

과제명	빅데이터 처리 기말고사 과제
-----	-----------------



- 과 목 명 빅데이터 처리
- 담당교수 김 마 루
- 학부(과) 컴퓨터소프트웨어공학과
- 학 년 4
- 분반번호 01
- 학 번 20173147
- 이 름 이 명 진
- 연 락 처 010-2999-7748
- 제 출 일 2020 년 06 월 17 일



원광대학교
WONKWANG UNIVERSITY

#1 이 데이터를 선택한 이유

제가 선택한 데이터는 ‘CSV 로드킬 데이터 정보(2019년)’ 파일로 전국 시도별 로드킬 데이터 정보(본부명, 지사명, 노선명, 구간, 방향, 5km, 발생건수, 사고율, 반기, 위도, 경도)등을 담고 있습니다. 로드킬은 사람에게도 매우 위험한 사고입니다. 차량 파손을 넘어 인명피해가 있을 수도 있습니다. 로드킬은 정확히 동물이 도로에서 교통사고를 당해 목숨을 잃는 것을 뜻합니다. 개체 수나 습성 때문에 특정 동물이 빈번하게 로드킬 당하지만, 로드킬이라는 개념의 객체에는 개, 고양이, 조류 등 모든 동물이 포함됩니다. 대다수 운전자가 로드킬 사고 후 동물을 방치하고 자리를 뜬다고 합니다. 숨이 붙어 있는 동물을 그대로 방치한다면 더 큰 고통 속에 천천히 죽어갈 수밖에 없습니다. 당연하지만 이러한 사고를 막기 위해선 애초에 사고가 일어나지 않도록 하는 것입니다. 그러나 주로 시야가 어두운 밤에 급작스럽게 일어나는 로드킬 특성 상 이를 막기란 쉬운 일이 아닙니다. 그나마 존재하는 방안 중 하나는 로드킬이 자주 일어나는 도로에 ‘야생동물 출몰 주의’ 표지판을 운전자의 눈에 잘 들어오는 곳에 설치하는 것입니다. 그러나 매년 새로운 도로가 생겨나고 있고 주변 환경의 변화에 따라 야생 동물이 자주 출몰하는 지역이 바뀌기 때문에 표지판 최신화가 필요한 상황입니다. 제가 선택한 데이터를 통하여 한국도로공사에서 이러한 표지판 최신화를 수행하는데 도움이 되고자 ‘CSV 로드킬 데이터 정보(2019년)’를 이용한 프로젝트를 기획하였습니다.



<데이터 다운로드 경로: <https://www.data.go.kr/data/15045544/fileData.do>>

[hmgjournal의 기사 및 사진 인용]

<https://news.hmgjournal.com/TALK/%EB%A1%9C%EB%93%9C%ED%82%AC-%EC%98%88%EB%B0%A9%EB%B2%95%EB%B6%80%ED%84%B0-%EB%8C%80%EC%B2%98%EB%B2%95%EA%B9%8C%EC%A7%80>

#2 이 데이터의 필요성과 활용 방안

위에서 언급하였듯이 ‘CSV 로드킬 데이터 정보(2019년)’ 데이터를 활용하여 프로젝트를 수행하고 이를 한국도로공사에 제공하여 ‘야생동물 출몰 주의’ 표지판을 업데이트를 하는데 기여할 수 있습니다. 이를 통해 야생동물이 자주 출몰하는 지역에서 운전자들에게 경각심을 심어줄 수 있습니다. 또한 불가피하게 일어나고 사고를 낸 운전자뿐 만아니라 차에 치인 동물들 모두 후유증이 큰 로드킬 사고 발생률을 낮출 수 있습니다.

#3 이 데이터를 R의 어떤 방법을 통해 처리하고 분석할 것인가?

파이차트를 이용하여 전국 시도별(수도권, 강원, 충북, 대전충남, 전북, 광주전남, 대구경북, 부산경남)로 나누어 로드킬 사고 분포를 한눈에 알아보기 쉽게 파악하고 구글 지도 API를 활용해 전국에서 발생한 로드킬 사고를 지도에 원으로 표시하여 분포를 자세히 확인할 수 있게 하였습니다.

#4 사용한 패키지(라이브러리)들의 명세

library(RColorBrewer) - n Provides color schemes for maps (and other graphics)designed by Cynthia Brewer as described at <http://colorbrewer2.org>

<출처:

<https://cran.r-project.org/web/packages/RColorBrewer/RColorBrewer.pdf> >

library(ggmap) - Google map으로 부터 street map을 다운로드 받고, ggplot의 background로 쓸 수 있다.

library(ggplot2) - R의 매우 유명한 시각화 패키지이다. grammar of graphics를 활용하여 layered, customizable plot을 만들 수 있다는 장점이 있다

<출처: <https://3months.tistory.com/291> [Deep Play]>

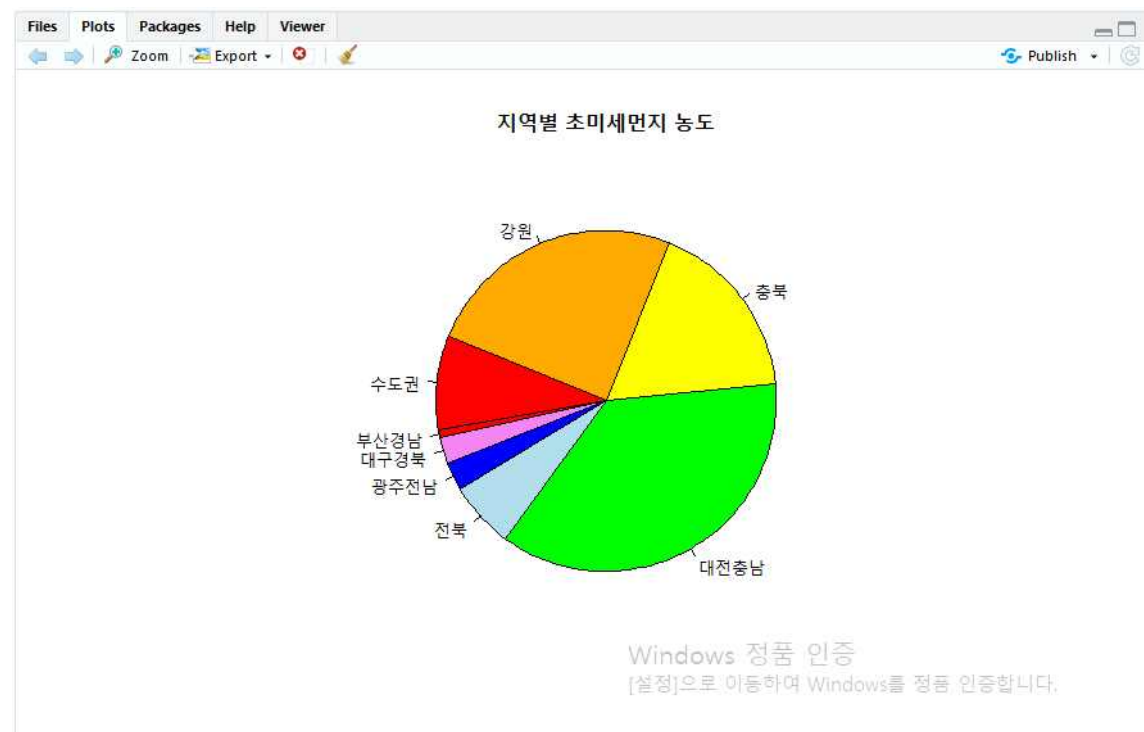
library(openxlsx) - Simplifies the creation of Excel .xlsx files by providing a high level interface to writing, styling and editing worksheets. Through the use of 'Rcpp', read/write times are comparable to the 'xlsx' and 'XLConnect' packages with the added benefit of removing the dependency on Java.

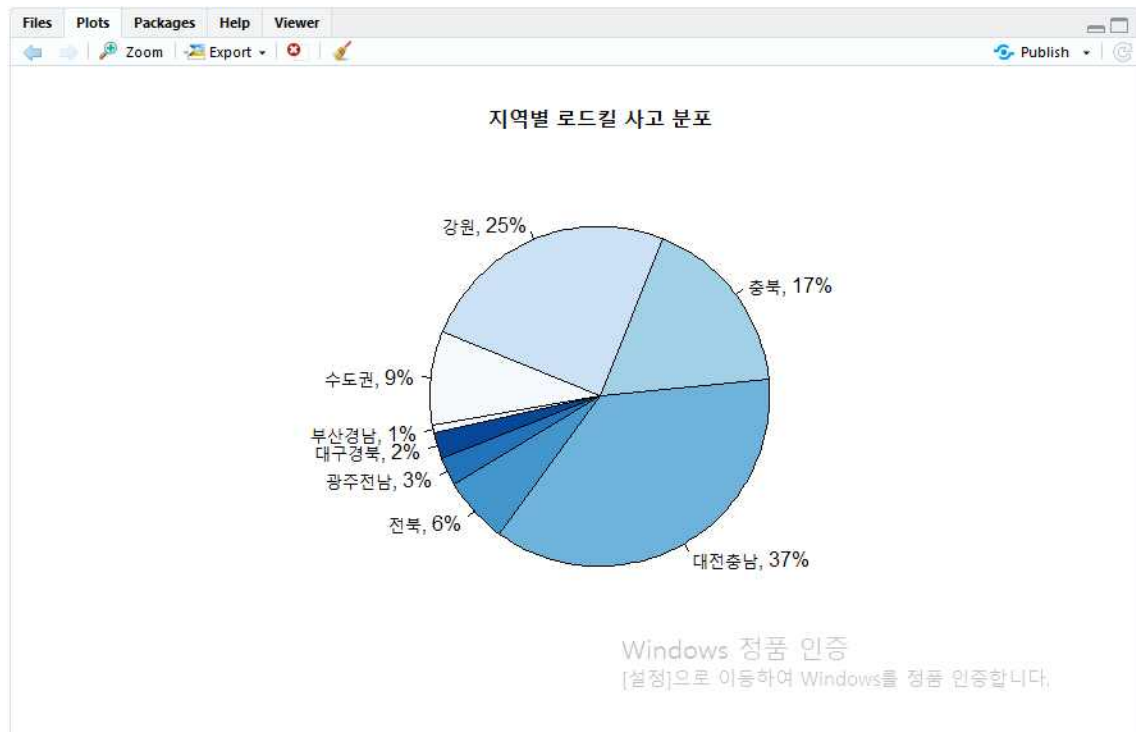
<출처: <https://cran.r-project.org/web/packages/openxlsx/openxlsx.pdf>>

#5 최종 완성 화면 (스크린샷) - 1

```
1 #20173147 이명진
2 # 지역별 로드킬 분포 파이차트
3 city <- c("수도권", "강원", "충북", "대전충남", "전북", "광주전남", "대구경북", "부산경남")
4 pm25 <- c(51, 141, 98, 208, 36, 15, 14, 4) #지역별 로드킬 사고건수
5 colors <- c("red", "orange", "yellow", "green", "lightblue", "blue", "violet")
6
7 pie(pm25,
8     labels=city,
9     col=colors,
10    main="지역별 초미세먼지 농도",
11    init.angle=190, clockwise=T)
12 #####
13 install.packages("RColorBrewer")
14 library(RColorBrewer)
15 blues <- brewer.pal(7, 'blues')
16 pct <- round(pm25/sum(pm25)*100, 0)
17 city_label <- paste(city, " ", pct, "%", sep="")
18 pie(pm25,
19     labels=city_label,
20     col=blues,
21     main="지역별 로드킬 사고 분포",
22     init.angle=190, clockwise=T)
23
```

```
> city <- c("수도권", "강원", "충북", "대전충남", "전북", "광주전남", "대구경북", "부산경남")
> pm25 <- c(51, 141, 98, 208, 36, 15, 14, 4) #지역별 로드킬 사고건수
> colors <- c("red", "orange", "yellow", "green", "lightblue", "blue", "violet")
> pie(pm25,
+     labels=city,
+     col=colors,
+     main="지역별 초미세먼지 농도",
+     init.angle=190, clockwise=T)
> library(RColorBrewer)
> blues <- brewer.pal(7, 'blues')
> pct <- round(pm25/sum(pm25)*100, 0)
> city_label <- paste(city, " ", pct, "%", sep="")
> pie(pm25,
+     labels=city_label,
+     col=blues,
+     main="지역별 로드킬 사고 분포",
+     init.angle=190, clockwise=T)
> |
```





#6 최종 완성 화면 (스크린샷) - 2

```

1 #20173147 이명진
2
3 install.packages("ggmap")
4 install.packages("ggplot2")
5 install.packages("openxlsx")
6 library(ggmap)
7 library(ggplot2)
8 library(openxlsx)
9
10 register_google(key = "AizaSyA3ouHF9iggF9ptj5LLRA_16yhyksrC1mY")
11 df <- read.xlsx(file.choose(), sheet=1, startRow=2)
12 head(df)
13
14 df2 <- data.frame(lon=df[,11], lat=df[,10], freq=df[,7]) # 위도, 경도, 사고 빈도수 데이터가 들어있는 10열, 11열, 7열
15 #만으로 df2 데이터 프레임 생성
16
17 str(df2)
18 head(df2)
19
20 df2[,1] <- as.numeric(as.character(df2[,1])) #Factor로 되어 있는 데이터 유형을 계산이 가능한 숫자형으로 변환
21 df2[,2] <- as.numeric(as.character(df2[,2]))
22 df2[,3] <- as.numeric(as.character(df2[,3]))
23 str(df2)
24
25 #지도 중심 좌표 설정
26 cen <- c((max(df2$lon)+min(df2$lon))/2, #위도의 최댓값+최솟값/2 = 위도 중심 값, 경도의 최댓값+최솟값/2 = 경도 중심 값
27         (max(df2$lat)+min(df2$lat))/2)
28
29 # 지도 출력
30 map <- get_googlemap(center=cen, zoom=8)
31 gmap <- ggmap(map)
32 gmap + geom_point(data=df2,
33                  aes(x=lon, y=lat),
34                  color="blue",
35                  size=df2$freq,
36                  alpha=0.6)

```

```

> library(ggmap)
필요한 패키지를 로딩중입니다: ggplot2
google's Terms of Service: https://cloud.google.com/maps-platform/terms/.
Please cite ggmap if you use it! See citation("ggmap") for details.
> library(ggplot2)
> library(openxlsx)
> register_google(key = "AizaSyA3ouHF9iggF9ptj5LLRA_16yhyksrC1mY")
> df <- read.xlsx(file.choose(), sheet=1, startRow=2)
> head(df)
  수도권 경기광주 제2중부선 330~335 하남 330 4 0.8 상반기 37.283000000000001 127.386
1 수도권 경기광주 중부선 325~330 통영 325 7 1.4 상반기 37.249 127.421
2 수도권 경기광주 중부선 330~335 통영 330 10 2.0 상반기 37.283 127.385
3 수도권 경기광주 중부선 335~340 통영 335 7 1.4 상반기 37.315 127.347
4 수도권 경기광주 중부선 340~345 통영 340 4 0.8 상반기 37.354 127.321
5 수도권 경기광주 중부선 345~350 통영 345 6 1.2 상반기 37.398 127.311
6 수도권 경기광주 중부선 350~355 통영 350 3 0.6 상반기 37.439 127.289
> df2 <- data.frame(lon=df[,11], lat=df[,10], freq=df[,7]) # 위도, 경도, 사고 빈도수 데이터가 들어있는 10열, 11열, 7열
> #만으로 df2 데이터 프레임 생성
> str(df2)
'data.frame': 121 obs. of 3 variables:
 $ lon : num 127 127 127 127 127 ...
 $ lat : num 37.2 37.3 37.3 37.4 37.4 ...
 $ freq: num 7 10 7 4 6 3 3 3 5 3 3 ...

```



```

> head(df2)
  lon lat freq
1 127.421 37.249 7
2 127.385 37.283 10
3 127.347 37.315 7
4 127.321 37.354 4
5 127.311 37.398 6
6 127.289 37.439 3
> df2[,1] <- as.numeric(as.character(df2[,1])) #Factor로 되어 있는 데이터 유형을 계산이 가능한 숫자형으로 변환
> df2[,2] <- as.numeric(as.character(df2[,2]))
> df2[,3] <- as.numeric(as.character(df2[,3]))
> str(df2)
'data.frame': 121 obs. of 3 variables:
 $ lon : num 127 127 127 127 127 ...
 $ lat : num 37.2 37.3 37.3 37.4 37.4 ...
 $ freq: num 7 10 7 4 6 3 3 5 3 3 ...
> #지도 중심 좌표 설정
> cen <- c((max(df2$lon)+min(df2$lon))/2, #위도의 최댓값+최솟값/2 = 위도 중심 값, 경도의 최댓값+최솟값/2 = 경도 중심 값
+ (max(df2$lat)+min(df2$lat))/2)
> cen
[1] 127.8425 36.5430
> # 지도 출력
> map <- get_googlemap(center=cen, zoom=8)
source : https://maps.googleapis.com/maps/api/staticmap?center=36.543,127.8425&zoom=8&size=640x640&scale=2&maptype=terrain
&key=xxx
> gmap <- ggmap(map)
> gmap + geom_point(data=df2,
+ aes(x=lon, y=lat),
+ color="blue",
+ size=df2$freq,
+ alpha=0.6)

```

