Road Safety Data 가설설정 및 분석

6조 보통사람들

1533005 곽나영
1533021 이예진
1602050 박지원
1602069 예지혜
1602099 이혜상

목차

CONTENTS 

서론 1) 자료 소개 및 정리

2017UKRoadSafetyData

[2017 UK Road Safety Data]

▶ 2017년 UK에서 발생한 도로 위 사고 관측 자료

▶ Accident Data : 관측치 수 – 129,983개
변수 개수 – 32개

▶ Vehicle Data : 관측치 수 – 170,994개
변수 개수 – 16개

▶ Casualty Data : 관측치 수 – 238,927개
변수 개수 – 23개

서론 1) 자료 소개 및 정리

2017UKRoadSafetyData

▶ 자료 선택
select 함수 이용해서 필요한 변수 추출

연속형 변수 : 〈chr〉

```
> acc
# A tibble: 129,982 x 11
   Accident_Index Accident_Severity Number_of_Vehicles Date       Time  Road_Type Speed_limit Light_Conditions Weather_Conditions Road_Surface_Conditions Urban_or_Rural_Area
   <chr>          <chr>             <chr>              <chr>      <chr> <chr>     <chr>       <chr>            <chr>              <chr>                  <chr>
 1 2017010001708  1                 2                  05/08/2017 03:12 6         30          4                1                  1                      1
 2 2017010009342  3                 2                  01/01/2017 01:30 6         30          4                1                  2                      1
 3 2017010009344  3                 3                  01/01/2017 00:30 6         30          4                1                  1                      1
 4 2017010009348  3                 2                  01/01/2017 01:11 1         30          4                2                  2                      1
 5 2017010009350  2                 1                  01/01/2017 01:42 3         20          4                1                  2                      1
 6 2017010009351  3                 2                  01/01/2017 03:31 6         30          4                1                  2                      1
 7 2017010009353  3                 2                  01/01/2017 04:07 3         40          4                1                  2                      1
 8 2017010009354  3                 2                  01/01/2017 05:20 3         30          4                2                  2                      1
 9 2017010009357  2                 1                  01/01/2017 03:18 3         50          4                2                  2                      1
10 2017010009358  2                 1                  01/01/2017 03:00 6         30          4                1                  2                      1
# ... with 129,972 more rows
```

▶ 변수 type 설정
default = character로 불러온 후,
type_convert 함수 이용하여 변수 type변경

```
> acc<- read_csv("Acc 2017 data.csv",col_types= cols(.default = col_character()))
```

```
> type_convert(acc)
Parsed with column specification:
cols(
  Accident_Index = col_character(),
  Accident_Severity = col_integer(),
  Number_of_Vehicles = col_integer(),
  Date = col_character(),
  Time = col_time(format = ""),
  Road_Type = col_integer(),
  Speed_limit = col_integer(),
  Light_Conditions = col_integer(),
  Weather_Conditions = col_integer(),
  Road_Surface_Conditions = col_integer(),
  Urban_or_Rural_Area = col_integer()
)
```

```
# A tibble: 129,982 x 11
   Accident_Index Accident_Severity Number_of_Vehicles Date       Time     Road_Type Speed_limit Light_Conditions Weather_Conditions Road_Surface_Conditions Urban_or_Rural_Area
   <chr>                     <int>              <int> <chr>      <time>       <int>       <int>            <int>              <int>                  <int>                <int>
 1 2017010001708                 1                  2 05/08/2017 03:12         6          30                4                  1                      1                    1
 2 2017010009342                 3                  2 01/01/2017 01:30         6          30                4                  1                      2                    1
 3 2017010009344                 3                  3 01/01/2017 00:30         6          30                4                  1                      1                    1
 4 2017010009348                 3                  2 01/01/2017 01:11         1          30                4                  2                      2                    1
 5 2017010009350                 2                  1 01/01/2017 01:42         3          20                4                  1                      2                    1
 6 2017010009351                 3                  2 01/01/2017 03:31         6          30                4                  1                      2                    1
 7 2017010009353                 3                  2 01/01/2017 04:07         3          40                4                  1                      2                    1
 8 2017010009354                 3                  2 01/01/2017 05:20         3          30                4                  2                      2                    1
 9 2017010009357                 2                  1 01/01/2017 03:18         3          50                4                  2                      2                    1
10 2017010009358                 2                  1 01/01/2017 03:00         6          30                4                  1                      2                    1
# ... with 129,972 more rows
```

서론 1) 자료 소개 및 정리

2017UKRoadSafetyData

▶ 자료 통합 방법

1) 동일한 사고에 관한 자료임에도 관측치 수가 다른 이유

Accident Data : 관측치 수 – 129,983개
Vehicle Data : 관측치 수 – 170,994개
Casualty data : 관측치 수 – 238,927개

Accident : 사고에 관한 자료
Vehicle : 사고 발생과 관련한 모든 탈 것
Casualty : 사고와 관련된 모든 사상자

Ex) 사고 하나에 사상자가 여러 명, 이중추돌사고

→ **Accident Data ⟨ Vehicle Data**
Accident Data ⟨ Casualty Data

2) Accident index가 유일한 acc에 veh, cas를 left_join으로 붙임
=⟩ acc_veh & acc_cas

acc

	Accident_Index
	<chr>
1	2017010001708
2	2017010009342
3	2017010009344
4	2017010009348
5	2017010009350
6	2017010009351
7	2017010009353
8	2017010009354
9	2017010009357
10	2017010009358

+

veh

	Accident_Index
	<chr>
1	2017010001708
2	2017010001708
3	2017010009342
4	2017010009342
5	2017010009344
6	2017010009344
7	2017010009344
8	2017010009348
9	2017010009348
10	2017010009350

=

acc_veh

	Accident_Index
	<chr>
1	2017010001708
2	2017010001708
3	2017010009342
4	2017010009342
5	2017010009344
6	2017010009344
7	2017010009344
8	2017010009348
9	2017010009348
10	2017010009350

서론 2) 변수 소개 – Accident

2017 UK Road Safety Data

이름	유형	설명
Accident_Severity	[범주형]	사고의 심각도
Date	[연속형]	사고 발생 날짜
Time	[연속형]	사고 발생 시각
Road_Type	[범주형]	도로 유형
Speed_limit	[정수형]	제한속도
Light_Conditions	[범주형]	사고 당시 빛의 밝기
Weather_Conditions	[범주형]	사고 당시 기상조건
Road_Surface_Conditions	[범주형]	도로의 지면 상태
Urban_or_Rural_Area	[범주형]	도시/시골

서론 2) 변수 소개 – Vehicles

2017 UK Road Safety Data

이름	유형	설명
Vehicle_Type	[범주형]	차량 유형
Number_of_Vehicles	[정수형]	사고 차량 수
Vehicle_Manoeuvre	[범주형]	사고 당시의 차량의 움직임
Junction_Location	[범주형]	교차로(합류점)의 위치
1st_Point_of_Impact	[범주형]	충돌지점
Was_Vehicle_ Left_Hand_Drive	[범주형]	운전자의 왼손잡이 여부
Sex_of_Driver	[범주형]	운전자의 성별
Age_of_Driver	[연속형]	운전자의 연령
Engine_of_Capacity_(CC)	[연속형]	차량 엔진의 배기량
Age_of_Vehicle	[연속형]	차량의 연식

서론 2) 변수 소개 – Casualties

2017 UK Road Safety Data

이름	유형	설명
Casualty_Class	[범주형]	운전자와 보행자, 승객 중 사상자의 유형
Sex_of_Casualty	[범주형]	사상자의 성별
Age_of_Casualty	[연속형]	사상자의 나이
Age_Band_of_Casualty	[범주형]	사상자의 연령대
Casualty_Severity	[범주형]	사상자의 사상 정도
Casualty_Type	[범주형]	사상자가 사고 당시 이용하던 교통수단 또는 도보

2. 본론 : 가설분석

2017UKRoadSafetyData 

Part1

1) 월별
2) 시간별

Part2

1) 운전자 성별 연령별
2) 왼손잡이 운전자

Part3

1) 차량의 연식
2) 사고 난 당시의 차량 개수
3) 도시/시골 사고의 차량 종류

Part1. 1) 월별

① 월별 boxplot 그리기

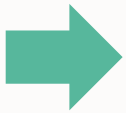
▶ Month 별로 사고 횟수를 분석하여 몇월 달에 사고가 많이 발생하는지 분석

▶ acc data의 Date 변수

```
> acc%>%select(Date)
# A tibble: 129,982 x 1
   Date
   <chr>
 1 05/08/2017
 2 01/01/2017
 3 01/01/2017
 4 01/01/2017
 5 01/01/2017
 6 01/01/2017
 7 01/01/2017
 8 01/01/2017
 9 01/01/2017
10 01/01/2017
# ... with 129,972 more rows
```

▶ "%d / %m /%y" 의 형태로 입력

▶ 월별 사고 건수 분석을 위해
Year, Month, Day 변수로 분리



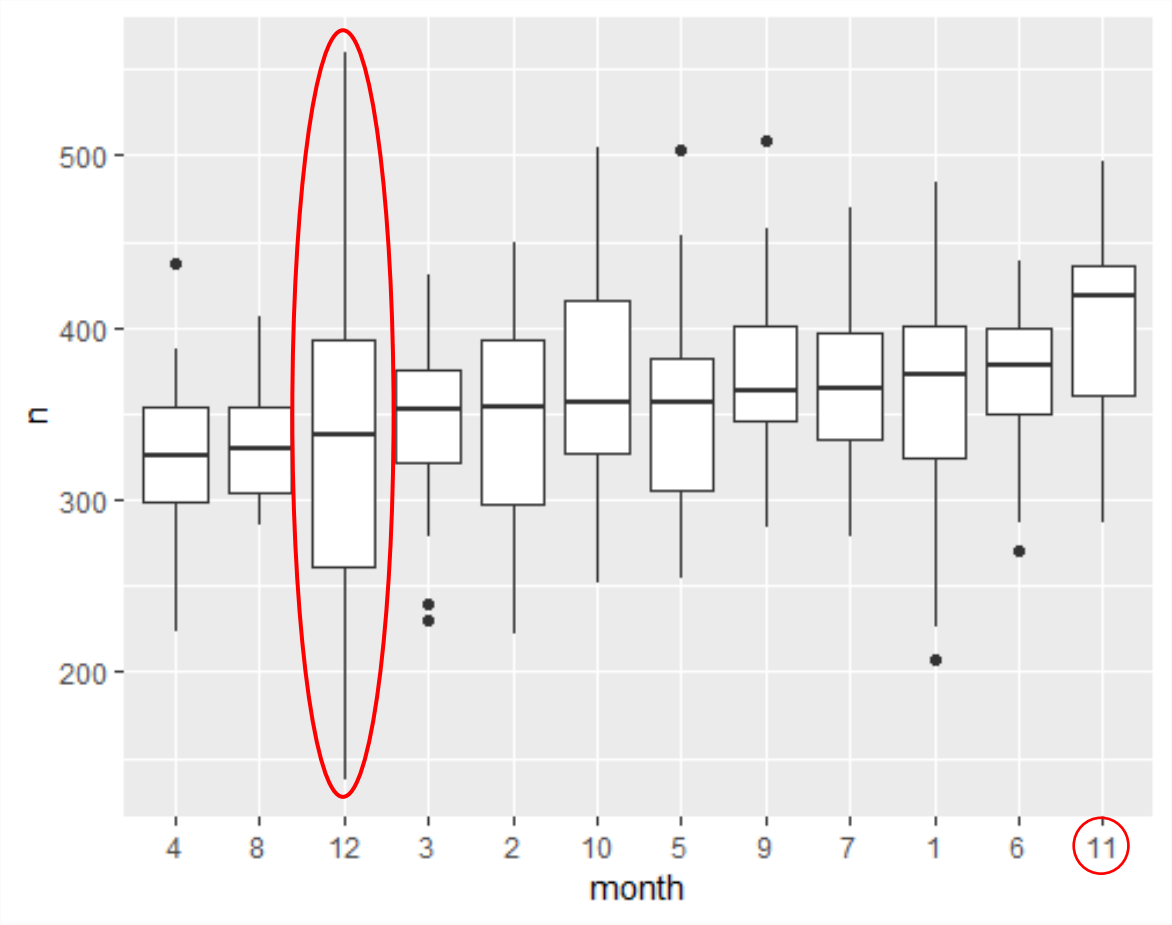
```
> acc_date<-acc%>%
+     filter(!is.na(Date),!is.na(Time))%>%
+     separate(Date,c("day","month","year"),sep="/",convert=T)%>%
+     group_by(year,month,day)%>%summarise(n=n())
> acc_date
# A tibble: 365 x 4
# Groups:   year, month [?]
    year month   day     n
   <int> <int> <int> <int>
 1  2017     1     1   242
 2  2017     1     2   291
 3  2017     1     3   301
 4  2017     1     4   334
 5  2017     1     5   437
 6  2017     1     6   373
 7  2017     1     7   257
 8  2017     1     8   226
 9  2017     1     9   424
10  2017     1    10   402
# ... with 355 more rows
```

Part1. 1) 월별

① 월별 boxplot 그리기

▶ acc_date 변수 이용한 월별 boxplot

▶ 월별 median을 중심으로 box 순서 재배열



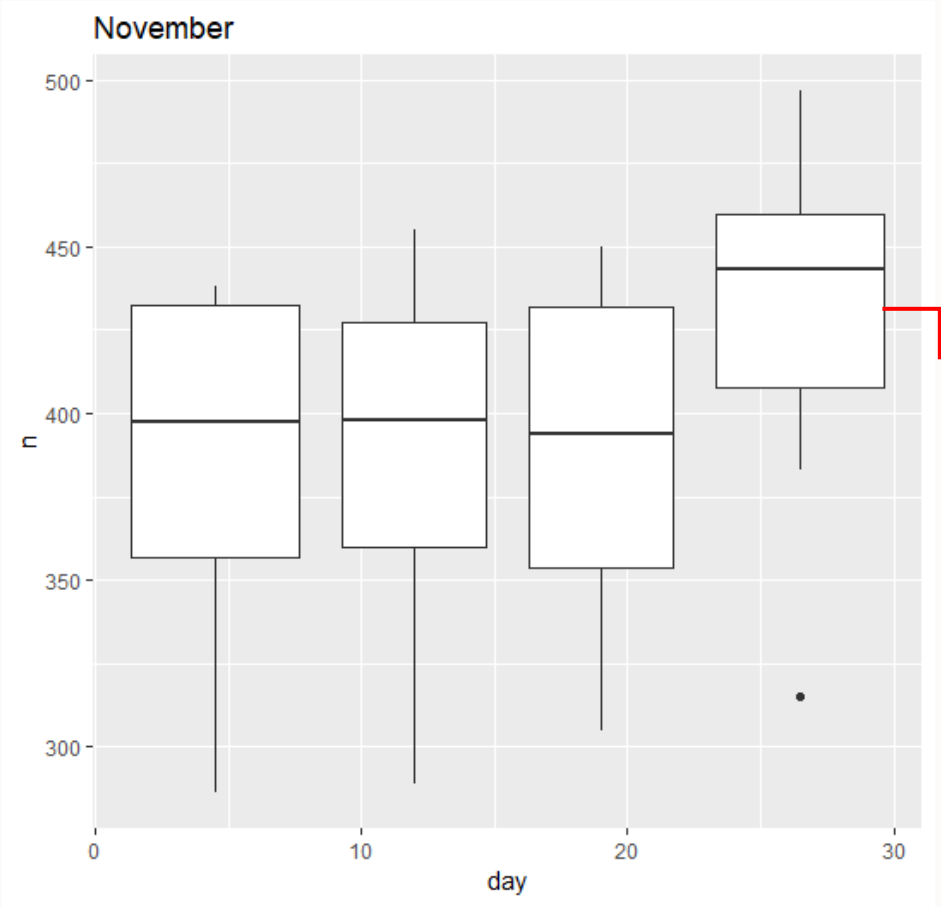
▶ 11월의 사고 건수의 median 가장 큼

▶ 12월의 variance 유독 넓게 퍼져있음

Part1. 1) 월별

② 11월 주차별 사고 건수

▶ 11월의 주차별 사고 건수 Boxplot

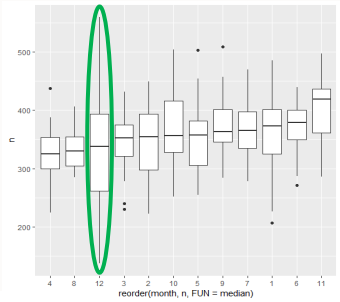


▶ 11월 마지막 주의 사고건수 가장 많음

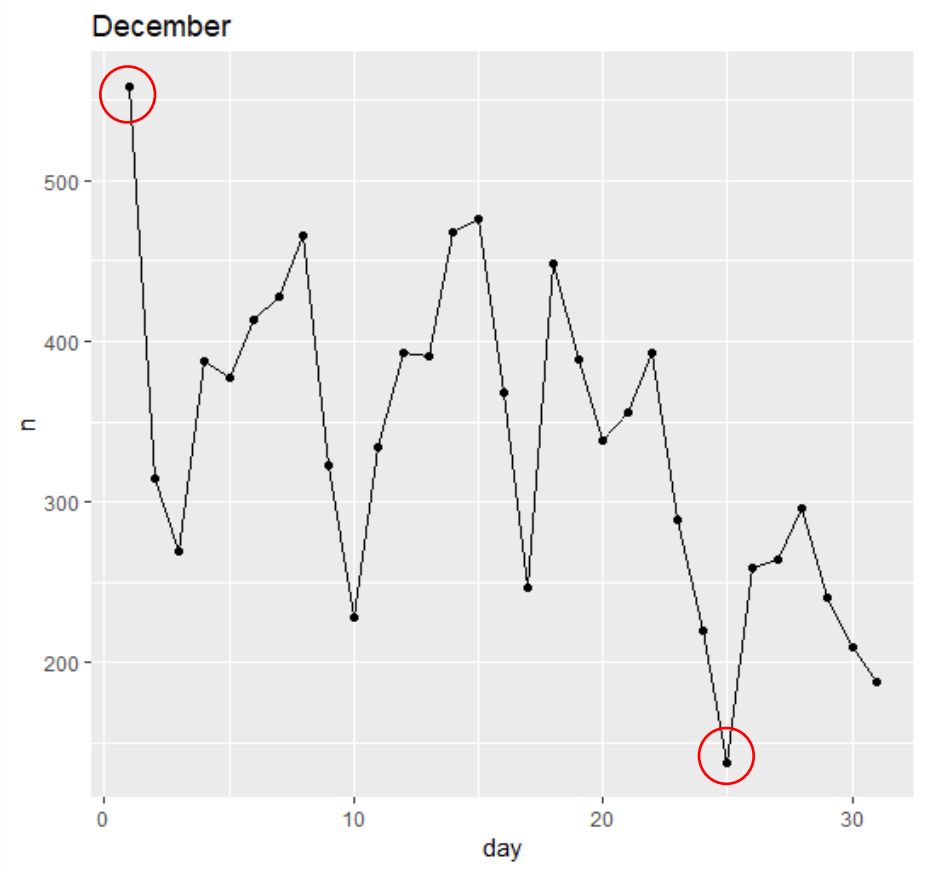
Part1. 1) 월별

③ 12월 일별 사고 변화량

▶ 12월 : range (min , max) 가장 넓음



▶ Day별 사고 건수 파악




```
> acc_date%>%
+     filter(month=='12')%>%
+     mutate(rank=rank(desc(n)))%>%
+     filter(rank %in% range(rank))
# A tibble: 2 x 5
# Groups:   year, month [1]
   year month   day     n  rank
  <int> <int> <int> <int> <dbl>
1  2017    12     1   559     1
2  2017    12    25   137    31
```

▶ 12월 1일 : 사고 건수 가장 많음

▶ 12월 25일 : 사고 건수 가장 적음

Part1. 1) 월별

④ 시간별 사고 건수

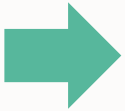
▶ Hour별로 사고 횟수를 분석하여 몇 시경에 사고가 많이 발생하는지 분석

▶ acc data의 Time 변수

```
> acc%>%select(Time)
# A tibble: 129,982 x 1
   Time
   <time>
 1 03:12
 2 01:30
 3 00:30
 4 01:11
 5 01:42
 6 03:31
 7 04:07
 8 05:20
 9 03:18
10 03:00
# ... with 129,972 more rows
```

▶ "%H : %M " 의 형태로 입력

▶ 시간 별 분석을 위해
Hour , Minute변수로 분리



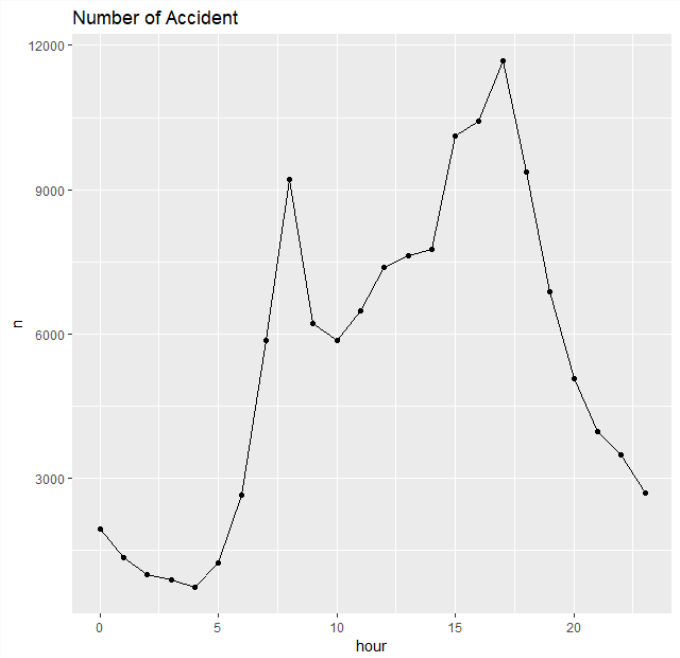
```
> acc_time<-acc%>%
+     filter(!is.na(Time))%>%select(Time)%>%
+     separate(Time,c("hour","minute"),sep=":",convert=T)
> acc_time
# A tibble: 129,979 x 2
    hour minute
   <int>  <int>
 1     3     12
 2     1     30
 3     0     30
 4     1     11
 5     1     42
 6     3     31
 7     4      7
 8     5     20
 9     3     18
10     3      0
# ... with 129,969 more rows
```

Part1. 2) 시간별

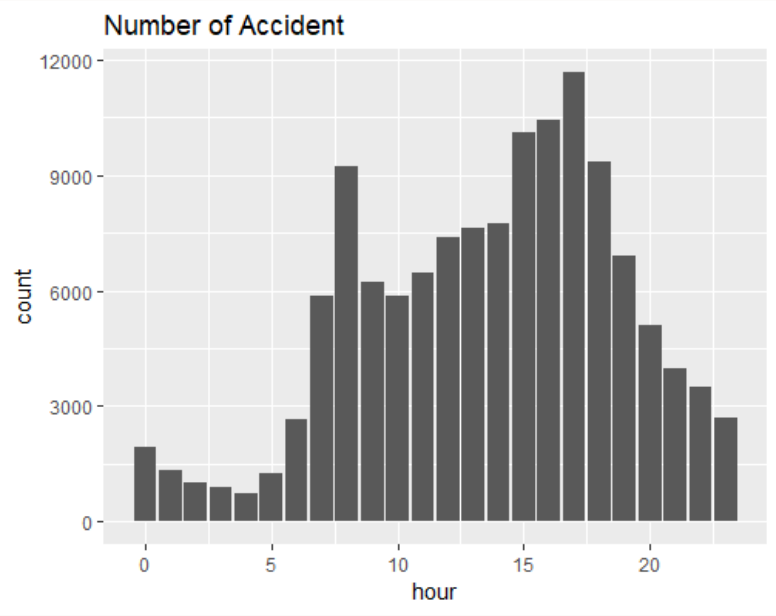
① 시간별 사고 건수

```
> table(acc_time$hour)

    0     1     2     3     4     5     6     7     8     9    10    11    12    13    14    15    16    17    18    19    20    21    22    23
1949  1352  1008   887   736  1252  2647  5876  9220  6228  5867  6474  7394  7636  7751 10107 10424 11681  9365  6890  5085  3968  3490  2692
```

▶ hour별 사고 건수의 freqpoly



▶ hour별 사고 건수의 histogram



▶ 17시 (오후 5시) 사고 건수 가장 많음

▶ 출퇴근 시간의 교통량이 증가하므로 사고 건수가 다른 시간에 비해 상대적으로 많음

▶ 00시-06시 새벽에는 운전을 하는 사람이 많지 않기 때문에 상대적으로 사고 건수도 매우 적음

2. 본론 : 가설분석

2017UKRoadSafetyData 

Part1

1) 월별
2) 시간별

Part2

1) 운전자 성별 연령별
2) 왼손잡이 운전자

Part3

1) 차량의 연식
2) 사고 난 당시의 차량 개수
3) 도시/시골 사고의 차량 종류

Part2. 1) 운전자별

① 운전자 성별 연령별 사고 건수

▶ 운전자의 성별과 연령별 사고 횟수를 분석하여, 어느 그룹에서 사고가 많이 발생하는지 분석

▶ veh data의 Sex_of_Driver , Age_of_Driver 변수

```
> veh%>%select(Sex_of_Driver,Age_of_Driver)
# A tibble: 238,926 x 2
   Sex_of_Driver Age_of_Driver
           <int>          <int>
 1             1             24
 2             1             19
 3             1             33
 4             1             40
 5             3             -1
 6             1             35
 7             2             31
 8             2             37
 9             2             29
10             1             78
# ... with 238,916 more rows
```

▶ Sex_of_Driver 변수

1 : Male

2 : Female

3 : Not known

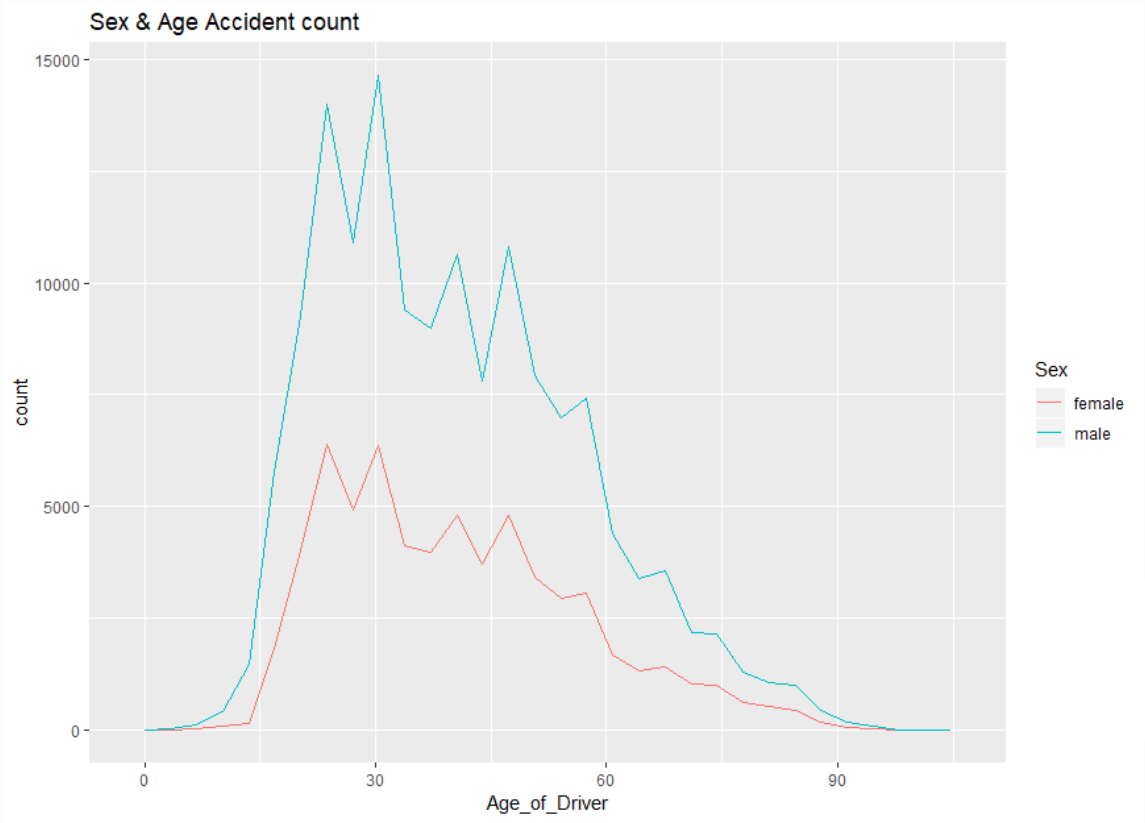
〈int〉 ---〉 〈chr〉 변수변환

```
> veh_agesex <-
+     veh %>% filter(Sex_of_Driver %in% 1:2 & Age_of_Driver>=1) %>%
+     mutate(Sex=ifelse(Sex_of_Driver==1,"male","female"))%>%
+     select(Sex,Age_of_Driver)
> veh_agesex
# A tibble: 209,349 x 2
   Sex    Age_of_Driver
   <chr>          <int>
 1 male              24
 2 male              19
 3 male              33
 4 male              40
 5 male              35
 6 female            31
 7 female            37
 8 female            29
 9 male              78
10 male              19
# ... with 209,339 more rows
```

Part2. 1) 운전자별

① 운전자 성별 연령별 사고 건수

▶ 운전자 Sex 별로 Age에 따른 사고 건수 분석




```
> table(veh_agesex$Sex)

female   male
 62998 146351
```

▶ Male이 Female에 비해 사고 빈도가 상대적으로 많음

▶ 정확한 분포 비교를 위해 density plot 그려보기

Part2. 1) 운전자별

② 운전자 성별 연령별 사고 패턴(Pattern)

▶ 운전자 Sex별로 Age에 따른 사고 건수 분석

▶ 남녀 운전자의 나이에 따른 사고 발생 pattern 거의 일치

▶ 25-30세에 가장 많은 사고 발생

Part2. 2) 왼손잡이 운전자

왼손잡이인지 오른손잡이인지에 따라 첫 번째 충돌지점의 차이가 있을 것이다

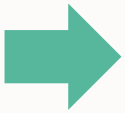
▶ 왼손잡이는 nearside, 오른손잡이는 offside의 1st point of impact 빈도가 더 높을 것이다.
(긴박한 상황에 핸들을 꺾는 방향이 다르기 때문에 차이가 있을 것이다.)

▶ veh data의 Left handed , 1st point of impact 변수

```
> veh%>%select(`Was_Vehicle_Left_Hand_Drive?`, `1st_Point_of_Impact`)
# A tibble: 238,926 x 2
   `Was_Vehicle_Left_Hand_Drive?` `1st_Point_of_Impact`
                            <int>                 <int>
 1                              1                     1
 2                              1                     2
 3                              1                     2
 4                              1                     1
 5                              1                     1
 6                              1                     1
 7                              1                     1
 8                              1                     1
 9                              1                     3
10                              1                     3
# ... with 238,916 more rows
```

▶ Was_Vehicle_Left_Hand_Drive 변수
1 : No
2 : Yes

⟨int⟩ ---⟩ ⟨chr⟩ 변수변환

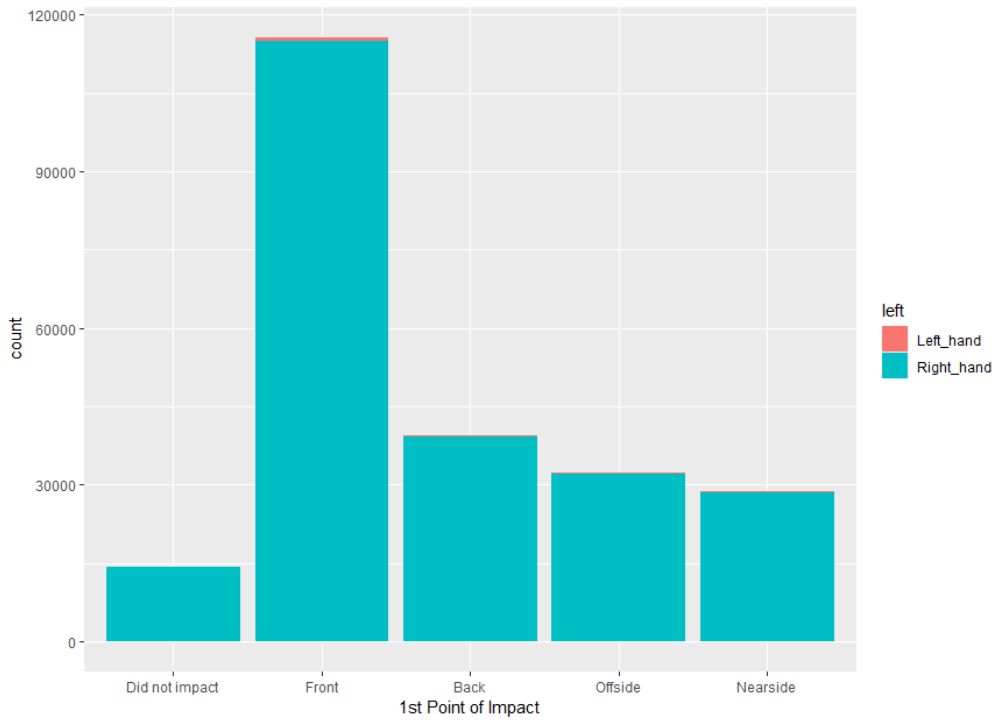


```
> veh_left<-veh %>%
+ filter(`Was_Vehicle_Left_Hand_Drive?` >=1, `1st_Point_of_Impact`>=1) %>%
+ mutate(left=ifelse(`Was_Vehicle_Left_Hand_Drive?`==1,"Right_hand","Left_hand"))%>%
+ select(left,`1st_Point_of_Impact`)
> veh_left
# A tibble: 216,163 x 2
   left       `1st_Point_of_Impact`
   <chr>                      <int>
 1 Right_hand                     1
 2 Right_hand                     2
 3 Right_hand                     2
 4 Right_hand                     1
 5 Right_hand                     1
 6 Right_hand                     1
 7 Right_hand                     1
 8 Right_hand                     1
 9 Right_hand                     3
10 Right_hand                     3
# ... with 216,153 more rows
```

Part2. 2) 왼손잡이 운전자

왼손잡이인지 오른손잡이인지에 따라 첫 번째 충돌지점의 차이가 있을 것이다

```
> veh_left %>%
    ggplot(aes(as.factor(`1st_Point_of_Impact`),fill=left)) +geom_bar() +
    scale_x_discrete(labels=c("Did not impact","Front","Back","Offside" ,"Nearside"))
    +xlab("`1st_Point_of_Impact`")
```




```
> table(veh_left$left)

Left_hand Right_hand
     1616     214547
```

▶ Right hand가 Left hand에 비해 사고 빈도가 상대적으로 많음

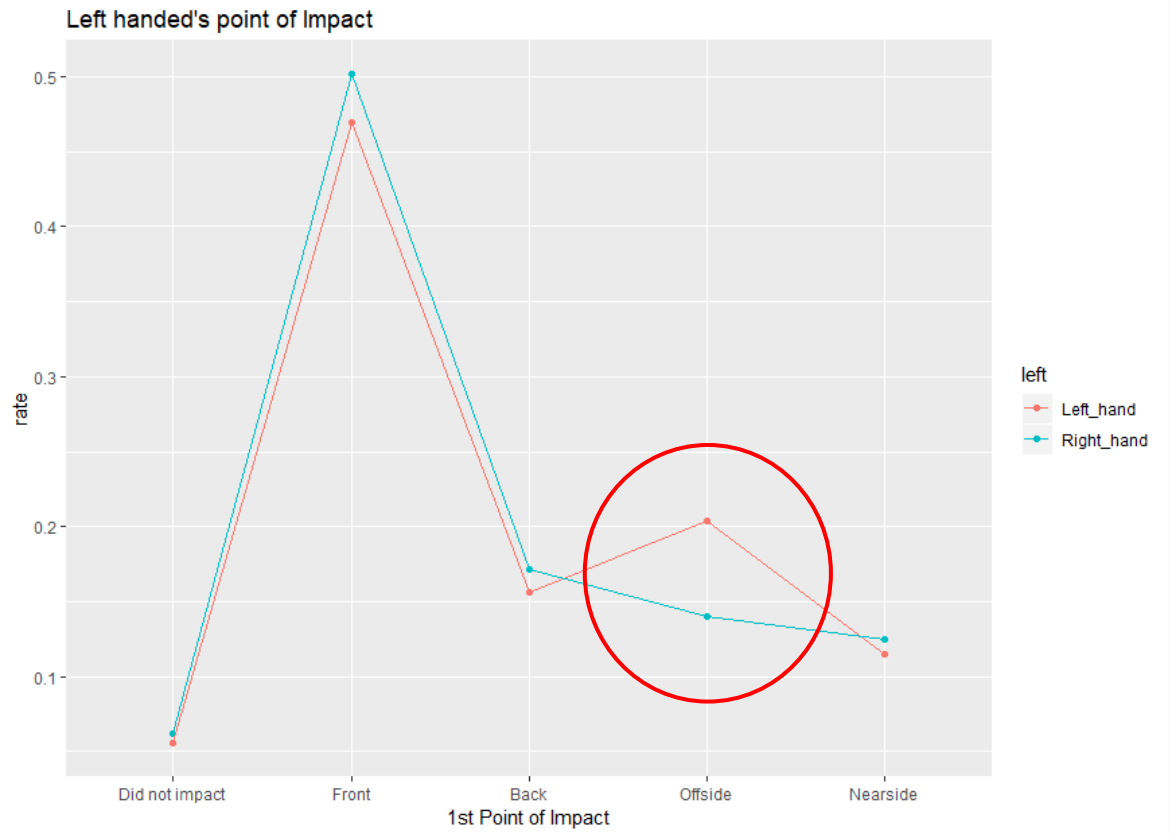
▶ 정확한 분포 비교를 위해 group 지정 필요

Part2. 2) 왼손잡이 운전자

왼손잡이인지 오른손잡이인지에 따라 첫 번째 충돌지점의 차이가 있을 것이다

▶ Grouping by Left hand , Right hand 후, 각 Group별 1st point of Impact의 rate 계산

```
> veh_rate<-veh_left %>%
+     group_by(left,`1st_Point_of_Impact`)%>%summarise(count=n())%>%
+     mutate(sum=sum(count),rate= count/sum)
> veh_rate
# A tibble: 8 x 5
# Groups:   left [2]
  left       `1st_Point_of_Impact`  count    sum  rate
  <chr>                      <int>  <int>  <int> <dbl>
1 Left_hand                      1    803   1616 0.497
2 Left_hand                      2    267   1616 0.165
3 Left_hand                      3    349   1616 0.216
4 Left_hand                      4    197   1616 0.122
5 Right_hand                     1 114838 214547 0.535
6 Right_hand                     2  39136 214547 0.182
7 Right_hand                     3  31987 214547 0.149
8 Right_hand                     4  28586 214547 0.133
```



▶ Offside의 경우에만 Right hand보다 Left hand의 사고 발생 비율이 훨씬 더 높다

2. 본론 : 가설분석

2017UKRoadSafetyData 

Part1

1) 월별
2) 시간별

Part2

1) 운전자 성별 연령별
2) 왼손잡이 운전자

Part3

1) 차량의 연식
2) 사고 난 당시의 차량 개수
3) 도시/시골 사고의 차량 종류

Part3. 1) 차량의 연식

차량의 연식이 높을수록 사고 횟수가 많을 것이다

▶ veh data의 Age_of_Vehicle 변수를 이용

```
> table(veh$Age_of_Vehicle)

   -1     1     2     3     4     5     6     7     8     9    10    11    12    13    14    15    16    17
61634 16084 14424 12950 11796 10952  9962 10017  9597 10528 11559 10881 10498  9654  8255  6865  4697  2841
   18    19    20    21    22    23    24    25    26    27    28    29    30    31    32    33    34    35
 1843  1125   751   465   329   237   170   121   127   107    78    59    45    27    28    29    27    13
   36    37    38    39    40    41    42    43    44    45    46    47    48    49    50    51    52    53
    8    18    15    11     4    11     6     5     1     6     1     2     3     2     3     4     3     4
   54    55    56    57    58    59    60    61    62    64    65    69    70    74    75    78    85
    5     3     3     8     3     3     1     5     2     1     2     1     1     2     2     1     1
```

▶ missing data를 제거하여 새로운 데이터셋 veh1 생성

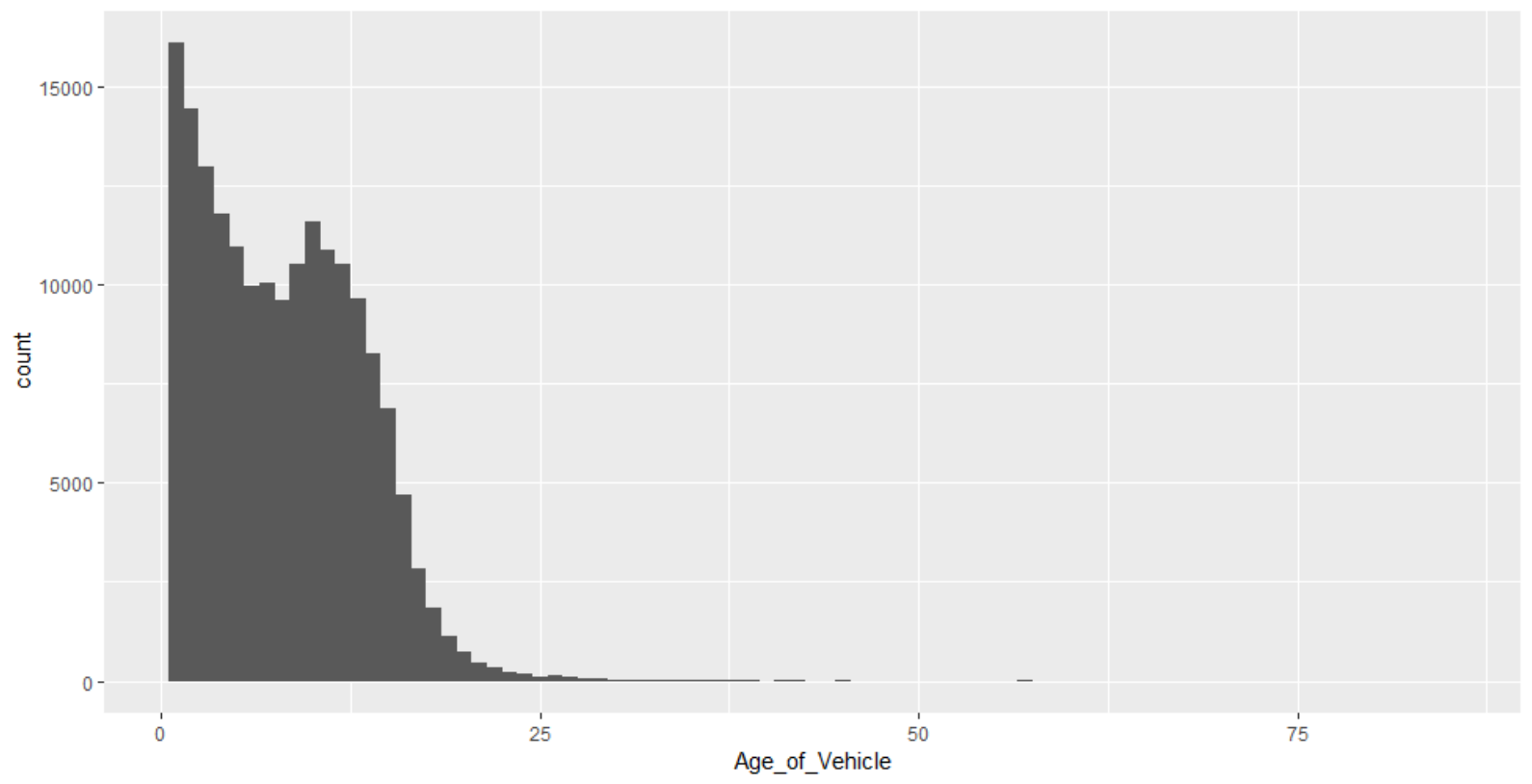
```
> veh1<-veh %>% filter(Age_of_Vehicle >= 0)
> summary(veh1$Age_of_Vehicle)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   4.000   8.000   8.109  12.000  85.000
```

Part3. 1) 차량의 연식

차량의 연식이 높을수록 사고 횟수가 많을 것이다

▶ 차량의 연식별 사고 발생 횟수 분석

```
> veh1 %>% ggplot(aes(Age_of_Vehicle)) + geom_histogram(binwidth = 1)
```



▶ 차량의 연식이 일정 부분을 넘어가면 빈도 수가 급격히 감소한다.
이는 연식이 오래된 차량 자체가 적기 때문으로 보인다.

Part3. 1) 차량의 연식

차량의 연식이 높을수록 사고 횟수가 많을 것이다

▶ 사고 건수가 1000건 이상인 데이터만을 분석

```
> veh1 %>% count(Age_of_Vehicle) %>% filter(n<1000)
# A tibble: 51 x 2
   Age_of_Vehicle     n
            <int> <int>
 1             20   751
 2             21   465
 3             22   329
 4             23   237
 5             24   170
 6             25   121
 7             26   127
 8             27   107
 9             28    78
10             29    59
# ... with 41 more rows
```

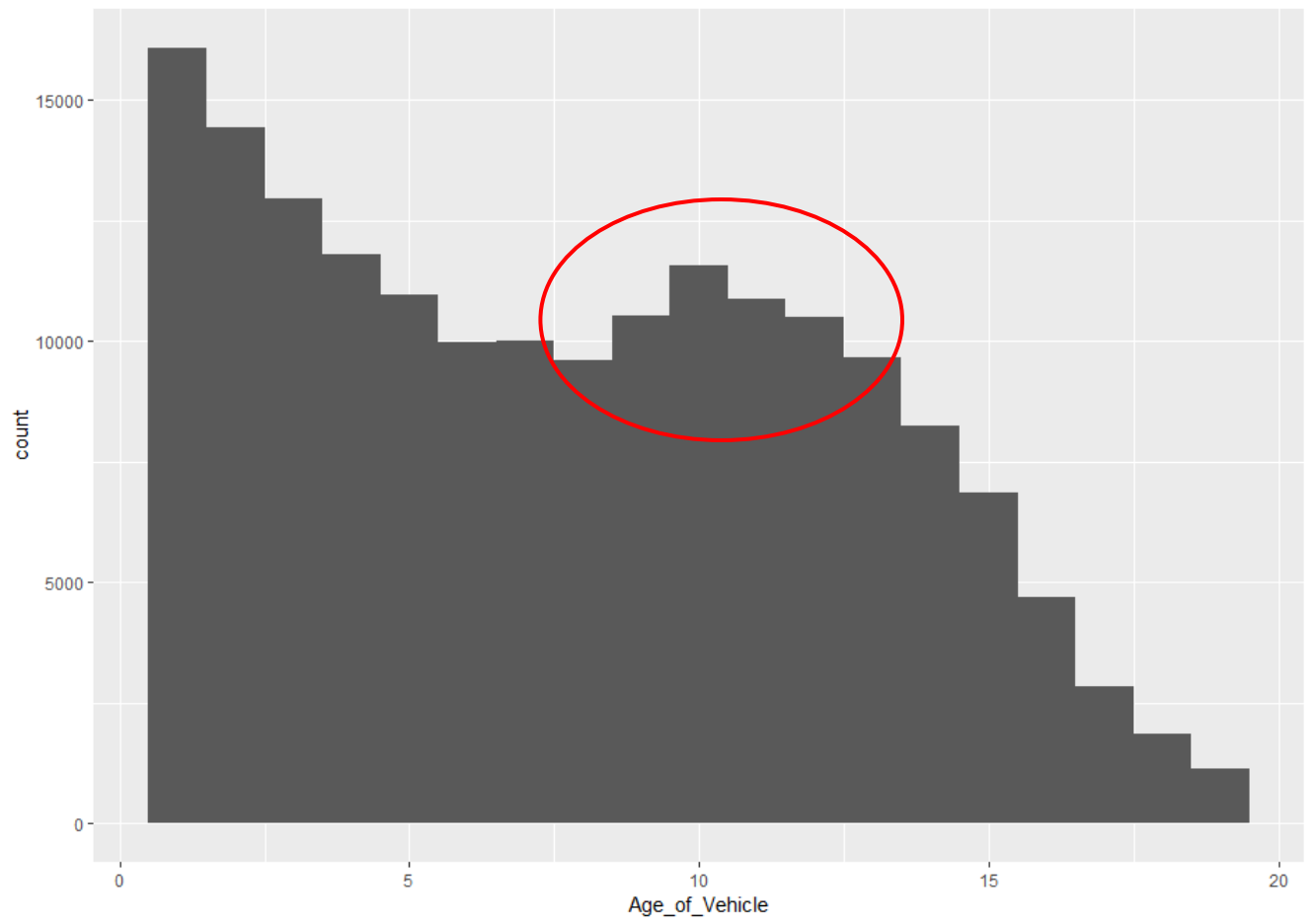
▶ 차량의 연식이 20년을 넘어가면서, 사고 발생은 1000건 이하로 나타난다.
따라서, 연식이 20년 안 된 차량의 경우만 분석해보겠다.

Part3. 1) 차량의 연식

차량의 연식이 높을수록 사고 횟수가 많을 것이다

▶ 차량의 연식이 20년 미만일 때

```
> veh1 %>% filter(Age_of_Vehicle < 20) %>% ggplot(aes(Age_of_Vehicle)) + geom_histogram(binwidth = 1)
```



▶ 대체로 연식이 높을수록 사고 횟수가 감소하는 경향을 보임

▶ 연식이 높아질수록 절대적인 대수 자체가 감소하는 것의 영향을 받은 것으로 보임

▶ 그럼에도 불구하고 급격한 감소 경향은 운전자의 숙련도와 관련 있어 보임

▶ 경향을 거슬러 차량이 9년~10년 되었을 때 사고 빈도 증가. 차량의 연식이 9년을 넘어가면, 상대적으로 사고가 발생하기 쉬움

Part3. 2) 차량의 개수

사고 난 차량 개수가 많을수록 사고의 심각성이 높을 것이다

▶ Veh data의 **Number_of_Vehicles, Accident_Severity 변수를 이용**

▶ Veh data의 **Accident_Severity 변수**

```
> acc %>% select(Number_of_Vehicles,Accident_Severity)
# A tibble: 129,982 x 2
   Number_of_Vehicles Accident_Severity
                <int>             <int>
 1                  2                 1
 2                  2                 3
 3                  3                 3
 4                  2                 3
 5                  1                 2
 6                  2                 3
 7                  2                 3
 8                  2                 3
 9                  1                 2
10                  1                 2
# ... with 129,972 more rows
```

```
> label<-tibble(Accident_Severity=c(1,2,3),Accident_Severity_Label=c("fatal","serious","slight"))
> label
# A tibble: 3 x 2
  Accident_Severity Accident_Severity_Label
              <dbl> <chr>
1                 1 fatal
2                 2 serious
3                 3 slight
```

▶ Accident_Severity 변수
  1 : Fatal
  2 : Serious
  3 : Slight

▶ ⟨int⟩ → ⟨chr⟩ 변환

▶ left_join 이용하여 새로운 변수, Accident_Severity_Label 생성

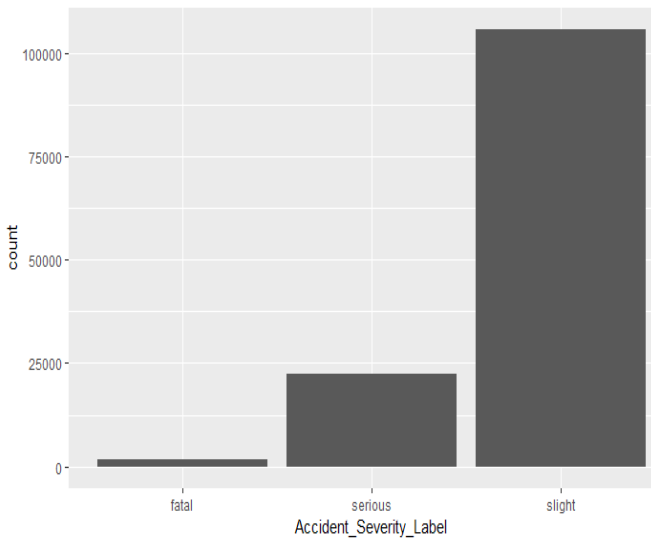
```
> acc %>% mutate(Accident_Severity=parse_double(Accident_Severity)) %>% left_join(label) %>% select(Acciden
t_Severity,Accident_Severity_Label)
Joining, by = "Accident_Severity"
# A tibble: 129,982 x 2
   Accident_Severity Accident_Severity_Label
               <dbl> <chr>
 1                 1 fatal
 2                 3 slight
 3                 3 slight
 4                 3 slight
 5                 2 serious
 6                 3 slight
 7                 3 slight
 8                 3 slight
 9                 2 serious
10                 2 serious
# ... with 129,972 more rows
```

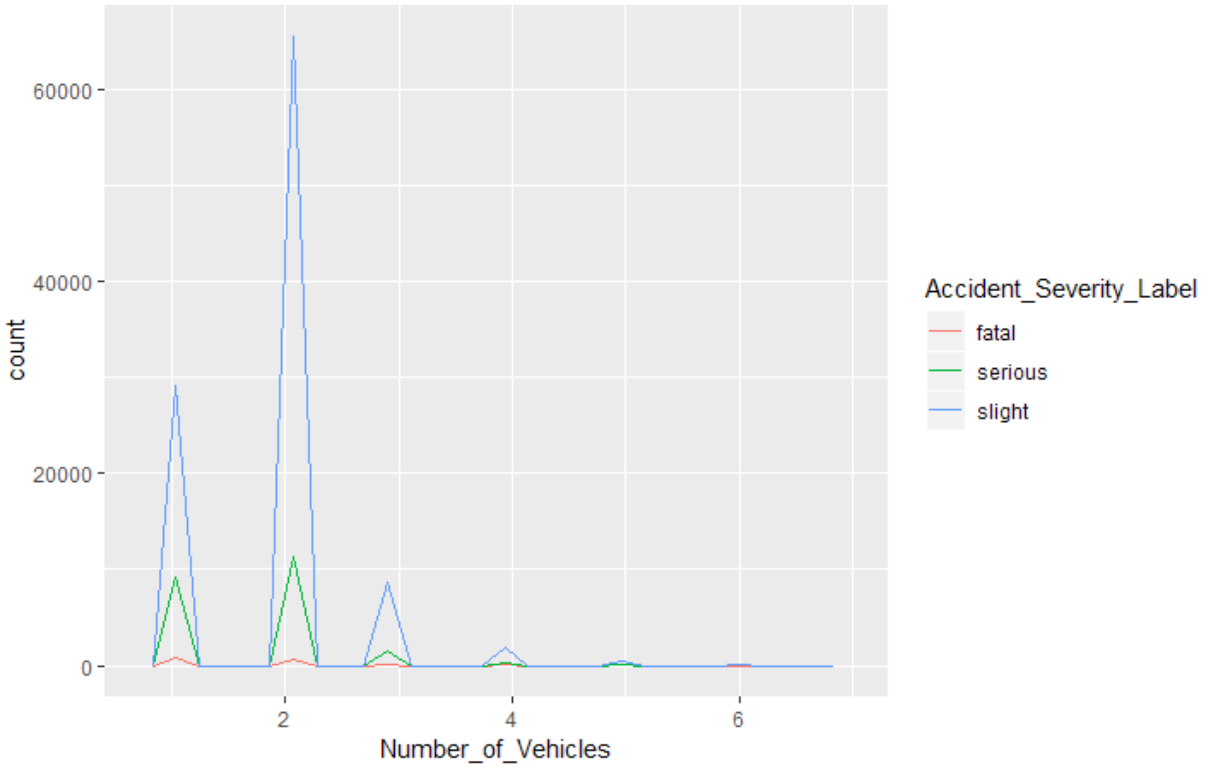
Part3. 2) 차량의 개수

사고 난 차량 개수가 많을수록 사고의 심각성이 높을 것이다

▶ 사고 심각도별 관측수의 차이로 인해 정확한 비교 불가

```
> acc1 %>% count(Accident_Severity_Label)
# A tibble: 3 x 2
  Accident_Severity_Label      n
  <chr>                    <int>
1 fatal                     1676
2 serious                  22534
3 slight                  105772
```



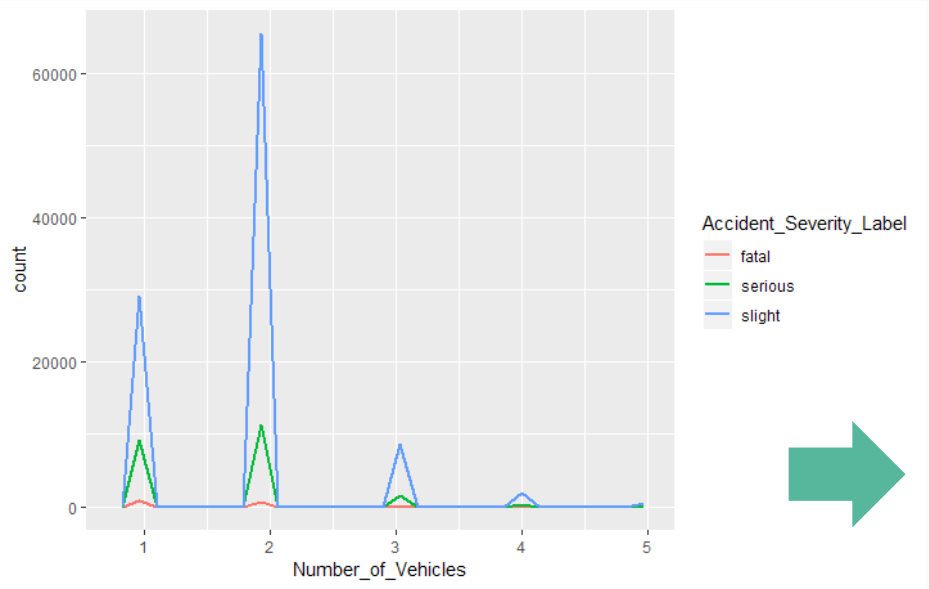


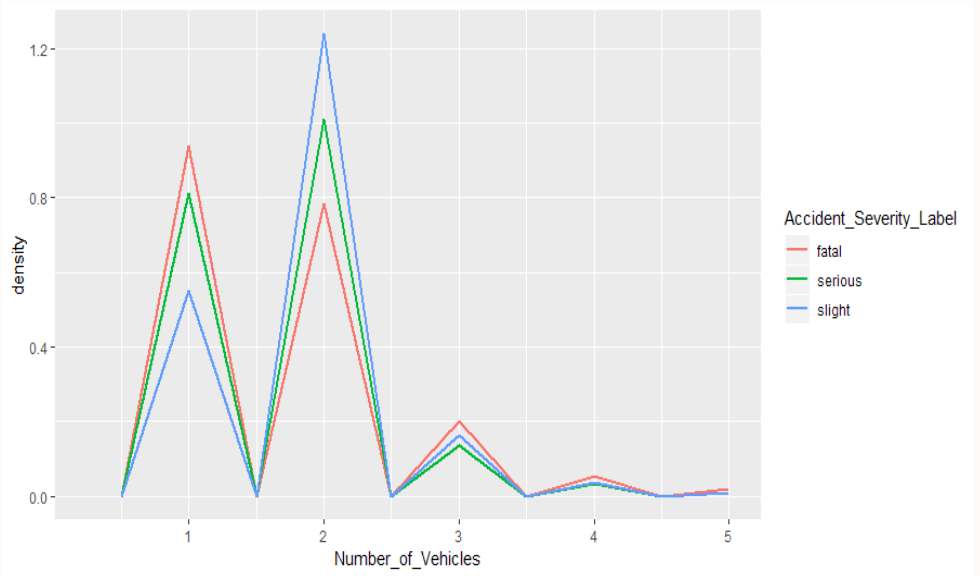
Part3. 2) 차량의 개수

사고 난 차량 개수가 많을수록 사고의 심각성이 높을 것이다

▶ 정확한 비교를 위해 **density 함수** 이용

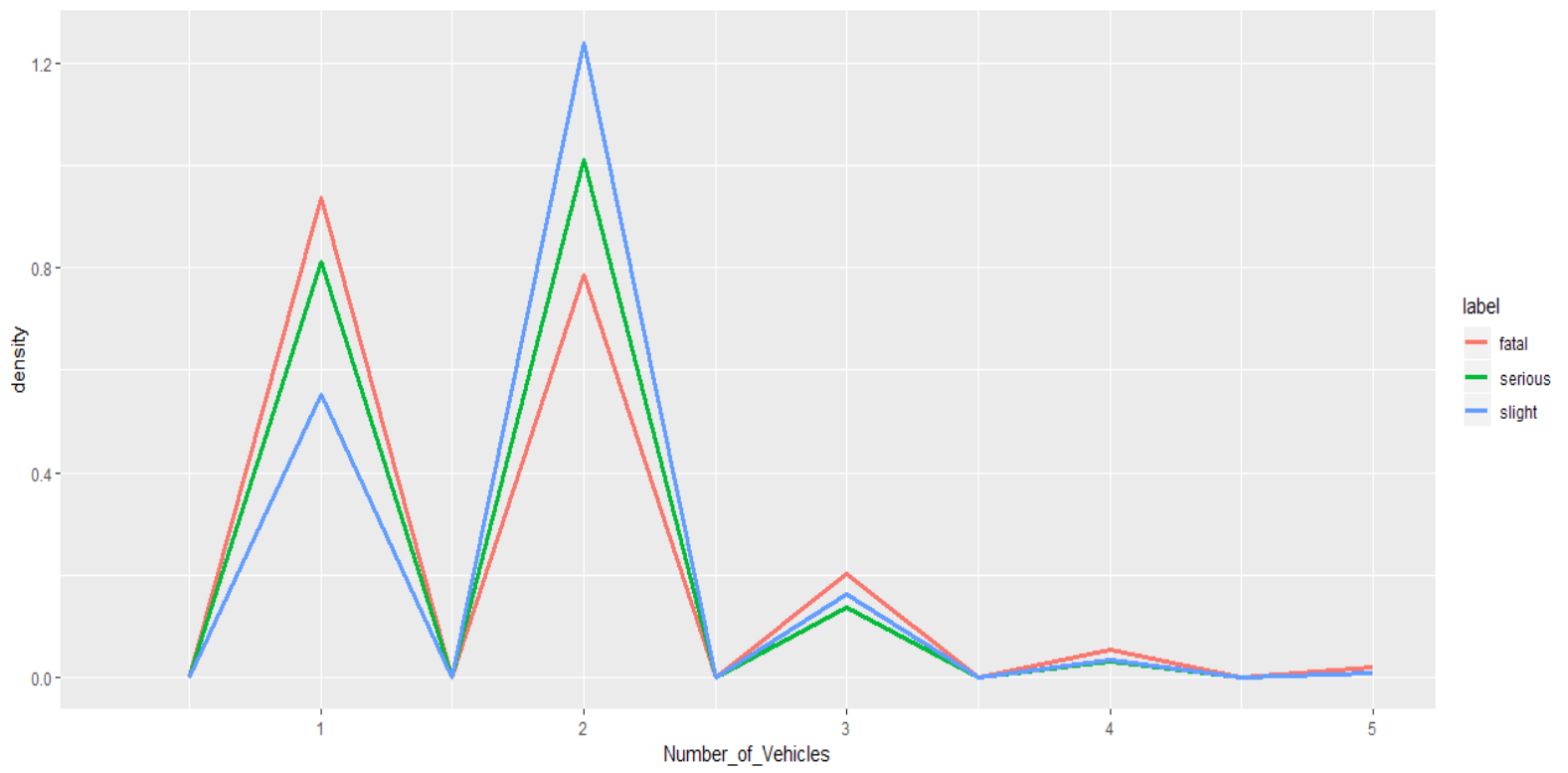
```
> acc%>% mutate(Number_of_Vehicles=parse_double(Number_of_Vehicles)) %>%
 ggplot(aes(Number_of_Vehicles,colour=Accident_Severity_Label))+geom_fre
qpoly(aes(y=..density..),binwidth=0.5,size=1)+xlim(NA,5)
```





Part3. 2) 차량의 개수

사고 난 차량 개수가 많을수록 사고의 심각성이 높을 것이다



▶ **사고 차량 1대**: Fatal 〉 Serious 〉 Slight

위험한 지역 (낭떠러지, 산간지역 등)에서 하나의 차량이 심각한 사고가 많이 발생했을것

▶ **사고 차량 2대**: Slight 〉 Serious 〉 Fatal

두 대의 차량 간 단순 접촉사고가 많이 발생했을 것

▶ **사고 차량 3대 이상**: Fatal 〉 Serious 〉 Slight

3대 이상의 차량이 추돌하여 크고 심각한 사고가 많이 발생했을 것

Part3. 3) 도시/시골 사고의 차량 종류

도시와 시골에서 사고가 발생하는 차량 종류가 다를 것이다

▶ acc_veh의 Vehicle_Type, Urban_or_Rural_Area 변수를 이용

▶ Vehicle_Type 변수

code	label
1	Pedal cycle
2	Motorcycle 50cc and under
3	Motorcycle 125cc and under
4	Motorcycle over 125cc and up to 500cc
5	Motorcycle over 500cc
8	Taxi/Private hire car
9	Car
10	Minibus (8 - 16 passenger seats)
11	Bus or coach (17 or more pass seats)
16	Ridden horse
17	Agricultural vehicle
18	Tram
19	Van / Goods 3.5 tonnes mgw or under
20	Goods over 3.5t. and under 7.5t
21	Goods 7.5 tonnes mgw and over
22	Mobility scooter
23	Electric motorcycle
90	Other vehicle
97	Motorcycle - unknown cc
98	Goods vehicle - unknown weight
-1	Data missing or out of range

▶ missing data인 code -1 과 기타 차량으로 분류된 code 90,97,98를 제외할 필요가 있음

▶ Urban_or_Rural_Area 변수
1 : Urban
2 : Rural
3 : Unallocated

▶ 코드 3을 missing data로 처리

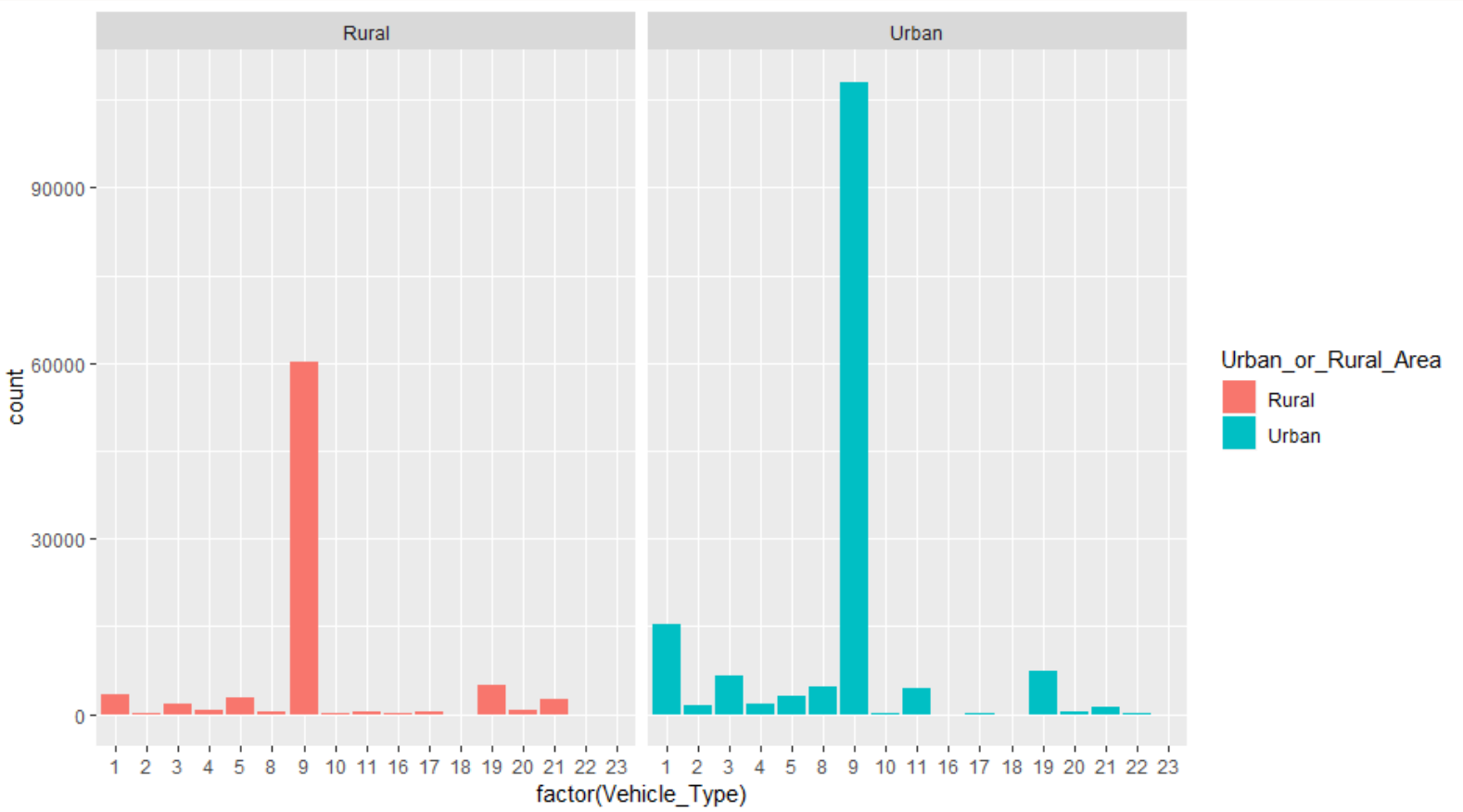
```
> table(acc_veh $ Urban_or_Rural_Area)

     1      2      3
158026  80880     20
```

▶ 위의 조건들을 반영하여 새로운 데이터셋 acc_veh1 생성

Part3. 3) 도시/시골 사고의 차량 종류

도시와 시골에서 사고가 발생하는 차량 종류가 다를 것이다

```
> acc_veh1 %>% ggplot(aes(factor(Vehicle_Type),fill=Urban_or_Rural_Area)) + geom_bar() + facet_wrap(~U
rban_or_Rural_Area)
```



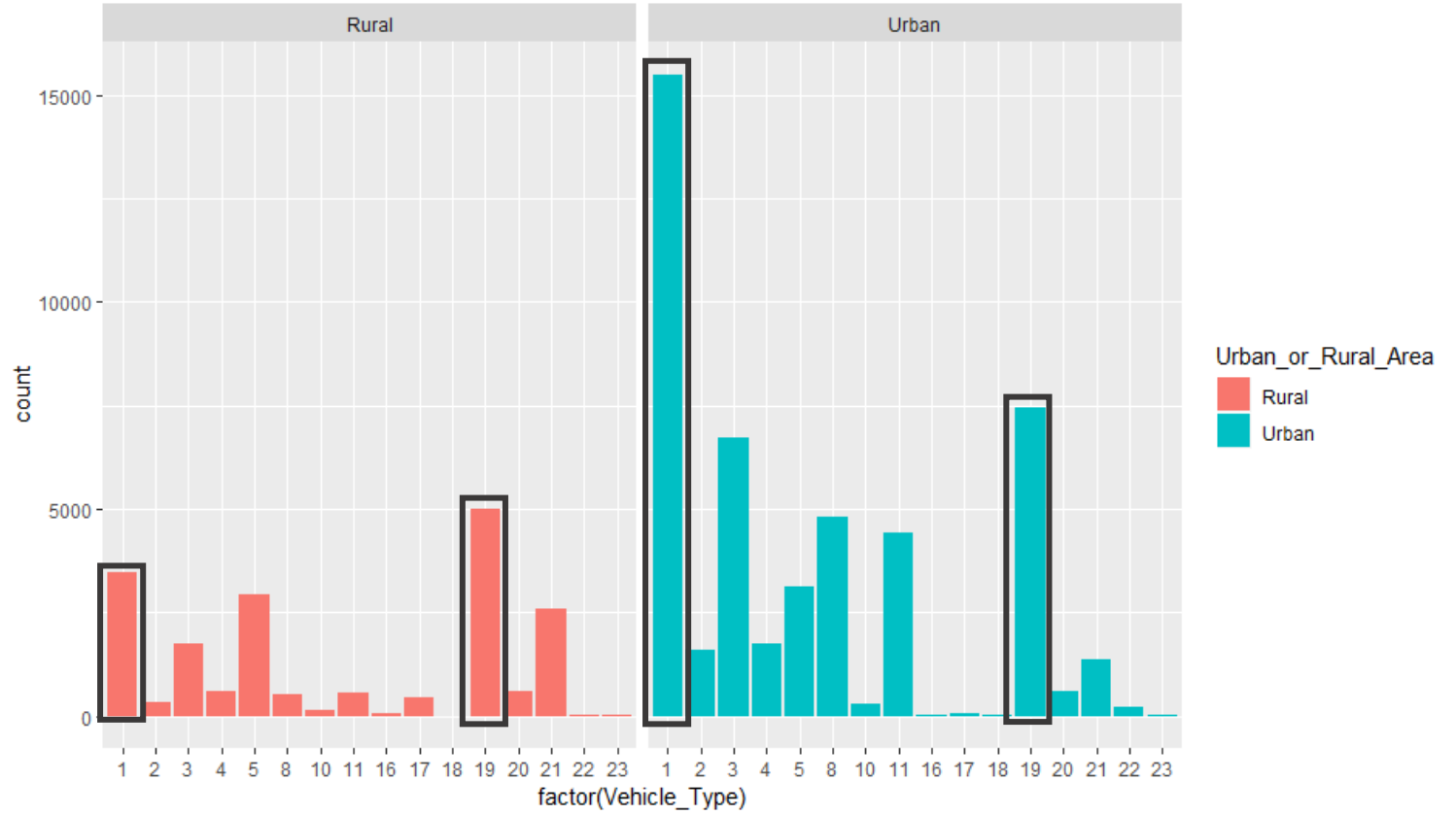
▶ **code 9 (Car)의 수가 너무 많아서 다른 교통수단의 비교가 어려움**

-〉 filter를 통해 code 9(Car)를 제외하고 다시 비교

Part3. 3) 도시/시골 사고의 차량 종류

도시와 시골에서 사고가 발생하는 차량 종류가 다를 것이다

```
> acc_veh1 %>% filter(Vehicle_Type != 9) %>% ggplot(aes(factor(Vehicle_Type),fill=Urban_or_Rural_Area)
) +  geom_bar() + facet_wrap(~Urban_or_Rural_Area)
```



▶ 시골 1위 : Code 19, 2위 : Code 1

Code 19 : Van / Goods 3.5 tonnes mgw or under
Code 1 : Pedal Cycle

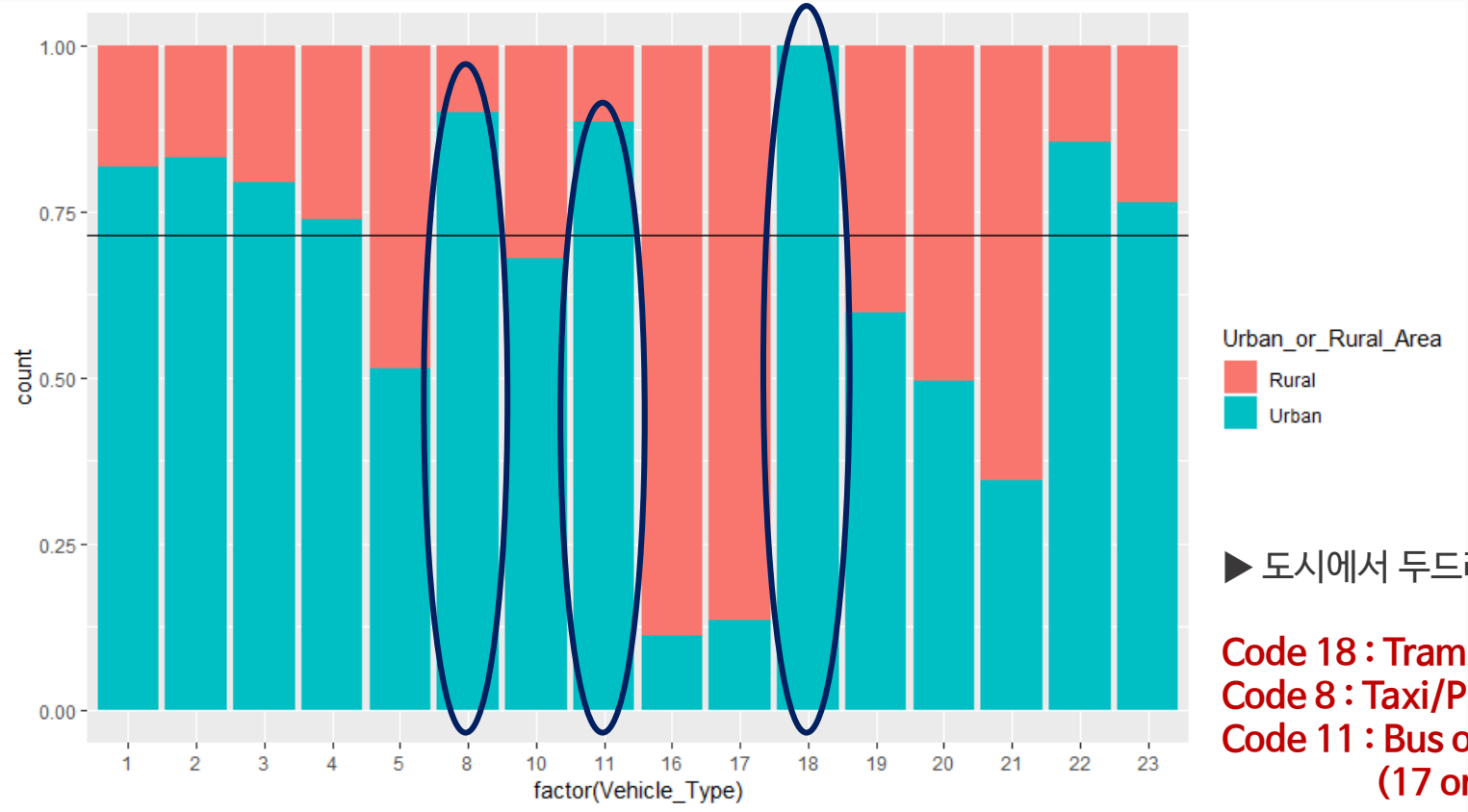
▶ 도시 1위 : Code 1, 2위 : Code 19

Code 1 : Pedal Cycle
Code 19 : Van / Goods 3.5 tonnes mgw or under

Part3. 3) 도시/시골 사고의 차량 종류

도시와 시골에서 사고가 발생하는 차량 종류가 다를 것이다

```
> acc_veh1 %>% filter(Vehicle_Type != 9) %>% mutate(rate=mean(Urban_or_Rural_Area == "Urban")) %>% ggp
lot(aes(factor(Vehicle_Type),fill=Urban_or_Rural_Area)) +  geom_bar(position="fill") + geom_hline(aes(
yintercept=rate))
```



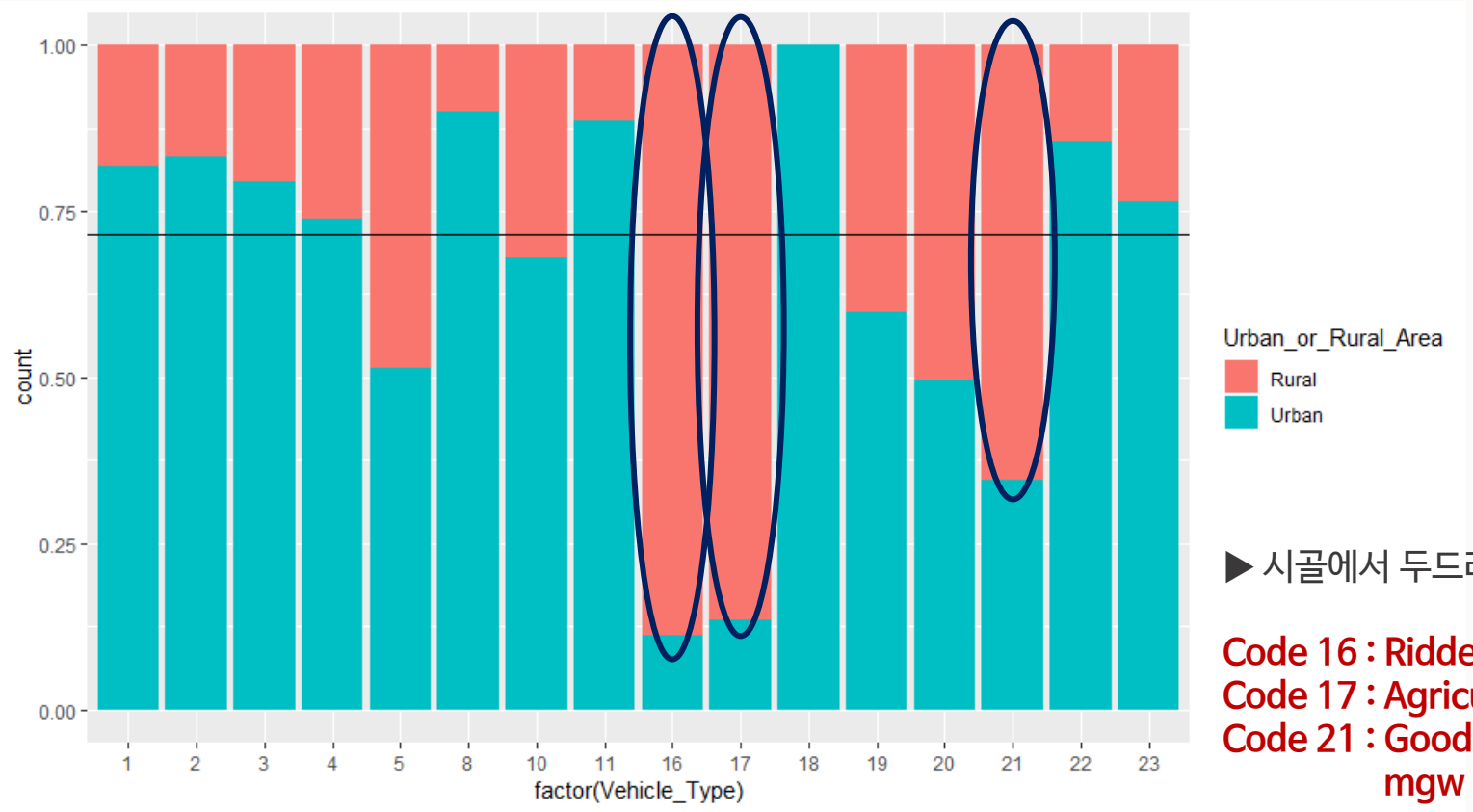
▶ 도시에서 두드러진 vehicle type

Code 18 : Tram
Code 8 : Taxi/Private hire car
Code 11 : Bus or coach
(17 or more pass seats)

Part3. 3) 도시/시골 사고의 차량 종류

도시와 시골에서 사고가 발생하는 차량 종류가 다를 것이다

```
> acc_veh1 %>% filter(Vehicle_Type != 9) %>% mutate(rate=mean(Urban_or_Rural_Area == "Urban")) %>% ggp
lot(aes(factor(Vehicle_Type),fill=Urban_or_Rural_Area)) +  geom_bar(position="fill") + geom_hline(aes(
yintercept=rate))
```



▶ 시골에서 두드러진 vehicle type

Code 16 : Ridden horse
Code 17 : Agricultural vehicle
Code 21 : Goods 7.5 tonnes mgw and over

정리

2017UKRoadSafetyData 

Part1

1) 월별
2) 시간별

Part2

1) 운전자 성별 연령별
2) 왼손잡이 운전자

Part3

1) 차량의 연식
2) 사고 난 당시의 차량 개수
3) 도시/시골 사고의 차량 종류

감사합니다!