데이터 과학

3주차

tashu_ggplot2

201202156 오희빈

1. 정류장 사용빈도 Top10

1-1 bar형태

```
1  library("ggplot2")
2  library("scales")
3  library("ggmap")
4  library("stringr")
5  library("hexbin")
6
7  setwd('C:/Users/heebin/Documents/R/tashu')
8  tashu <- read.csv('tashu.csv')
9  tashu_station <- read.csv('station.csv')
```

=> setwd()함수를 이용해서 csv파일을 가져오게끔 경로를 지정해 준다.
tashu.csv와 station.csv를 read.csv()함수를 이용해서 파일안의 칼럼값들을 가져온다.

```
14 station <- data.frame(table(tashu$RENT_STATION))
15 return_station <- data.frame(table(tashu$RENT_STATION))
```

=> station변수에 변수 tashu에 읽어온 데이터인 RENT_STATION값을 data.frame함수를 이용해서 table형태로 저장하고 return_station변수에 위와 마찬가지로 RETURN_STATION값을 table형태로 저장한다.

```
23 x <- station[[2]][1:144] + return_station[[2]][1:144]
24 station[[2]][1:144] <- x[1:144]
25 station2 <- station[order(-station$Freq),]
26 result <- station2[1:10,]
27 result_station <- result[order(result$Var1),]
28 names(result_station)[1] <- "station"
```

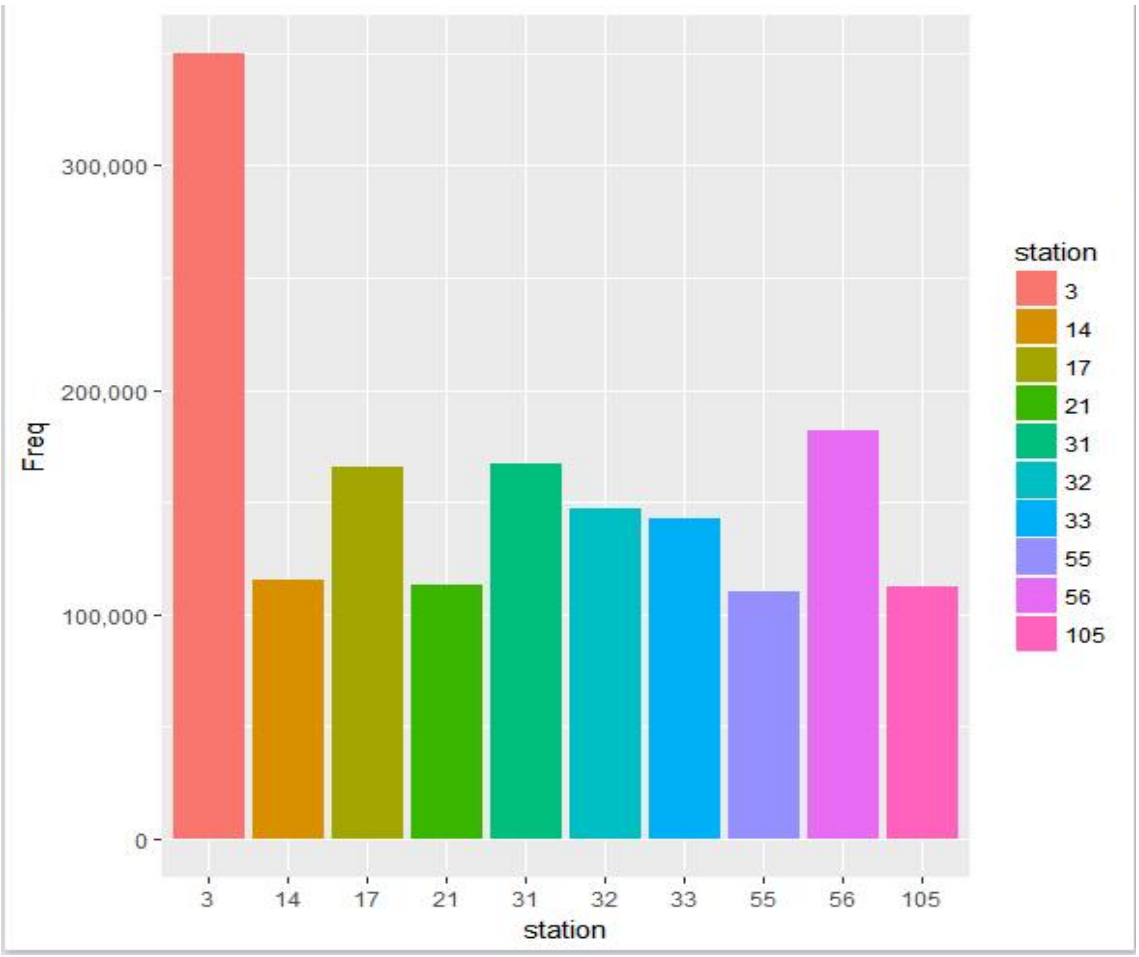
=> x변수에 RENT_STATION과 RETURN_STATION의 값을 더해서 정류장의 수만큼 저장해준다. 테이블형태에서 값을 더하기 힘들었기에 x변수에 따로 두 값을 더해 저장시킨후 다시 그값을 테이블형태인 station변수에 값을 저장시켰다. 그리고 테이블의 Freq 칼럼의 값을 큰순서로 정열시켰다. 다음으로 그중 top10의 값을

result변수에 저장시키고 그래프를 station순서로 보여주게 하기위해 다시 station중심으로 내림차순으로 정렬시켰다.

```
30 bar <- ggplot(result_station, aes(x=station, y=Freq, fill=station)) + geom_bar(stat="identity") + scale_y_continuous(labels = comma)
31 bar
```

=> 막대 그래프 형태로 출력시키기위해 x축은 station, y축은 Freq로 해주고 각 station마다 색깔을 다르게 하기위해
fill = station 으로 값을 주었다. 그리고 y축의 값이 정수 값으로 제대로 나오지 않아서 scale_y_continuous(labels = comma)를 이용해서 값을 제대로 나오게끔 만들었다.

결과



1-2 map으로 정류장 위치 찍기

=> library와 read.csv는 위와 똑같이 사용했다.

```
daejon_gc <- geocode('Daejon')
daejon_cent <- as.numeric(daejon_gc)
```

=> 대전의 map을 가져오기위해 geocode('Daejon')을 사용해서 대전의 지도를 가져오고 daejon_cent함수에 대전의 중심 좌표를 저장시켰다.

```
station_location <- data.frame(table(tashu_station))
result_station_location <- str_split_fixed(tashu_station$좌표, ",", 2)
Info <- cbind(tashu_station,result_station_location)
names(Info)[9] = "lat"
names(Info)[10] = "lon"
```

=> station.csv의 각 칼럼의 데이터들을 station_location변수에 table 형태로 저장시킨후 좌표를 각각 위도와 경도로 나누기위해 str_split_fixed(tashu_station$좌표, ",", 2)를 이용해서 좌표를 ,를 경계로 2개로 나누었고 그 나눈 변수와 기존의 station_location를 cbind 함수를 이용해 합쳐서 Info변수에 저장시켰다. 그리고 위도와 경도의 칼럼이름을 lat, lon으로 변형시켜 저장하였다.

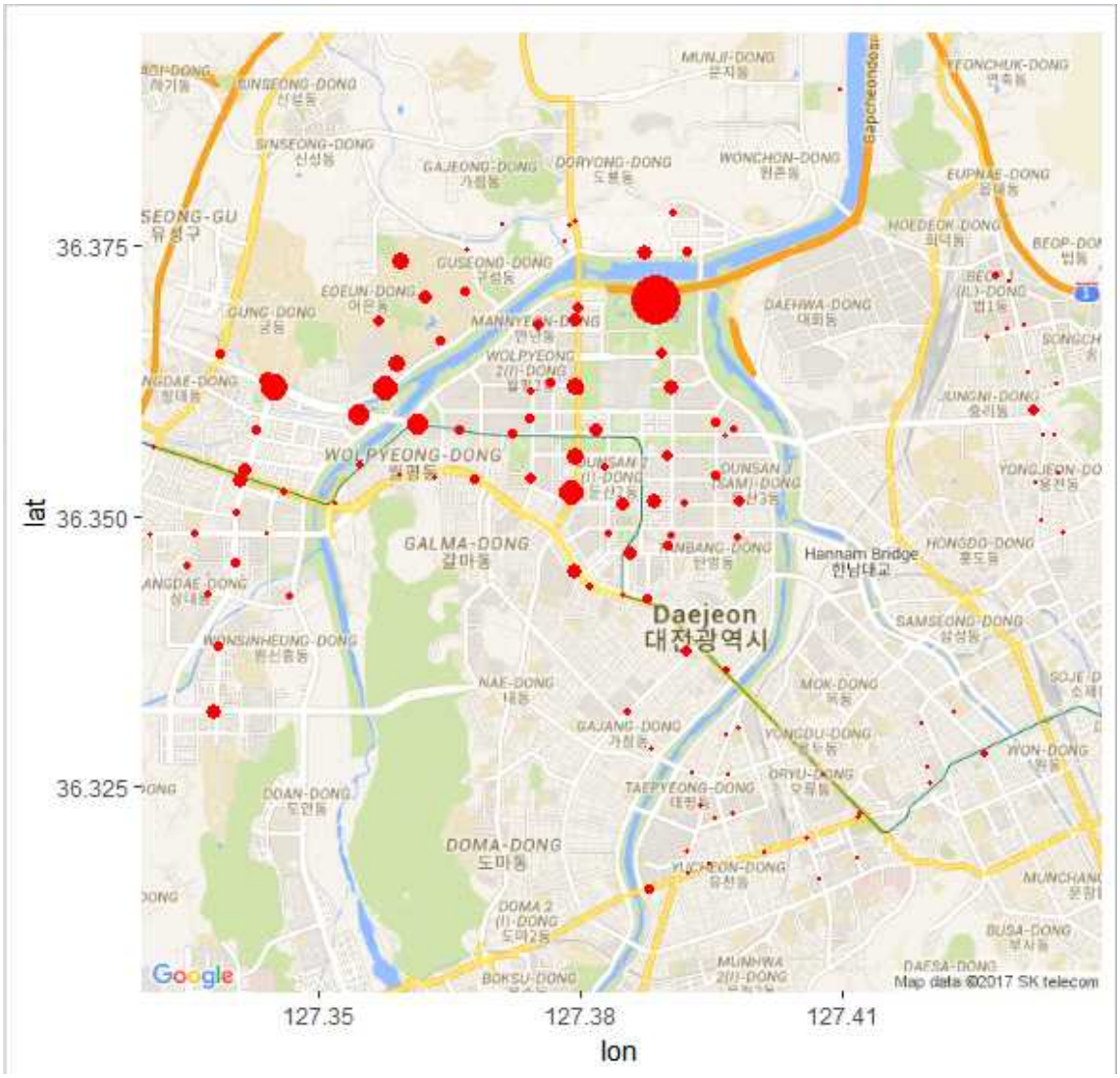
```
station_map <- get_googlemap(center = daejon_cent, scale = 1, zoom=13, maptype="roadmap")
map <- ggmap(station_map) + geom_point(data=Info, aes(x=as.numeric(as.character(lon)), y=as.numeric(as.character(lat))),size=x^0.0000248,colour='red')
map

```

=> station_map변수에 googlemap을 이용해서 대전의 지도를 zoom=13 그리고 roadmap형태로 저장시켰다. 다음으로 지도에 각 station의 위도, 경도를 계산해 점을 찍기위해 geom_point함수를 이용했는데 각 x축과 y축에 as.numeric(as.character(lon))과 as.numeric(as.character(lat))을 사용한 이유는 factor형태로 저장되어 있기에 num으로 바꾸어야 에러가 없이 지도에 위도 경도를 찾을수 있기 때문에 사용했다. 그리고 size를 1-1번에서 사용했던 각 station의

총 freq값을 사용해서 0.0000248 * x로 점의 크기를 조절하고 colour = “red”로 색을 빨간색으로 만들었다.

결과



2. 가장 인기 있는 경로 Top20

```
1  library("ggplot2")
2  library("scales")
3
4  setwd('C:/Users/heebin/Documents/R/tashu')
5  tashu <- read.csv('tashu.csv')
6
7  station_route <- data.frame(table(tashu$RENT_STATION,tashu$RETURN_STATION))
8
9  names(station_route)[1] <- "rent"
10 names(station_route)[2] <- "return"
11 names(station_route)[3] <- "routeFreq"
```

=> setwd함수를 이용해 파일의 경로를 저장하고 read.csv함수를 이용해 tashu.csv파일을 읽어왔다. 그리고 data.frame(table(tashu$RENT_STATION,tashu$RETURN_STATION))을 이용해서 rent_station과 return_station의 모든 경로를 table형태로 만들고 빈도수를 저장하였다. 그리고 각각의 칼럼의 이름을 변경하였다.

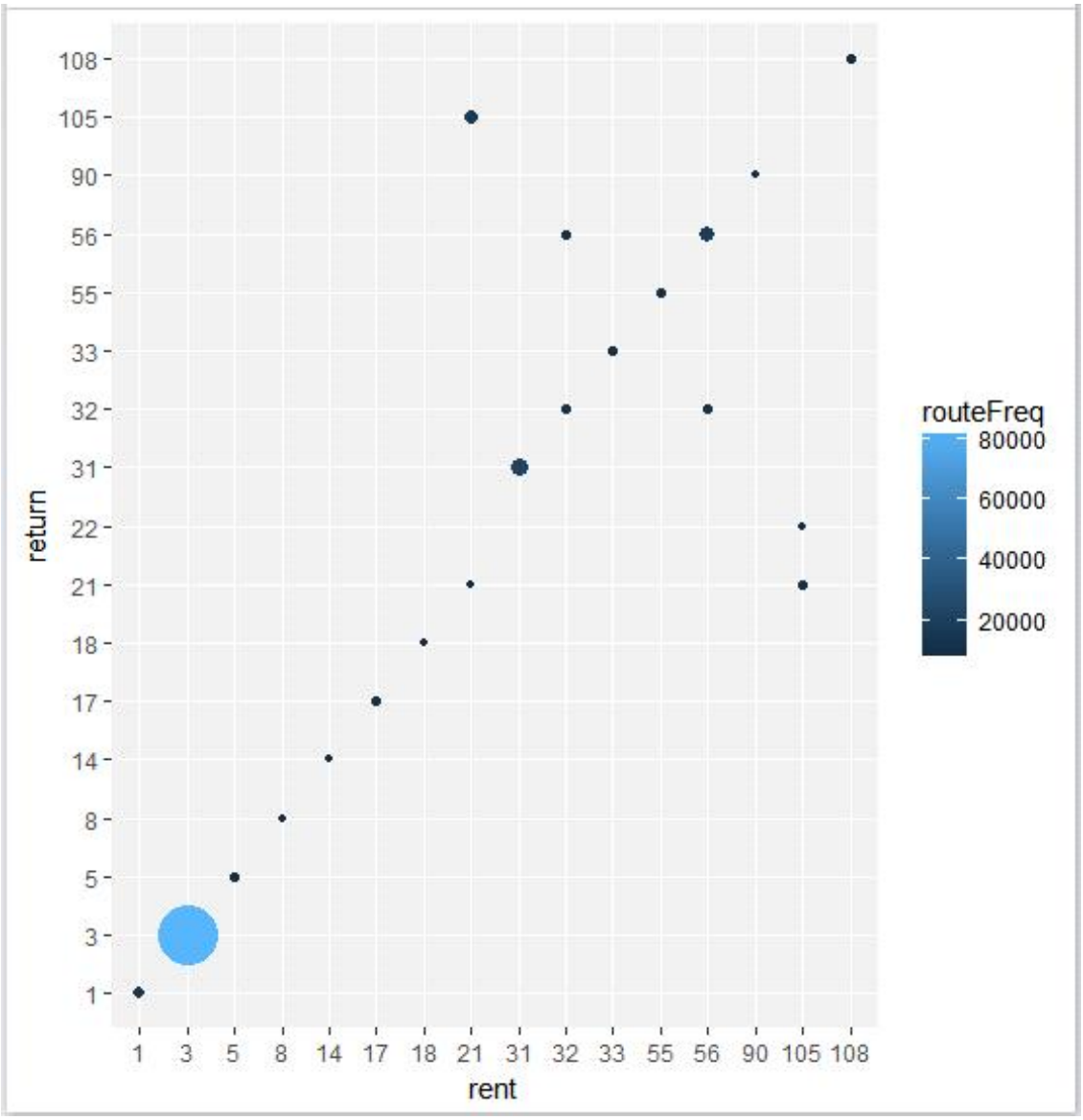
```
12 station <- station_route[order(-station_route$routeFreq),]
13 result <- station[1:20,]
14 result_route <- result[order(result$rent),]
15 freq <- result_route[[3]][1:20]
```

=> 경로의 빈도수를 중심으로 table을 큰 순서로 정렬 시킨 후 top20의 경로를 result변수에 저장 시킨 후 result_route변수에 station의 번호 수를 작은 순서로 다시 정렬 시켰고 freq에 빈도수만 따로 저장해 그래프의 점의 크기를 조절했다.

```
17 point <- ggplot(data = result_route, aes(rent, return, colour = routeFreq))
18 point + geom_point(shape=19, size=freq*0.0001259)
```

=> ggplot함수를 이용해서 x축은 rent, y축은 return으로 그래프를 만들면서 빈도수에 따라 색을 다르게 하기 위해 colour = routeFreq로 지정해 주었다. 그리고 geom_point함수를 이용해서 shape=19로 점을 원 형태로 size를 빈도수 * 0.0001259로 각각의 빈도수에 따라 점의 크기를 조절했다.

결과


return
108
105
90
56
55
33
32
31
22
21
18
17
14
8
5
3
1
rent
1 3 5 8 14 17 18 21 31 32 33 55 56 90 105 108
routeFreq
80000
60000
40000
20000