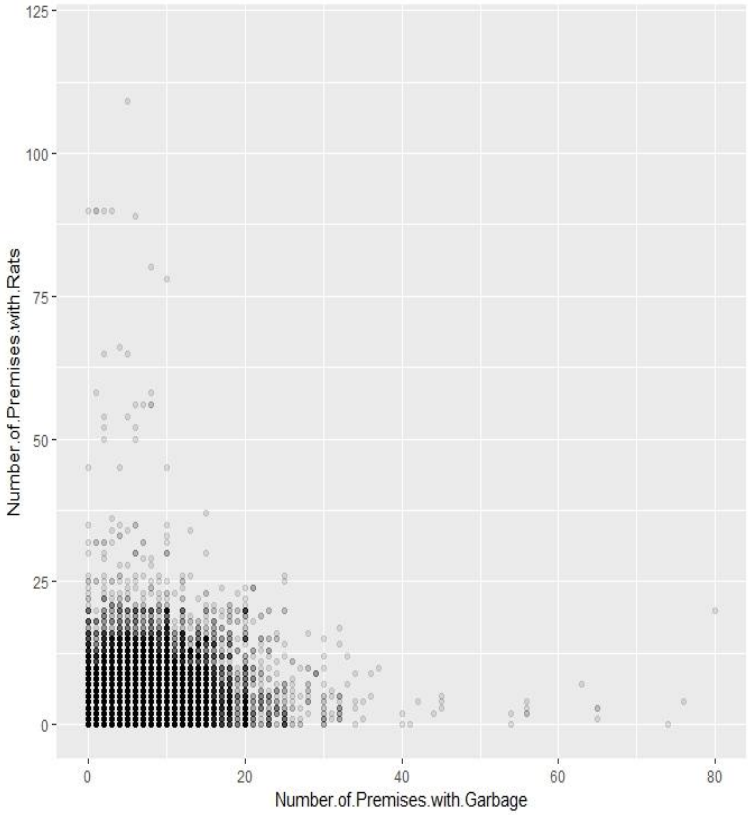
```
> ggplot(data=na.omit(data1), aes(`Number.of.Premises.with.Garbage`,`Number.of.Premises.with.Rats`))+geom_po
nt(alpha=1/10)+coord_cartesian(xlim=c(0,250),ylim=c(0,125))
```

쓰레기가 있는 구내의 수인 x축의 범위는 0에서 250까지, 쥐가 있는 구내의 수인 y축의 범위는 0에서 125까지 조정하여 geom_point 그래프를 그려보았다. Alpha=1/10으로 앞과 동일하게 하였다.



상관관계가 없는 것처럼 보이지만 데이터가 한쪽에 여전히 몰려있는 것을 확인 할 수 있다. 데이터가 몰려 있는 부분을 자세하게 보고 싶어 x축과 y축의 범위를 조정하여 그래프를 다시 그려보았다.

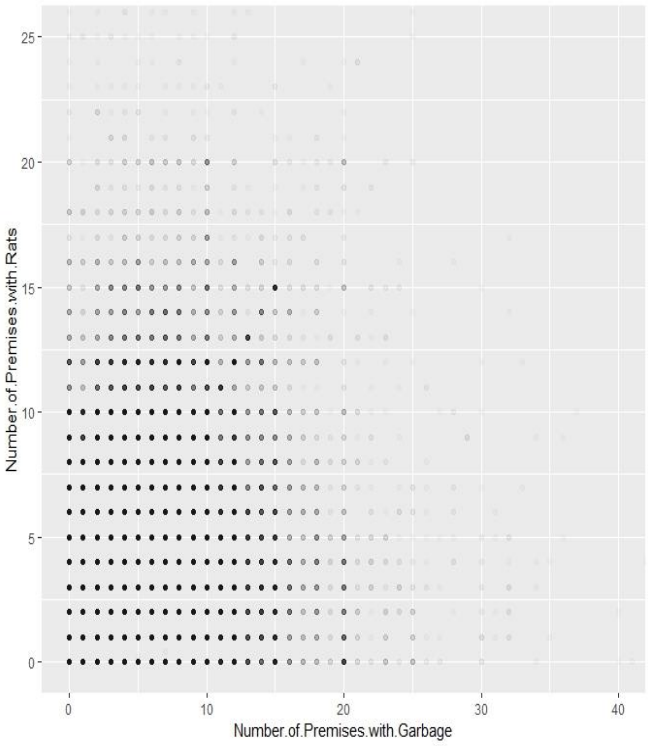
```
> ggplot(data=na.omit(data1), aes(`Number.of.Premises.with.Garbage`,`Number.of.Premises.with.Rats`))+geom_poi
nt(alpha=1/10)+coord_cartesian(xlim=c(0,80),ylim=c(0,120))
```



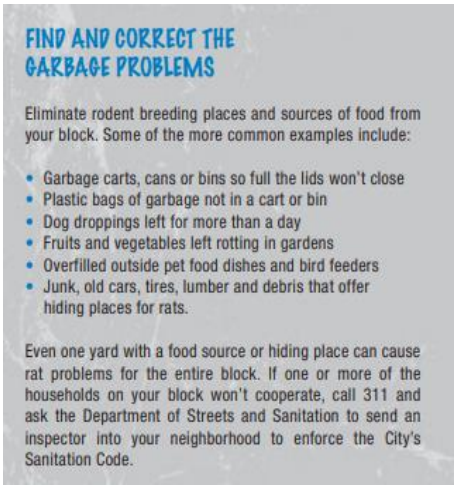
그림을 그려 보았더니 양의 상관관계나 음의 상관관계를 띄지 않는다고 할 수 있다. 불투명도 사용하여 어느 값에 몰려있는지 보고 싶었으나 한 곳에 몰려있기에 대부분 진하게 나와있어 한눈에 알아보기 어려웠다..

```
> ggplot(data=na.omit(data1), aes(`Number.of.Premises.with.Garbage`,`Number.of.Premises.with.Rats`))+geom_poi
nt(alpha=1/50)+coord_cartesian(xlim=c(0,40),ylim=c(0,25))
```

X축 y축 값을 작게 조절하여 그래프를 그리면 다음과 같은 그래프를 얻을 수 있다.



투명도를 더 조절해보고, x값과 y값의 범위를 더 작게 해보았지만 특별한 관계는 보이지 않는다. 따라서 쓰레기가 있는 구내수가 많아져도 쥐가 있는 구내 수가 많아지는 것에는 특별한 상관관계가 없다는 것을 알 수 있다. 쓰레기가 많으면 쥐가 많다고 생각했지만 가정이 틀렸다고 볼 수 있다.
이 그래프에서 쓰레기가 있는 구내의 수가 20개 이하에서 쥐가 많이 있다고 볼 수 있다.
왜 가설이 틀렸는지 생각해보기 위해 자료를 더 찾아보았다. 시카고에서 시민들에게 당부한 자료를 보면


FIND AND CORRECT THE GARBAGE PROBLEMS

Eliminate rodent breeding places and sources of food from your block. Some of the more common examples include:

- Garbage carts, cans or bins so full the lids won't close
- Plastic bags of garbage not in a cart or bin
- Dog droppings left for more than a day
- Fruits and vegetables left rotting in gardens
- Overfilled outside pet food dishes and bird feeders
- Junk, old cars, tires, lumber and debris that offer hiding places for rats.

Even one yard with a food source or hiding place can cause rat problems for the entire block. If one or more of the households on your block won't cooperate, call 311 and ask the Department of Streets and Sanitation to send an inspector into your neighborhood to enforce the City's Sanitation Code.

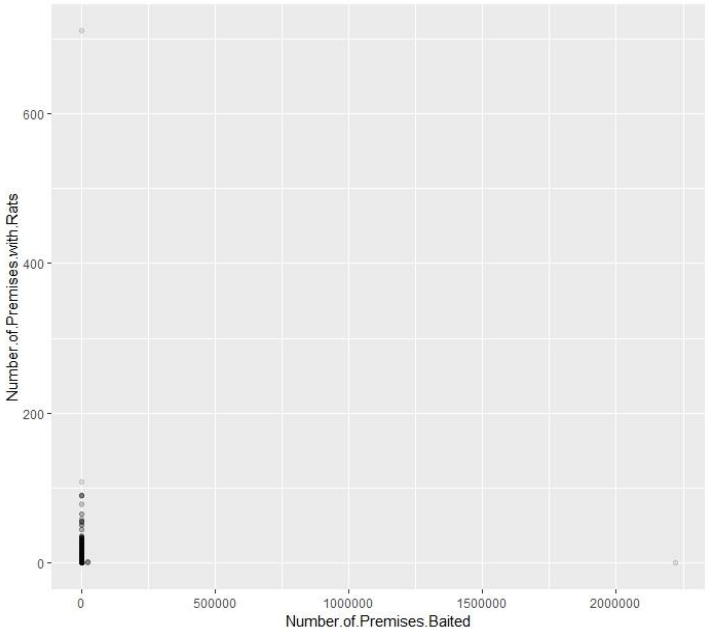
Figure 1 출처: http://ward32.org/wp-content/uploads/2010/03/Bureau-of-Rodent-Control-Brochure.pdf
식량이나 은신처가 있는 1야드 조차도 쥐 문제를 일으킬 수 있으므로 뚜껑이 닫히지 않을 정도로 가득 찬 쓰레기통, 강아지 배설물, 음식물 쓰레기 등은 제거하라고 나와있다.
블록에 있는 가구 중 하나 이상이 협조하지 못하면 311번에 전화를 걸어 도로 및 위생부가 도시의 위생법을 시행하기 위해 경찰을 보내달라고 요청하라는 당부가 나온다. 따라서 쓰레기가 많아지기 전에, 시민들은 자발적으로 쓰레기를 치웠을 가능성이 있기 때문에 이러한 결과가 나오지 않았을까 하고 생각해볼 수 있다.

「Ⅱ-3」 -(2) 쥐덫이 많이 설치된 구내(건물 부지)에는 쥐가 많은가 ? (쥐덫과 쥐의 관련성 탐색)
쥐덫이 많으면 쥐가 많이 있는지 알아보기 위해 쥐덫이 설치된 구내의 수와 쥐가 있는 구내의 수의 관련성을 따져보고자 했다. x축을 쥐덫이 있는 구내의 수 (number of premises baited) y축을 쥐가 있는 구내의 수 (number of premises with rats)로 하여 alpha=1/10인 geom_point 그래프를 그려보았다.

```
> ggplot(data1, aes(`Number.of.Premises.Baited`,`Number.of.Premises.with.Rats`))+geom_point(alpha=1/10)
Warning message:
Removed 23568 rows containing missing values (geom_point).
```

역시나 missing value가 존재하여 r이 제거해주었지만 자체적으로 제거 하고 싶어 na.omit()을 이용하였다.

```
> ggplot(data=na.omit(data1), aes(`Number.of.Premises.Baited`,`Number.of.Premises.with.Rats`))+geom_point(
alpha=1/10)
```



이 역시도 데이터가 한 곳에 몰려있는 것을 알 수 있었다. 이는 관련성을 알아보기에 어렵다는 판단을 내려 범위를 조절하고자 하였다.

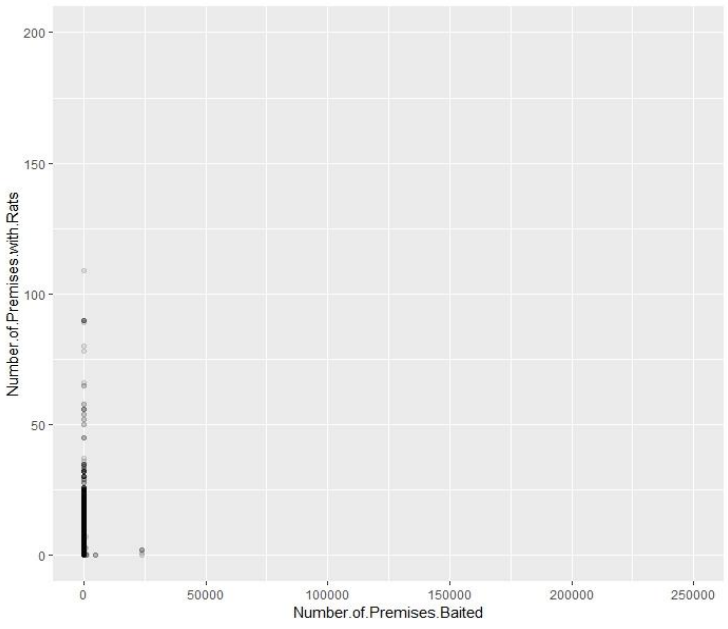
Number of premises baited는 25000이상이 없다고 생각하여 filter를 이용하여 쥐덫이 있는 부지의 수 중 250000 이상인 데이터의 개수를 구해보았더니 하나도 없다는 것을 알 수 있었다.

```
> data1%>% filter(`Number.of.Premises.Baited`>=250000)
        Creation.Date    Status     Completion.Date Number.of.Premises.Baited
1 2018-06-15T00:00:00 Completed 2018-06-19T00:00:00                   2222220
  Number.of.Premises.with.Garbage Number.of.Premises.with.Rats
1                               0                           0
```

number of premises with rats는 앞에서 분석해보았던 대로 200 이상은 8개임을 알 수 있다.

따라서 X(쥐덫이 있는 구내의 수)축의 범위를 0에서 250000, y(쥐가 있는 구내의 수)축의 범위를 0에서 200까지 조절하고 그려보았더니 다음과 같은 그래프가 나왔다.

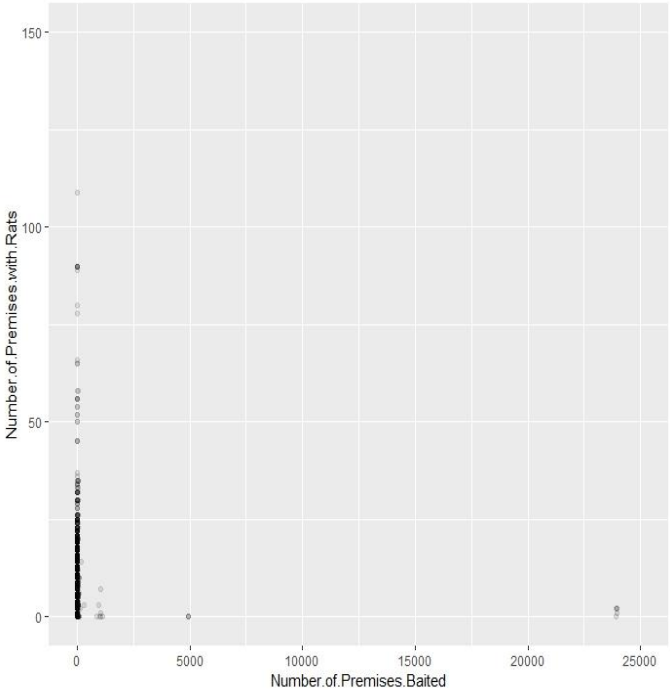
```
> ggplot(data=na.omit(data1), aes(`Number.of.Premises.Baited`,`Number.of.Premises.with.Rats`))+geom_point(
alpha=1/10)+coord_cartesian(xlim=c(0,250000),ylim=c(0,200))
```



아직도 데이터가 한쪽에 모여있기에 x축의 범위를 (0,25000), y축의 범위를 (0,150)으로 다시 조사

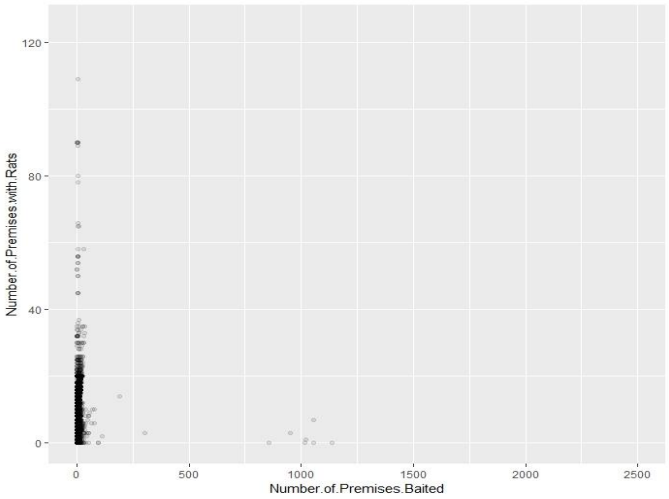
해보았다.

```
> ggplot(data=na.omit(data1), aes(`Number.of.Premises.Baited`,`Number.of.Premises.with.Rats`))+geom_point(
alpha=1/10)+coord_cartesian(xlim=c(0,25000),ylim=c(0,150))
```



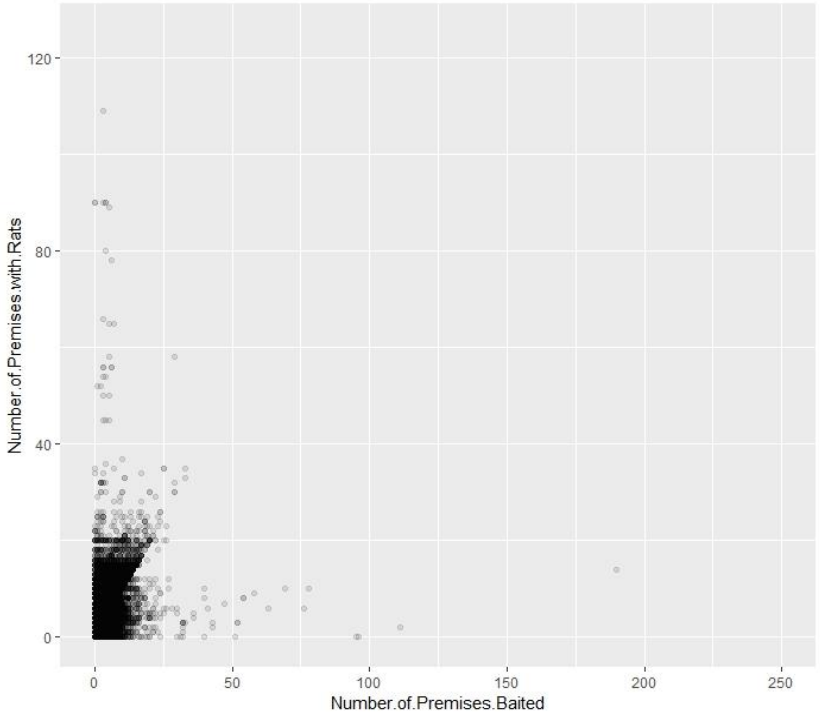
x축의 범위를 크게 조절하였음에도 여전히 0 부근에서 몰려있는 것을 볼 수 있었다. x변수 중 5000과 25000 근처에 자료가 있는 것을 보았지만 불투명도를 보았을 때 데이터가 많지 않다는 것을 알 수 있고 이는 제거해도 관련성 예측에 큰 영향을 미치지 않을 것이라 생각되었다. x을 2500까지, y축은 125 이상이 거의 없을 것이다 라고 생각이 되어 y축은 125까지 조절하여 그래프를 다시 그려보았다.

```
> ggplot(data=na.omit(data1), aes(`Number.of.Premises.Baited`,`Number.of.Premises.with.Rats`))+geom_point(
alpha=1/10)+coord_cartesian(xlim=c(0,2500),ylim=c(0,125))
```



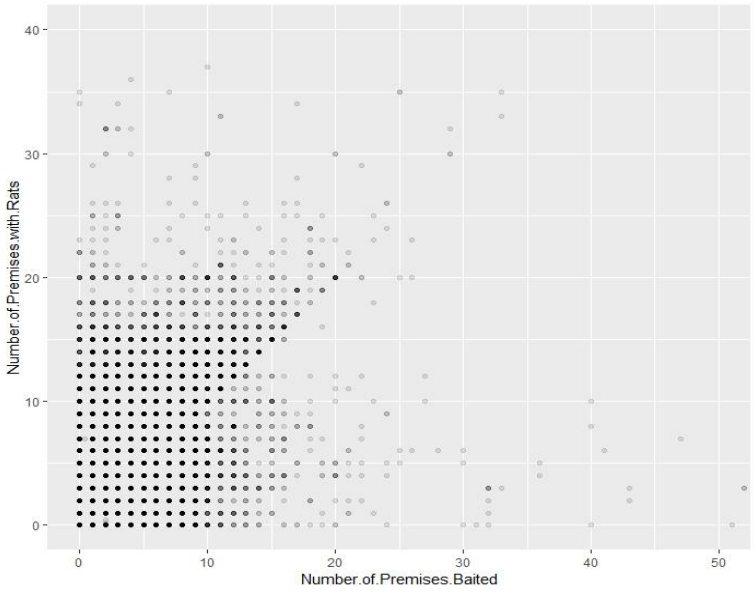
앞서 보았던 것보다 데이터가 더 퍼져있지만 여전히 몰려있는 것을 보아 x범위를 0~250까지, y 범위를 0~125까지 조절하여 다시 그림을 그려보았다.

```
> ggplot(data=na.omit(data1), aes(`Number.of.Premises.Baited`,`Number.of.Premises.with.Rats`))+geom_point(
alpha=1/10)+coord_cartesian(xlim=c(0,250),ylim=c(0,125))
```



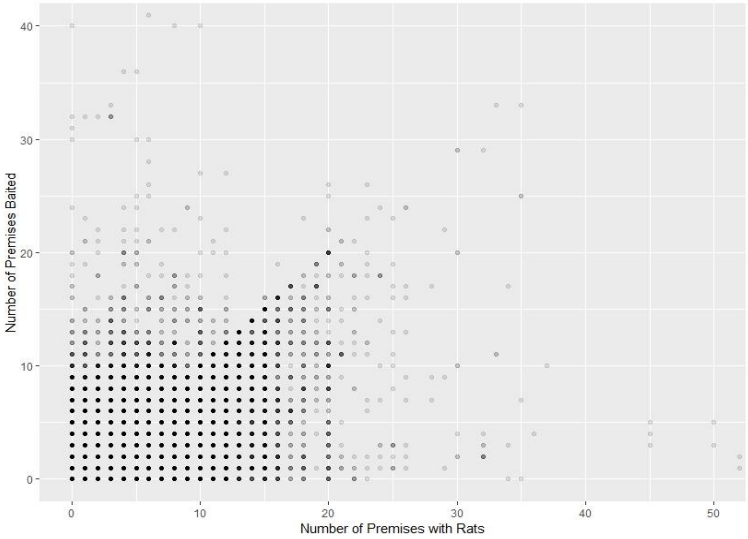
그림으로 보아 조금은 양의 상관관계가 있는 것 처럼 보이지만, 쥐덫이 있는 건물 부지 수와 쥐가 있는 건물 부지 수 사이에는 뚜렷한 상관관계가 없음을 알 수 있다.

```
> ggplot(data=na.omit(data1), aes(`Number.of.Premises.Baited`,`Number.of.Premises.with.Rats`))+geom_point(
alpha=1/10)+coord_cartesian(xlim=c(0,50),ylim=c(0,40))
```



x축과 y축을 서로 바꿔서 그래프를 그려 보았다.

```
> ggplot(data=na.omit(data1), aes(`Number of Premises with Rats`,`Number of Premises Baited
`))+geom_point(alpha=1/10)+coord_cartesian(xlim=c(0,50),ylim=c(0,40))
```



이 역시 양의 상관관계가 있는 것처럼 보이지만 뚜렷한 상관관계가 없음을 알 수 있다. 쥐가 많아질수록 쥐덫이 많다던가 그 반대의 경우의 가설도 오류가 있다고 말할 수 있다.
이로써 우리가 설정한 쥐덫이 있는 건물의 수가 많으면 쥐가 많은가? 라는 가설은 틀렸음을 알 수 있다. 이 이유를 유추해보니, 쥐덫이 설치된 것이 쥐가 있다고 신고 받기 전에 건물에 사는 사람이 설치한 것인지, 아니면 쥐가 있다고 신고 받은 후, 시카고 시에서 건물에 출동하여 쥐덫을 설치한 것인지 알 수 가 없기에 가설이 틀렸을 가능성이 높다.
다른 이유는 시카고에서 사용되는 쥐덫의 종류로 알아볼 수 있다. 시카고에서 사용되는 쥐덫은 쥐를 불임 상태로 만드는 독을 미끼 상자 안에 넣고 쥐를 유인하여 죽이는 것이다. (출처: https://chicago.suntimes.com/news/chicagos-new-rat-control-approach-is-poison-that-makes-them-infertile/) 쥐를 불임으로 만들어 번식할 수 없게 만듦으로써 쥐의 개체 수를 줄일 수 있다. 쥐덫에 대한 정보로 보아 만약 하나의 쥐덫이 있으면 쥐 한 마리가 아닌 여러 마리의 쥐를 없앨 수 있기 때문에 쥐덫이 많은 건물일수록 쥐가 많이 나온다는 가설에는 오류가 있음을 알 수 있다. 또한 쥐덫이 특정 개수 이하에 몰려 있는 것을 보아 쥐덫을 설치하는 데에 필요한 예산이 정해져 있는 것이 아닐까 하고 유추해 볼 수 도 있다.

『Ⅱ-4』

「Ⅱ-4」 -(1) 불만을 처리하기 위해 월별로 가장 많이 한 활동 내역은 무엇인가?

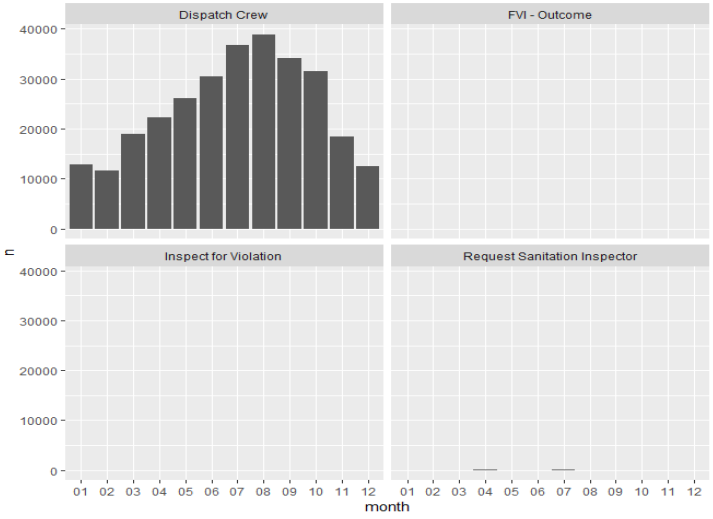
데이터 처리 과정에서 separate 함수를 통해 Creation Date, Current Activity, Most Recent Activity를 뽑아주어 rodent에 저장했으며, rename 함수를 통해 Current Activity는 Current.activity로 Most Recent Activity는 Most.recent.action로 바꿨다. missing value를 처리하기 위해 rodent<-filter(!is.na(Current.activity),!is.na(Most.recent.action))로 분석을 시작하였다.

->분석에 쓰인 변수
: Creation Date (불만이 제기된 날짜), Current Activity (불만이 제기된 후 활동 내역)

->분석방법

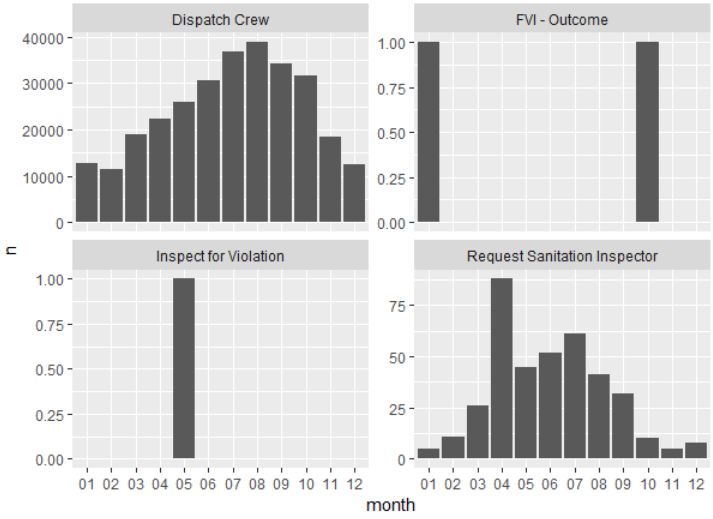
: Creation Date를 separate 함수를 통해 year, month, datetime으로 나눠준 후, 월별로 비교하기 위해 group_by 함수로 month와 Current.activity를 묶어주었고 그 빈도수를 판단하였다.

```
>rodent%>%group_by(month,Current.activity)%>%summarise(n=n())%>%ggplot(aes(month,n))+geom_bar(stat="identity")+facet_wrap(~Current.activity)
```



이 그래프의 경우 각각의 빈도수를 판단하기 어렵기 때문에 scales="free_y" 옵션을 넣어주어 각각의 빈도를 시각화시켰다.

```
>rodent%>%group_by(month,Current.activity)%>%summarise(n=n())%>%ggplot(aes(month,n))+geom_bar(stat="identity")+facet_wrap(~Current.activity,scales="free_y")
```



「Ⅱ-4」-(2) 불만을 처리하기 위해 월별로 어떤 방법을 가장 많이 사용했는가?

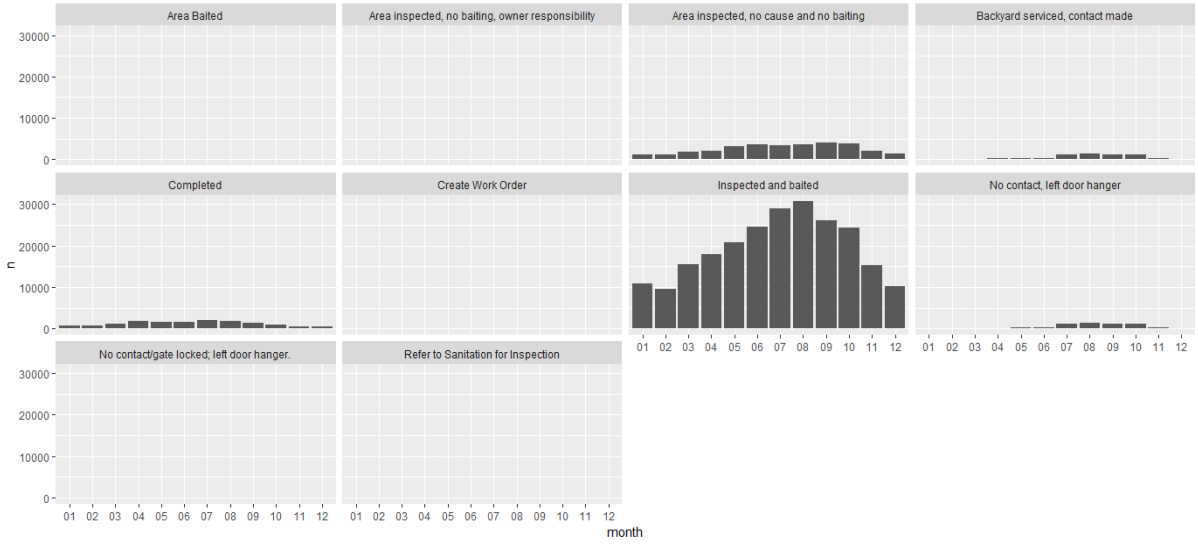
->분석에 쓰인 변수

: Creation Date (불만이 제기된 날짜), Most Recent Activity(불만이 제기된 후 처리 방법)

->분석방법

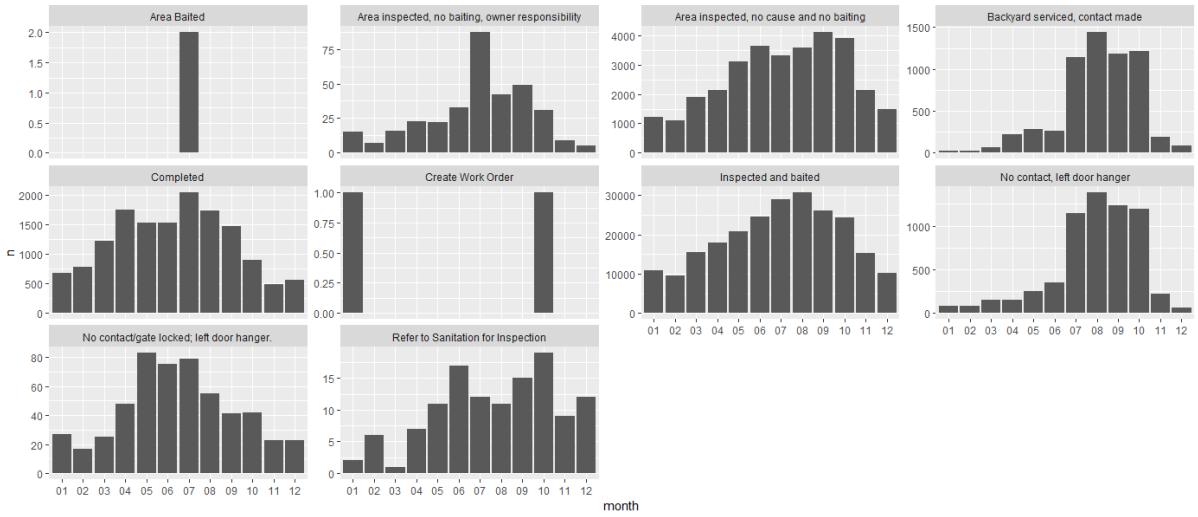
: Creation Date를 separate 함수를 통해 year, month, datetime으로 나눠준 후, 월별로 비교하기 위해 group_by 함수로 month와 Most.recent.action(불만이 제기된 후 처리 방법)를 묶어주었고 그 빈도수를 판단하였다.

```
>rodent%>%group_by(month,Most.recent.action)%>%summarise(n=n())%>%ggplot(aes(month,n))+geom_bar(stat="identity")+facet_wrap(~Most.recent.action)
```



: 이 그래프의 경우 각각의 빈도수를 판단하기 어렵기 때문에 scales="free_y" 옵션을 넣어주어 각각의 빈도를 시각화시켰다.

```
>rodent%>%group_by(month,Most.recent.action)%>%summarise(n=n())%>%ggplot(aes(month,n))+geom_bar(stat="identity")+facet_wrap(~Most.recent.action,scales="free_y")
```



『Ⅱ-5』 월별, 쥐 덫, 쓰레기, 쥐가 발견되는 구내 수의 분포가 어떻게 달라지는가? 또한, 분포를 보고 어떤 해결 방법이 필요한가?

「Ⅱ-5」 -(1) 쥐 덫이 발견된 구내 수의 월별 분포

월별 분포를 가장 정확하게 파악 할 수 있다고 결정한 날짜는 Creation.Date 이다. 사람들이 불반이 있을 때 말을 하기 때문에, 이 변수가 그나마 가장 정확하다는 결론이다. 날짜형식이 년도 월 날짜 시간순으로 하나로 합쳐져 있다.

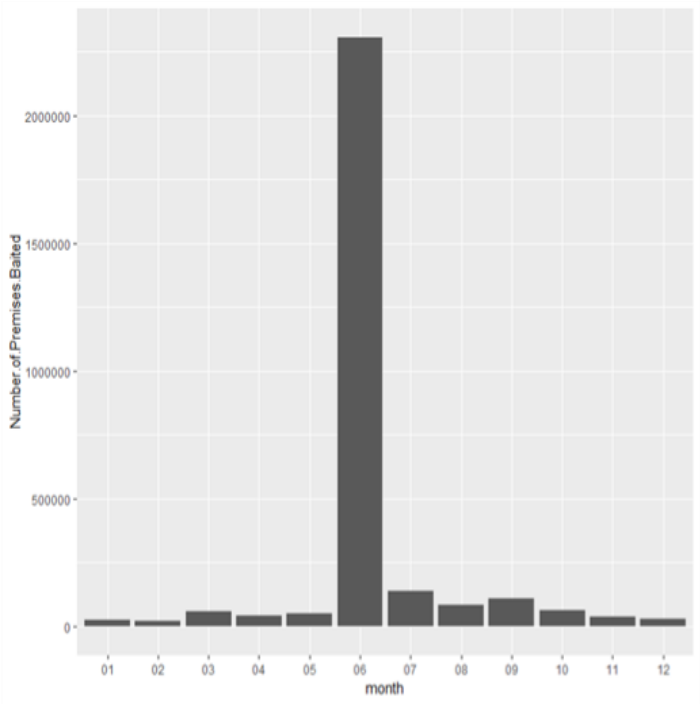
```
> head(rodent)
        Creation.Date
1 2011-01-01T00:00:00
2 2011-01-01T00:00:00
3 2011-01-01T00:00:00
4 2011-01-01T00:00:00
5 2011-01-01T00:00:00
6 2011-01-01T00:00:00
```

월만 나타내는 변수를 만들기 위해, separate 함수를 썼고, sep를 -로 지정하여 2번째 변수의 이름을 month로 지정하였다.

```
> rodent<-read.csv("rodent311.csv")
> rodent<-as_tibble(rodent)
> rodent<-rodent%>%separate(Creation.Date,into=c("year","month","datetime"),sep="-")
```

그 다음 ggplot의 geom_bar을 이용하여, x변수는 월 y변수는 쥐가 발견된 구내수로 하여 그래프를 그렸다. 여기에서 na.rm = TRUE로 한 이유는 NA값을 미리 없애고 그래프를 그려 오류를 범하지 않기 위함이다.

```
> ggplot(rodent,aes(month,Number.of.Premises.Baited))+geom_bar(stat="identity",na.rm=TRUE)
```

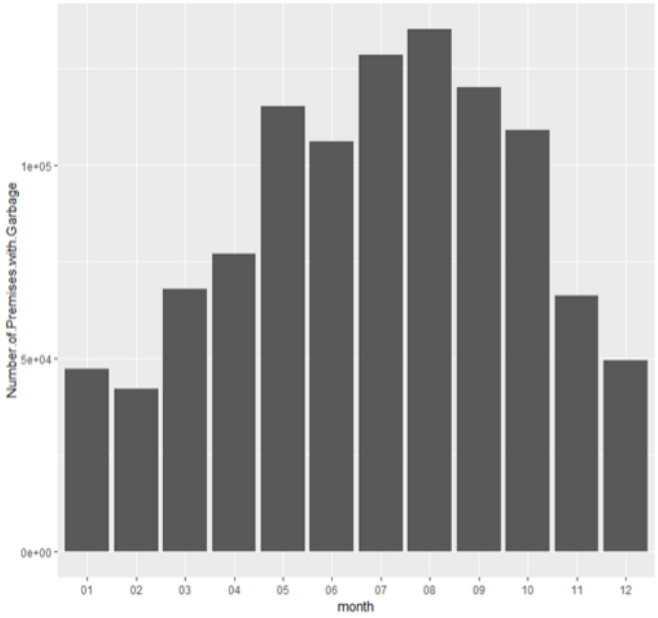


결과로는 6월에 이상하릴만큼 높고, 7월 9월 8월 10월 순으로 높았다.

「Ⅱ-5」-(2) 쓰레기가 발견 된 구내 수의 월별 분포

똑같은 방식으로 y 변수만 쓰레기가 발견 된 구내수로 바꾸고 ggplot을 그려본다.

```
> ggplot(rodent,aes(month,Number.of.Premises.with.Garbage))+geom_bar(stat="identity",na.rm=TRUE)
```

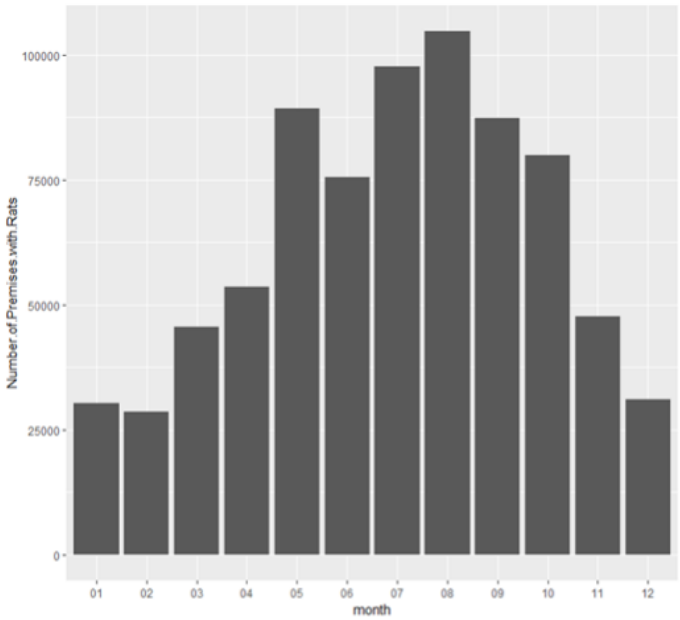


8월이 현저하게 높고, 7월 9월 5월 10월 순으로 높았다.

「Ⅱ-5」-(3) 쥐가 발견된 구내 수의 월별 분포

동일한 방법으로 y변수만 쥐가 발견된 구내 수로 바꾸고 ggplot을 돌려본다.

```
> ggplot(rodent,aes(month,Number.of.Premises.with.Rats))+geom_bar(stat="identity",na.rm=TRUE)
```



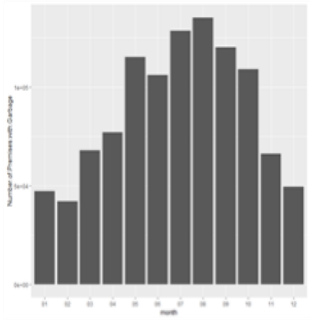
8월이 가장 많았고, 7월 5월 9월 10월 순으로 높았다.

결과를 정리해보면, 쥐덫은 6월만 너무 많고, 다른달은 조금만 설치했습니다. 또한, 쓰레기가 발견된 구내수와 쥐가 발견된 구내수의 월별 분포는 비슷하다는 것을 알 수 있다.

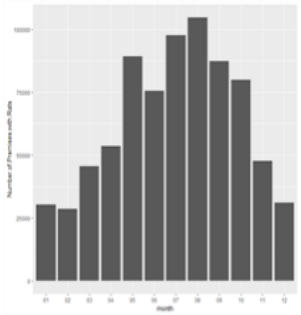
결론은 쥐의 발견이 그리 많지 않은 달에 쥐 덫을 가장 많이 설치한 점이 비효율 적이라는 것이다.

〈해결방안〉

쓰레기가 발견된 구내 수의 분포에 따라서, 월별 쥐 덫을 설치하는 양을 조절한다. 6월 달에만 많이 설치하기 보다는 쓰레기가 많고 쥐가 많은 8월, 7월, (9월 or 5월), 10월 순으로 더 많이 설치한다.



쓰레기가 발견된 구내 수



쥐가 발견된 구내 수

Ⅲ. 결론

「Ⅲ-1」 결과

각 경찰관할구역 별로 1구역이 쓰레기와 쥐가 있는 건물 부지수가 가장 적었는데, 오히려 평균 요청해결기간이 가장 오래 걸린 구역은 1구역(15.6일), 가장 빠른 구역은 5구역 (10일)

→ 쓰레기와 쥐가 적기 때문에, 해결 시스템을 활용한 경우가 적고 솔루션이 잘 갖추어지지 않아서, 요청해결기간이 오래 걸린 것으로 판단됨.

「Ⅲ-2」 해결 방안

- 쥐덫은 6월에 가장 많고, 쓰레기가 발견된 구내 수의 분포에 따라 월별 쥐덫을 설치하는 양을 조절하면 Good!
- 불만을 처리하기 위해 가장 많이 한 활동 내역은 “Dispatch Crew”, 사용방법으로는 “Inspected and baited” ⇒ 적극활용하자!