~\OneDrive\3학년(2025)\빅데이터\study_0528.R

```r
setwd("C:/r_workdate")
Sys.setlocale("LC_ALL", "Korean")

library(lubridate)
library(dplyr)
library(ggplot2)
library(googleVis)

library(plyr)
# =======================================================
# ggplot2 패키지 : 다양한 형태의 그래프를 쉽게 표현
# =======================================================
# =======================================================
# 🟦 ggplot(): 그래프 틀
# =======================================================
# - plot() 함수의 확장 버전
# - 다양한 레이어와 옵션을 조합하여 고급 그래프 구현 가능
# - 형식: ggplot(데이터프레임, aes(x = x축 데이터, y = y축 데이터)) + geom_*() 함수

# =======================================================
# 🟦 geom 함수 (geom_*) : 실제 그래프 형태
# =======================================================
# - 그래프의 종류와 모양을 지정하는 함수
# - 항상 ggplot() 뒤에 + 기호로 연결

# =========================
# 1. geom 함수의 stat 옵션
# =========================
#   - stat: 주어진 데이터에서 geom에 필요한 데이터를 생성한다.
#   - stat_bin: 아래와 같은 데이터를 갖는 dataframe을 출력
#     1. count    : 항목별 빈도수 (기본 막대그래프)
#     2. density  : 항목별 밀도 (히스토그램 분석용)
#     3. ncount   : count를 0~1 범위로 정규화 (상대 비교용)
#     4. ndensity : density를 0~1 범위로 정규화 (밀도 비교용)

# =========================
# 2. geom 함수의 종류
# =========================

# -------------------------------------------------------
# 2-1. 산점도 (geom_point)
# -------------------------------------------------------
kor = read.table("학생별국어성적_new.txt", header=TRUE, sep=',')
ggplot(kor, aes(x=이름, y=점수)) + geom_point()

# -------------------------------------------------------
# 2-2. 막대그래프 (geom_bar): barplot와 비슷
# -------------------------------------------------------
ggplot(kor, aes(x=이름, y=점수)) + geom_bar(stat='identity')
             # stat = "identity": y 값을 그대로 막대 높이로 사용

```

```r
# ● 막대 테두리/채우기 색상 지정
gg1 = ggplot(kor, aes(x=이름, y=점수)) +
      geom_bar(stat='identity', color='red', fill='green')

# ● x축 이름 색상, 크기 변경
gg1 + theme(axis.text.x = element_text(angle=45, hjust=1, vjust=1, colour='blue', size=8))


# ------------------------------------------------------
# 2-3. 누적 막대그래프 (geom_bar + geom_text)
# ------------------------------------------------------
kem = read.csv("학생별과목별성적_국영수_new.csv")
skem = arrange(kem, 이름, 과목) #오름차순 정렬


# 하나의 막대그래프에 국/영/수 성적을 표현
skem2 = ddply(skem, '이름', transform, 누적합계 = cumsum(점수)) # 누적합계 열 생성: 사람 기준 점수 누적합

# 각 점수 영역 중간에 점수를 표현
skem3 = ddply(skem2, '이름', transform,
              누적합계 = cumsum(점수),
              label = cumsum(점수) - 0.5 * 점수) # 점수 영역의 50% 위치 지정

# 누적 막대그래프 + 텍스트 + 범례 반전
gg2 = ggplot(skem3, aes(x=이름, y=점수, fill=과목)) +
      geom_bar(stat='identity', position=position_stack(reverse=T)) +  # 쌓는 순서 역전
      geom_text(aes(y=label, label=paste(점수,'점')), color='black', size=4) +  # 텍스트 표시
      guides(fill=guide_legend(reverse=T))  # 범례 순서도 역전
      # position=position_stack(reverse=T)가 없으면 데이터가 반대로 나온다

# x축 이름 회전 및 정렬
gg2 + theme(axis.text.x = element_text(
      angle = 45,         # 글자를 45도 기울임
      hjust = 1,          # 수평 정렬 (horizontal justification)
      vjust = 1,          # 수직 정렬 (vertical justification)
      color = 'black',    # 글자 색상
      size = 8            # 글자 크기
))

# ------------------------------------------------------
# 2-4. 롤리팝 그래프 (geom_segment + geom_point)
# ------------------------------------------------------
install.packages("gridExtra")
library(gridExtra)
mt = mtcars

# (1) x축 기준 롤리팝
# x축에서 수직선으로 연결된 그래프:  geom_segment()의 aes(xend=변수x, yend=0) 설정
ggplot(mt, aes(x = mpg, y = disp)) +
  geom_segment(aes(xend = mpg, yend = 0, color = gear), size = 1.3) +  # y = 0부터 y = disp 까지 수직선(위)
  geom_point(aes(color = gear), size = 6) +                            # 점 찍기
```

```
  scale_color_continuous(type = "gradient") +                              # 색상
  theme_minimal()                                                          # 배경(깔끔)

# (2) y축 기준 롤리팝
# y축에서 수직선으로 연결된 그래프: geom_segment()의 aes(xend=0, yend=변수y) 설정
ggplot(mt, aes(x = mpg, y = disp)) +
  geom_segment(aes(xend = 0, yend = disp, color = gear), size = 1.3) + # x = 0부터 x = mpg 까지 수평선 (오른쪽으로)
  geom_point(aes(color = gear), size = 6) +                                # 점 찍기
  scale_color_continuous(type = "viridis") +                               # 색상
  theme_minimal()                                                          # 배경(깔끔)

# ------------------------------------------------------
# 2-5. geom_point()의 다양한 옵션
# ------------------------------------------------------
g1=ggplot(mt, aes(x=hp, y=mpg))

# ● 단순 점
g1+geom_point()

# ● 색상 지정
g2= g1+geom_point(color ='blue')
g2

# ● 그룹별 색상 (am: 0 = 자동, 1 = 수동 (두 가지 색))
g3= g1+geom_point(aes(color =factor(am)))
g3

# ● 사이즈 변경
g4 = g1+geom_point(size=7)
g4

# ● 각각 사이즈 조절
g5=g1+geom_point(aes(size=wt))
g5

# ● 모양 변경
g6=g1+geom_point(aes(size=wt, shape=factor(am)))
g6

# ● 색상을 다르게
g7 = g1+geom_point(aes(size=wt, shape=factor(am), color=factor(am)))
g7

# ● 원하는 색으로
g8 = g1+geom_point(aes(size=wt, shape=factor(am), color=factor(am)))+
        scale_color_manual(values=c('red','green'))
g8

# ● 라인 추가(점 + 선)
g9 = g1+geom_point(aes(size=wt, shape=factor(am), color=factor(am)))+
        scale_color_manual(values=c('red','green'))+
        geom_line()
g9
```

```r

# ● 축 이름 변경
g10 =g1 + geom_point(aes(size=wt, shape=factor(am), color=factor(am))) +
          scale_color_manual(values=c('red','green')) +
          geom_line() +
          labs(x='마력', y='연비')
g10


# ------------------------------------------------------
# 2-6. 선 그래프 (geom_line)
# ------------------------------------------------------
th = read.csv("학생별과목별성적_3기_3명.csv")
ss = arrange(th, 이름, 과목)

# 학생별 과목 점수 이름 기준으로 선 연결 + 점 표시
# - 한 학생이 여러 과목을 가지고 있기 때문에(6개) 표현도 그만큼 해줘야함 6개를 이름으로 묶어서 표현
ggplot(ss, aes(x=과목, y=점수, group=이름, color=이름)) +
  geom_line() +
  geom_point(size=6, shape=22) #shape : 0~25

# ======================================================
# R 데이터분석을 위한 패키지 설치
# ======================================================
# 1. 다국어 처리를 위한 멀티링구얼 패키지
install.packages("multilinguer")
library(multilinguer)

# 2. KoNLP 설치 전 필수 패키지 (문자열 처리, 사전, 데이터 처리용)
install.packages(c('string','hash','tau','Sejong','RSQLite','devtools'), type='binary')

# 3. GitHub에서 KoNLP 설치 (공식 지원 종료 → 수동 설치)
install.packages("remotes")
remotes::install_github("haven-jeon/KoNLP", upgrade='never', INSTALL_opts=c("--no-multiarch"), force=TRUE)

# 4. KoNLP 기본 설정
library(KoNLP)
useSejongDic() # 세종 사전 사용

# ======================================================
# 워드클라우드 : 텍스트 데이터를 시각적으로 표현
# ======================================================
# - 텍스트 데이터에서 단어를 추출하고, 단어의 빈도수에 따라 글자 크기를 다르게 표시
# - 핵심 패키지: KoNLP (명사 추출), wordcloud (시각화), RColorBrewer (색상 팔레트)

# ------------------------------------------------------
# 0. 시각화에 필요한 필수 패키지 설치 및 로딩
# ------------------------------------------------------
install.packages("stringr")
install.packages("wordcloud")
install.packages("RColorBrewer")

```

```r
library(stringr)                    # 문자열 처리 (단어 전처리)
library(wordcloud)                  # 워드클라우드 생성
library(RColorBrewer)               # 색상 팔레트 제공

# ------------------------------------------------------
# 1. 데이터에서 단어만 추출
# ------------------------------------------------------
d1 = readLines("BTS유엔연설_국문.txt")            # 텍스트 파일 한 줄씩 읽기
d2 = sapply(d1, extractNoun, USE.NAMES = FALSE)   # extractNoun로 명사를 추출 (벡터형 반환)

# ------------------------------------------------------
# 2. 단어 집합 생성 (unlist 이용)
# ------------------------------------------------------
d3 = unlist(d2)                                   # 리스트 → 벡터 변환

# ------------------------------------------------------
# 3. 단어 필터링 (2글자 이상 단어만 추출)
# ------------------------------------------------------
d3=Filter(function(x){
               nchar(x) >=2
               }, d3)

# ------------------------------------------------------
# 4. 단어 핸들링 (불필요한 단어 제거 - 생략 가능)
# ------------------------------------------------------
# 불필요한 단어를 일일히 삭제하기 힘들어 미리 목록작성 후 반복문으로 제거
# 남산 이라는 단어가 있으면 남/산으로 각각 쪼개지기 때문에 쪼개지지 않게 설정

# ------------------------------------------------------
# 5. 텍스트 파일로 저장 후 table로 다시 불러오며 공백제거
# ------------------------------------------------------
write(unlist(d3), "BTS_kor.txt")                  # 단어 벡터를 파일로 저장
d4 = read.table("BTS_kor.txt")                    # 다시 불러옴→ 데이터프레임 형태(공백제거)

# ------------------------------------------------------
# 6. 단어 빈도수 저장 (table() 함수)
# ------------------------------------------------------
wc = table(d4)

# ------------------------------------------------------
# 7. 워드클라우드 출력
# ------------------------------------------------------
pal = brewer.pal(9, "Set3")         # 색상 팔레트 정의

wordcloud(words = names(wc),        # 단어 목록 (table의 이름)
          freq = wc,                # 빈도수
          scale = c(5, 1),          # 단어 크기 (가장 큰 단어: 5, 작은 단어: 1)
          rot.per = 0.25,           # 회전될 단어 비율 (25%)
          min.freq = 2,             # 최소 빈도수: 2번 이상 등장만 표시
          random.color = TRUE,      # 색상 랜덤 적용
          random.order = FALSE,     # 빈도 높은 단어 중심으로
          colors = pal)             # 색상 팔레트 적용
```