Potato Snacks Analysis - 김한범

전체 개요

1. 주제 선정 과정

    (1) 과자 시장 규모

    (2) 유형별 과자 비율

    (3) 감자과자 선정 및 이유

2. 데이터 수집 및 분석 개요

    (1) 데이터 수집: 수집 대상, 수집 출처, 수집방법

    (2) 분석 방법 및 개요

3. Text Mining - 빈도분석

    (1) word cloud : 트위터 워드 클라우드와 네이버 블로그 워드 클라우드 비교

    (2). word frequency : 단어들 빈도 분석

4. Text Mining - 상관분석

    (1). 상관관계(network)

    (2). Graphical Lasso

5. Text Mining - 연관규칙

    (1). 연관규칙 생성

    (2). 연관규칙 시각화

6. Time series analysis

    (1). 검색어 분석

    (2). 검색어 시계열 예측

7. Summary

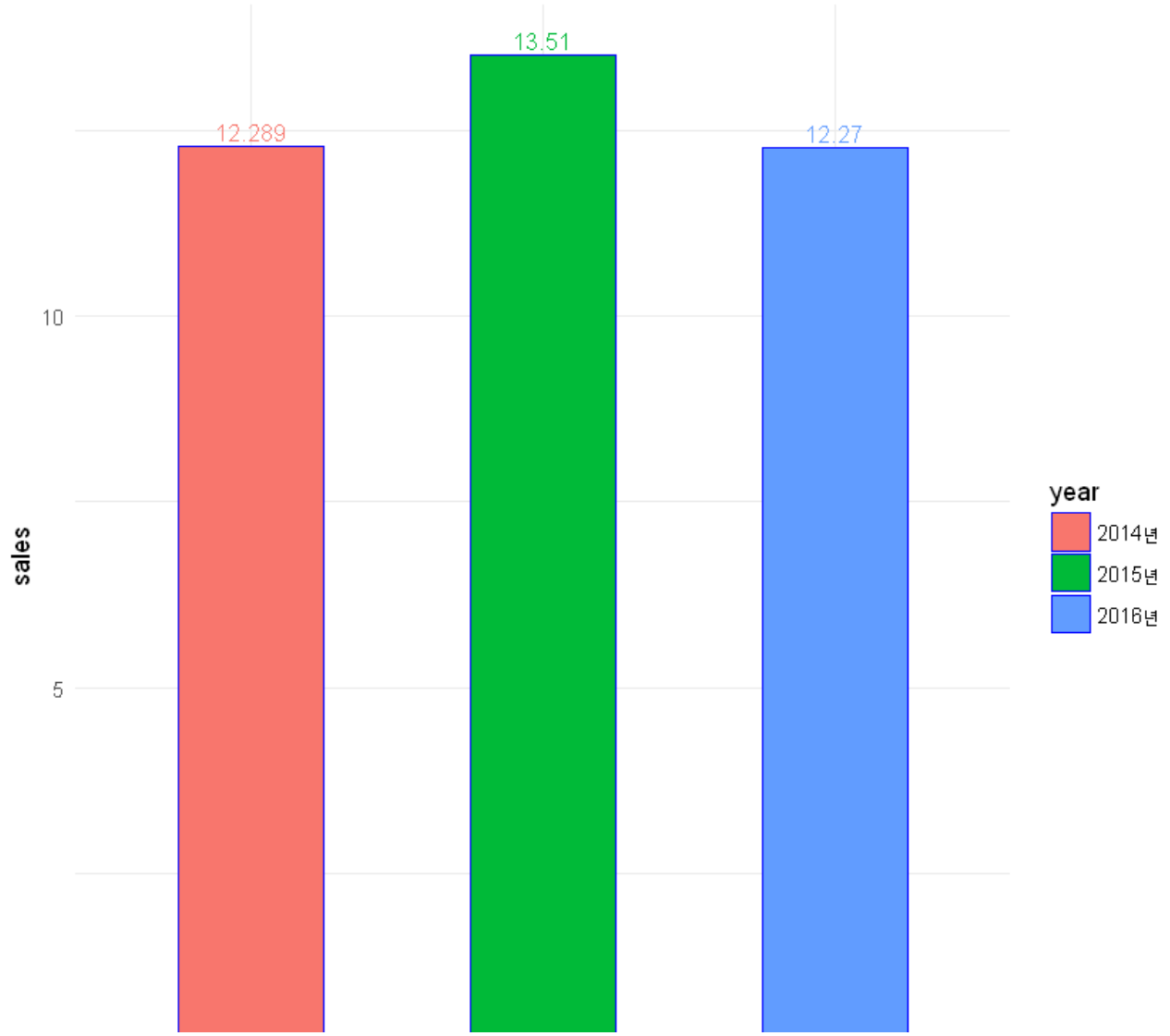
1. Text Mining 주제 선정 과정

1 - (1) 우리 나라 스낵 과자 시장 규모

In [17]:

```
snack_year <- data.frame(year=c("2014년", "2015년", "2016년"),
                 sales=c(12.289, 13.510, 12.270))
head(snack_year)

bp<-ggplot(data=snack_year, aes(x=year, y=sales, width = 0.5, color = year,
fill = year)) +
  geom_bar(stat="identity", color= 'blue')+
  geom_text(aes(label=sales), vjust=-0.3, size=3.5)+
    theme_minimal()
bp
```

year	sales
2014년	12.289
2015년	13.510
2016년	12.270



**-> 스낵과자 소매시장은 2014년 1조 2,289억원에서 2015년 1조 3,510억원으로 9.9% 증가했으나, 2016년 9.2% 감소하며 다시 정체되는 양상을 보이고 있다. **

- 식품 의약품 안전처에 의하면 과자류는 식물성원료 등을 주원료로 하여 다른 식품 또는 식품첨가물을 가하여 가공한 과자(비스킷류, 스낵과자류 등), 한과, 캔디(사탕,캐러멜, 양갱, 젤리 등), 추잉껌(무설탕껌, 일반추잉껌, 풍선껌), 빙과류를 말한다. 이 리포트에서는 과자류 중 가공한 과자에 속하는 스낵과자를 다룬다.

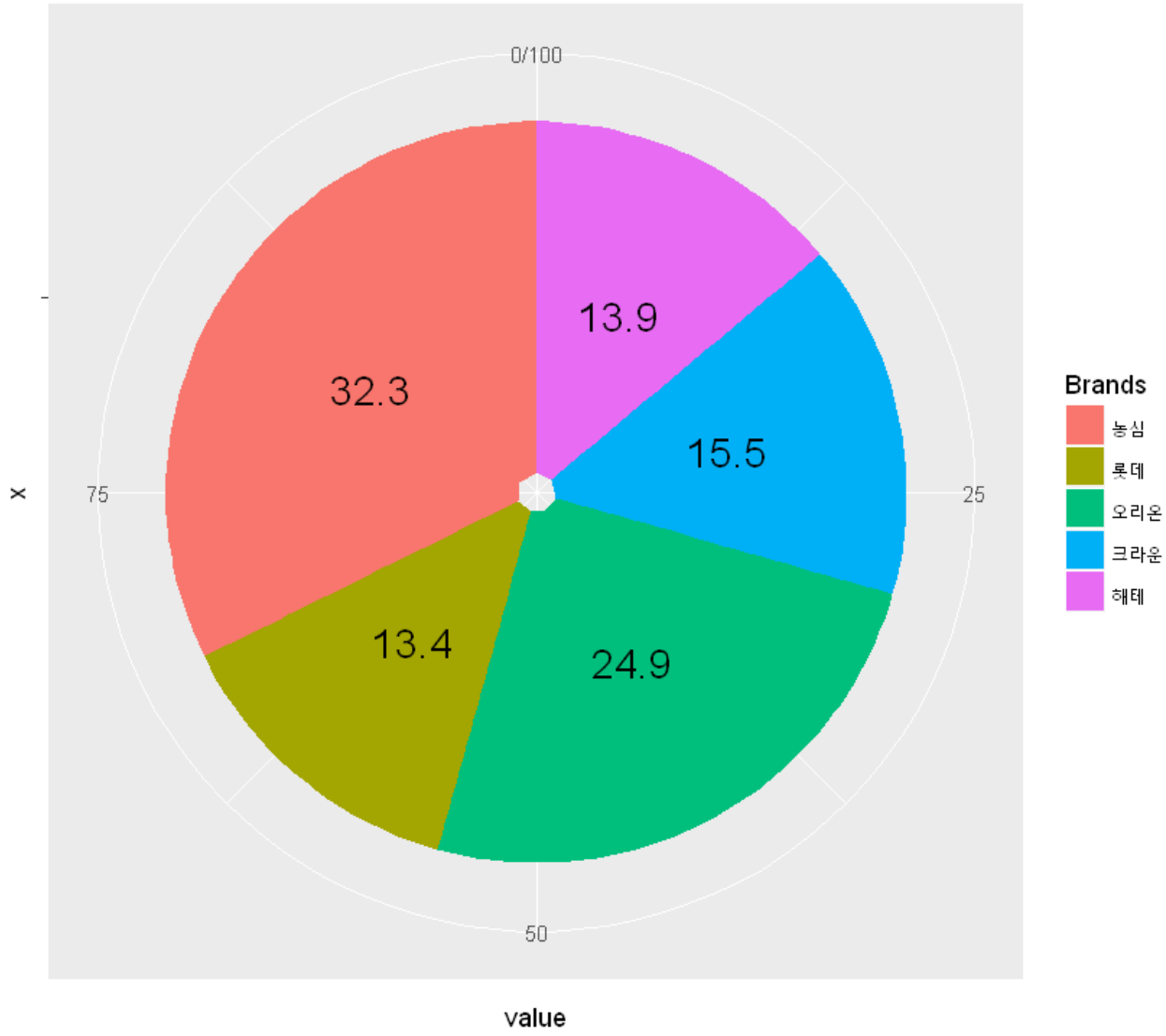
** 자료: 식품산업통계정보-www.atfix.or.kr*

1 - (2) 국내 스낵 과자 시장 점유율

In [62]:

```
snack_brand <- data.frame(
  group = c("오리온", "농심", "해테","롯데",'크라운'),
  value = c(24.9, 32.3, 13.9,13.4,15.5)
  )
head(snack_brand)

blank_theme <- theme_minimal()+
  theme(
  axis.title.x = element_blank(),
  axis.title.y = element_blank(),
  panel.border = element_blank(),
  panel.grid=element_blank(),
  axis.ticks = element_blank(),
  plot.title=element_text(size=14, face="bold")
  )


cp<- ggplot(snack_brand, aes(x='', y=value, fill=group))+
geom_bar(stat = "identity")+
coord_polar("y")+
geom_text(aes(label = value),position=position_stack(vjust=0.5), size=6)+
labs(fill='Brands')

cp
```

group	value
오리온	24.9
농심	32.3
해테	13.9
롯데	13.4
크라운	15.5



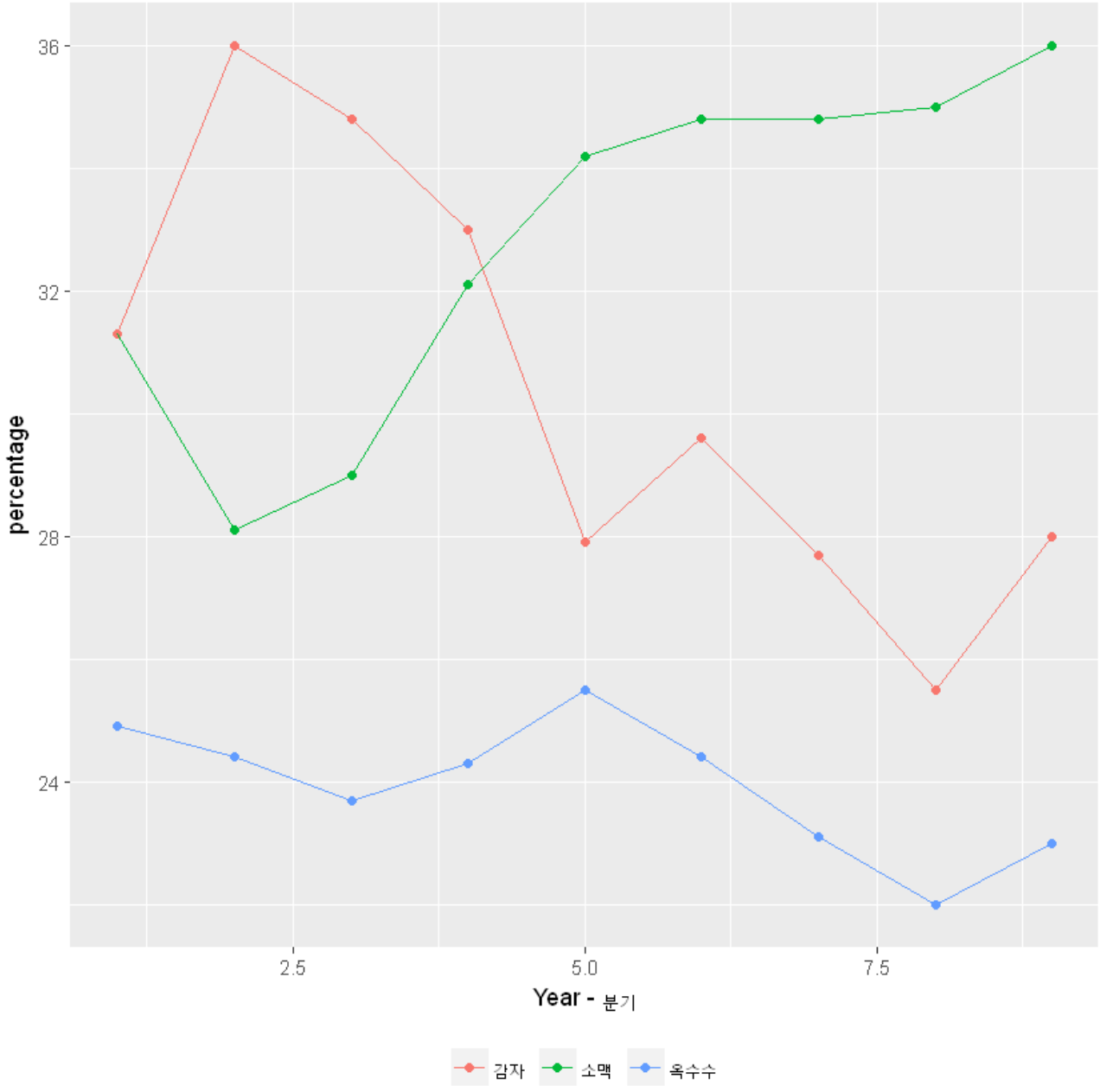
-> 국내 스낵 과자 시장 점유율은 매출액을 기준으로 농심-오리온-해태-롯데-크라운 순이다. (2016년 1분기 기준)
** 자료: AC 닐슨*

1 - (3) 과자 매출 비교(주요 원료별)

In [106]:

```
snack_raw <- read.csv('C:/Users/hanbum/Desktop/Data/Visualization Task/snack/snack_raw.csv')
head(snack_raw)

g1 <- ggplot(aes(y = percentage, x = as.numeric(year), colour = product), size=2,
                          data = snack_raw, stat="identity") +
geom_point() +
geom_line()+ xlab('Year - 분기')+
theme(legend.position="bottom", legend.direction="horizontal", legend.title = element_blank())


g1
```

product	year	percentage
소맥	2015-04-01	31.3
소맥	2015-07-01	28.1
소맥	2015-10-01	29.0
소맥	2015-12-01	32.1
소맥	2016-04-01	34.2
소맥	2016-07-01	34.8



-> 위 그래프는 2015년 1분기 부터 2017년 1분기까지, 총 9분기의 과자 원료별 Market Share을 시간 순대로 표현한 것이다
-> 2017년 1분기 매출 기준으로 소맥과자가 37.2%로 가장 많은 비중을 차지 하고 있으며, 감자, 옥수수 순으로 상위 3가지 유형이 전체의 88.2%를 차지한다.

-> 허니버터칩의 등장으로 매출 1위를 차지했던 감자과자는 인기가 사그라진 모습이다</big>

** 소맥과자는 밀을 원료로 한 과자를 뜻함(**ex. 새우깡, 꿀꽈배기**)*

** 자료 : 식품산업통계정보(**www.atfis.or.kr**)*

1 - (4) 주제 선정

-> 감자를 주원료로 하는 과자는 짭짤한 대중적인 맛과 다양한 양념에 잘 어울리기 때문에, 일정한 수익을 보장해주는 제과업계의 효자 상품이다.

-> 2014년 하반기에 나온 허니버터칩 열풍은 감자 과자를 효자 상품군에서 주력 상품군으로 확장시켰다.
3년이 지난 현재에는 감자 과자의 매출은 허니버터칩 열풍 이전으로 돌아간 상태이다,

-> 그러나, 스낵 과자 시장의 전통의 강자인 감자 과자 시장의 패권을 차지하기 위해 각 업체들은 각기 다른 마케팅을 시도하는 중이다.

-> 따라서, 이 리포트에선 각 제과 업체별로 감자 과자의 주력상품들을 선정하여 **Text Analysis**를 통해 소비자들이 해당 감자 과자에 대해 어떻게 생각하는지를 분석하고자 한다.

-> 위와 같은 분석을 하기 위한 데이터는 네이버 블로그, 트위터, 페이스북 **Crawling**을 통해 모은다.

->여기에 더해, 시계열 분석을 통해 해당 감자 과자 검색어 에측을 해보도록 한다.

선정한 과자들 **= #**오리온**-**포카칩 **/** 농심**-**수미칩 **/** 해테 **-** 허니버터칩 **/** 롯데 **-** 레이즈**(lays)** 감자칩**, PB-**이마트 노브랜드 감자칩

In [107]:

```
display_png(file='C:/Users/hanbum/Desktop/Data/Visualization
Task/snack/포카칩.png')
display_png(file='C:/Users/hanbum/Desktop/Data/Visualization
Task/snack/수미칩.png')
display_png(file='C:/Users/hanbum/Desktop/Data/Visualization
Task/snack/허니버터칩.png')
display_png(file='C:/Users/hanbum/Desktop/Data/Visualization
Task/snack/레이즈.png')
display_png(file='C:/Users/hanbum/Desktop/Data/Visualization
Task/snack/노브랜드감자칩.png')
```


Original
生
끝까지! 생감자칩


농심
맛있는 수미감자
秀美칩
[빼어날 수] [아름다울 미]
Original


Calbee
달콤~햐
생꿀이 듬뿍
100%
허니버터칩
HONEY BUTTER CHIP


Frito Lay
Lay's
BRAND

2. Text Mining 데이터 수집 및 분석 개요

2 - (1) 데이터 수집

수집 대상 : 오리온-포카칩 / 농심-수미칩 / 해테 - 허니버터칩 / 롯데 - 레이즈 감자칩 / PB-이마트 노브랜드 감자칩

수집 방법 : R을 통한 크롤링

수집 출처 : 네이버 블로그, 트위터, 페이스북

2 - (1) - (i). 데이터 수집(네이버 블로그)

In [130]:

```
#네이버 블로그 Text 수집
#네이버 API 아이디와 비밀번호

Sys.setlocale("LC_ALL", "Korean")
client_id = '5vYAgVwBHv7rgNU2l7Yv';
client_secret = 'WAqz45PQfV';
header = httr::add_headers('X-Naver-Client-Id' = client_id,
                           'X-Naver-Client-Secret' = client_
secret)

# 검색할 단어와 encoding 변환

query.n =  query ='이마트 노브랜드 감자칩'
query = iconv(query, to = 'UTF-8', toRaw = T)
query = paste0('%', paste(unlist(query), collapse = '%'))
query = toupper(query)

# url 저장(여기선 블로그 url)
i = 1
end_num = 1000
display_num = 100
start_point = seq(1,end_num,display_num)

url = paste0('https://openapi.naver.com/v1/search/blog.xml?q
uery=',query,'&display=',display_num,'&start=',start_point[i
],'&sort=sim')
url_body = read_xml(GET(url, header))

# 텍스트 정보 추출
final_dat = NULL

for(i in 1:length(start_point))
{
  url =
paste0('https://openapi.naver.com/v1/search/blog.xml?query='
```

```
,query,'&display=',display_num,'&start=',start_point[i],'&so
rt=sim')
  url_body = read_xml(GET(url, header), encoding = "UTF-8")
  title = url_body %>% xml_nodes('item title') %>% xml_text(
)
  bloggername = url_body %>% xml_nodes('item bloggername') %
>% xml_text()
  postdate = url_body %>% xml_nodes('postdate') %>% xml_text
()
  link = url_body %>% xml_nodes('item link') %>% xml_text()
  description = url_body %>% html_nodes('item description')
%>% html_text()
  temp_dat = cbind(title, bloggername, postdate, link, descr
iption)
  final_dat = rbind(final_dat, temp_dat)
  cat(i, '\n')
}

final_dat = data.frame(final_dat, stringsAsFactors = F)

head(final_dat)

# 형태소 분석을 위한 데이터 정제
dat_tmp <- final_dat
for (i in 1:nrow(final_dat))
{
  dat_tmp[i,5]<-   gsub(pattern = "<[/|A-Za-z]*>",
       replace = "", final_dat[i,5])
}
```

```
'LC_COLLATE=Korean_Korea.949;LC_CTYPE=Korean_Korea.949;LC_MONETA
```

```
1
2
3
4
5
6
7
8
9
10
```

title	bloggername	postdate	link	descript
				이마 노브랜드 자칩 마트 노브 드 제품이 나오면서 희집 식토

title	bloggername	postdate	link	description
이마트 노브랜드 감자칩	먹고노는 이야기	20170913	http://blog.naver.com/thro4321?Redirect=Log&logNo=221094853871	늘이 많아 바뀐듯 같다. 집안 곳저곳 살 보다보면 아.. 나도 르게 많이 바뀐게 많 나 해요.. 요일 저녁 와이프하 아들녀석 이...
이마트 노브랜드 감자칩 오리지널 후기 & 영양성분 칼로리	이쿤의 봄여름가을겨울	20170813	http://blog.naver.com/kun-lee?Redirect=Log&logNo=221072810113	단거좋아 시는분은 브랜드 고 마칩을 더 좋아하더 구요~ <b 이마트 노 랜드 감자 오리 널 후기 ㄱ 만약 이마 가 가까이 없다면 이 트몰을 이 하는것도 법이에요 <이마 인터넷쇼 몰 바로가 기...
이마트 24 노브랜드 감자칩 사워크림 & 어니언	Juny's Blog	20170919	http://blog.naver.com/hbj1646?Redirect=Log&logNo=221099994704	이마 24 < 노브랜드 자칩 워크림 & 어 언 후기가 적으로 < 노브랜드 에서 오는 제품 중 추천하 스낵 중 하 나입니다 감자 가격 저렴한데

title	bloggername	postdate	link	description
				슷합니다 짭짤하고 삭한 맛..
이마트 노브랜드 감자칩 오리지널 & 자색고구마칩 솔직한 리뷰!	party shadow	20170818	http://blog.naver.com/ysju94?Redirect=Log&logNo=221077098961	이마 노브랜드 자칩 리지널 & 자 고구마칩 직한 리뷰 이마트 노 랜드하면 자칩 오리 널, 샤워크 림 &amp 자색고구 칩을 많이 드시는데 최근에도 을정도로 용하고 있 노브랜드 자칩...
이마트 노브랜드 감자칩 자색고구마칩 가격도 맛도 좋네요	술패랭이	20170606	http://blog.naver.com/1030ysk?Redirect=Log&logNo=221022678731	일요일에 랑과 마트 다녀왔어 사실 마트 자주 가는 편은 아닌 우리 신랑 워낙 쇼핑 을... 브랜드 감 칩과 색고구마 애들이 좋 하는 같은 형태의 오 브랜드 < 감자칩</ 은 너무 비 싸요. 요즘
				이번에 먹 본 과자는 이마 노브랜드 자칩

title	bloggername	postdate	link	description
이마트 노브랜드 감자칩 사워크림 & 어니언	좀좀이의 여행	20170328	http://zomzom.tistory.com/2007	워크림 &어니언이에요 이 과자는 노브랜드 품 중 꽤 기가 많다 알려진 제 이에요. 이 과자의 정 은 천원 초 안 된다는 점. 그렇다 고 해서 작 소...

In [ ]:

```
#crawling data들을 csv파일로 저장해두자
write.table(posts,'poca_blog.csv',sep=',',row.names=F)
```

2 - (1) - (ii). 데이터 수집(트위터)

In [2]:

```
#key확보 및 정상 작동 확인
consumer_key ='RI2MdUEhcKVBqSDDwzKjnlShQ'
consumer_secret ='l1Lf9W3X8QjcwYdSl9amVRNPllNfuQFfJCwsjNEClmIpnNyaEg'
access_token = '884200639504334848-XeH2JlMrufay8vUV9hQ4EwLhH0tz5M4'
access_secret = '1iX6m5a4y0wofmA1s9l1VROxF48NgtvxqxtllOxPYXD7Y'

setup_twitter_oauth(consumer_key,consumer_secret,access_token,access_secret)

#찾고자 하는 단어 입력, 3000개 트위터 검색
string <- '이마트 노브랜드 감자칩'
string <- iconv(string, 'CP949', 'UTF8')
tweets <- searchTwitter(searchString = string, n = 3000, lang="ko", retryOnRateLimit = 10000, )
head(tweets)

# 결과 중에서 텍스트에 해당하는 부분만 추출하고 확인
text_extracted <- sapply(tweets, function(t) t$getText())

#불필요한 문자들은 모두 제거
text_extracted <- gsub("\n", "", text_extracted)
```

```
text_extracted <- gsub("\r", "", text_extracted)
text_extracted <- gsub("RT", "", text_extracted)
text_extracted <- gsub("http", "", text_extracted)
text_extracted <- gsub("rt", "", text_extracted)
text_extracted <- gsub("ㅋㅋㅋㅋㅋㅋ", "", text_extracted)
text_extracted <- gsub("@\\w+", "", text_extracted) #user na
me
text_extracted <- gsub("[ |\t]{2,}", "", text_extracted) #re
move tabs
text_extracted <- gsub("노브랜드", "", text_extracted)
text_extracted <- gsub("감자칩", "", text_extracted)
head(text_extracted)

# 문자 분리
result_nouns <- Map(extractNoun, text_extracted)
result_wordsvec <- unlist(result_nouns, use.name=F)
result_wordsvec <- result_wordsvec[-which(result_wordsvec %i
n% stopwords("english"))]
result_wordsvec <- gsub("[[:punct:]]","", result_wordsvec)
result_wordsvec <- Filter(function(x){nchar(x)>=2}, result_w
ordsvec)

head(result_wordsvec)
```

```
[1] "Using direct authentication"
```

```
Warning message in doRppAPICall("search/tweets", n, params =
params, retryOnRateLimit = retryOnRateLimit, :
"3000 tweets were requested but the API can only return 3"
```

```
[[1]]
[1] "owo_away: 어제 티지님 드린거: 후드티\n어제 짜누님 드린거: 자색
고구마칩, 이마트 노브랜드 감자칩 8ㅅ8 그리고 이룸 모르는 저거 https:
//t.co/mXhLDgtM5e"

[[2]]
[1] "knyck1107: 오는길에 이마트 편의점 들러서 샐러드랑 노브랜드 감자
칩 사왔는데 샐러드가 싱싱하니 가득 담겨있어서 기분이 좋다"

[[3]]
[1] "gyxp: 마트가니까 온리프라이스라고 누가봐도 이마트 노브랜드 베낀
거 있던데 감자칩 개맛없다 ㄹㅇ루"
```

'어제 티지님 드린거: 후드티어제 짜누님 드린거: 자색고구마칩, 이마트 8ㅅ8 그리고 이룸 모르는 저거 s://t.co/mXhLDgtM5e'
'오는길에 이마트 편의점 들러서 샐러드랑 사왔는데 샐러드가 싱싱하니 가득 담겨있어서 기분이 좋다'
'마트가니까 온리프라이스라고 누가봐도 이마트 베낀거 있던데 개맛없다 ㄹㅇ루'

In [ ]:

```
#crawling data들을 csv파일로 저장해두자
write.table(posts,'emart_twitt.csv',sep=',',row.names=F)
```

- 트위터 데이터는 허니버터칩, 포카칩, 수미칩만 모았다. 레이즈 감자칩과 이마트 노브랜드 감자칩도 모으고자 했으나, **error message**가 뜬다. 아마도, 일정기간동안 **crawling** 할 수 있는 양이 정해져있는 것 같다.

2 - (1) - (iii). 데이터 수집(페이스북)

- 페이스북 데이터를 모으기 위해선 **crawling**할 주소와 기간이 필요하다. 주소는 각 과자 업체의 공식 페이지로 했고, 기간은 **2015/01/1**부터 현재까지로 설정하였다.

In [3]:

```
# Facebook 데이터 수집

#토큰 받고 저장하기
fb_oauth=fbOAuth(app_id="1529704550421200", app_secret= "735
347178fa297dd1c597f88585e2ee3", extended_permissions=FALSE)

#내 계정으로 확인
getUsers('me',token=fb_oauth)

start_date='2015/01/01'
end_date='2017/09/24'
scrape_days=seq(from=as.Date(start_date), to=as.Date(end_dat
e), by='days')
posts=c()

for(scrape_day in scrape_days)
{
  daypost=c()
  tryCatch({daypost=getPage(page="onlyorion",
                            token=fb_oauth,
                            since=as.Date(scrape_day, origin
="1970-01-01"),
                            until=as.Date(scrape_day, origin
="1970-01-01")+1
                            )
             },
           error=function(e){}
           )
  posts= rbind(posts, daypost)
}
head(posts)
```

```
Copy and paste into Site URL on Facebook App Settings: http:
//localhost:1410/
```

```
When done, press any key to continue...
```

```
Waiting for authentication in browser...
Press Esc/Ctrl + C to abort
Please point your browser to the following url:
https://www.facebook.com/dialog/oauth?client_id=152970455042
1200&scope=public_profile%20user_friends&redirect_uri=http%3
%2F%2Flocalhost%3A1410%2F&response_type=code&state=CXdhHlyd6

Authentication complete.
Authentication successful.
```

id	name	username	first_name	middle_name	last_name	g
279212895911049	Hanbum Kim	NA	NA	NA	NA	N

In [18]:

```
#crawling data들을 csv파일로 저장해두자
write.table(posts,'poca_facebook.csv',sep=',',row.names=F)
```

*페이스북 자료는 분석에 사용하지 못했다. 페이스북 자료를 crawling하는 연습 정도의 의미로 이 리포트에 실었다.

2. (2) 분석 개요

지금까지 **Text Analysis**를 위한 **text** 데이터를 네이버 블로그, 트위터, 페이스북 **Crawling**을 통해 모았다. 다음 장에서는 모은 데이터를 활용해 실제로 이 데이터가 어떤 의미를 가지고 있는 지 분석해 보자.

분석 방법과 절차는 아래와 같다.

3. Text Mining - 빈도분석

4. Text Mining - 상관분석

5. Text Mining - 연관규칙

6. Time series analysis - 시계열 분석
****

사용한 라이브러리 -

library_basic = c('dplyr', 'ggplot2','igraph')

library_crawling = c('rvest', 'httr','xml2','curl','twitteR','Rfacebook')

library_textM = c('KoNLP','tm','Matrix','wordcloud')

library_visual=c('qgraph','rgl','scales','lRdisplay','scales','RColorBrewer')

library_association=c('arules','arulesViz')

library_times=c('tseries','TTR','forecast')

---

3. Text Mining - 빈도분석

3- (1) Wordcloud 비교

1. 네이버 블로그와 트위터에서 모은 각 과자 데이터별로 **wordcloud**를 그려 보고, 어떤 차이가 있는지 알아보자

In [68]:

```
#데이터를 불러오자
dat_tmp<-read.csv('C:/Users/hanbum/Desktop/Data/Visualizatio
n Task/snack/sumi_blog.csv') #네이버 블로그 데이터
head(dat_tmp)
str(dat_tmp)

result_wordsvec<-read.csv('C:/Users/hanbum/Desktop/Data/Visu
alization Task/snack/sumi_twitt.csv') #트위터 데이터
head(result_wordsvec)
str(result_wordsvec)

# 사전을 불러오자
useSejongDic()

# tm package is based on Eng. Addition option is required
text = dat_tmp[,5]
cps = Corpus(VectorSource(text))
dtm = tm::DocumentTermMatrix(cps,
             control = list(tokenize = extractNoun,
                                       removeNumber = T,
                                       removePunctuation = T)
)
str(dtm)

# matrix class
rmat <- as.matrix(dtm)

# sparseMatrix
rmat <-spMatrix(dtm$nrow,dtm$ncol, i=dtm$i, j=dtm$j, x=dtm$v
)
```

```
wcount<-colSums(rmat)
wname <- dtm$dimnames$Terms
wname <- repair_encoding(dtm$dimnames$Terms)
colnames(rmat)<- wname

sort.var <- sort(wcount,decreasing = T)[100]
idx <- !( grepl('수미칩', wname)| (wcount<=sort.var) )
wname.rel <- wname[idx]
wcount.rel <- wcount[idx]

pal <- brewer.pal(9, "Set2")
# twitter wordcloud
result_wordcount <- table(result_wordsvec)

pal <- brewer.pal(7, "Set1")
```

```
Warning message in scan(file = file, what = what, sep = sep,
quote = quote, dec = dec, :
"따옴표로 묶인 문자열내에 EOF가 있습니다"
```

title	bloggername	postdate	link	descriptio
수미칩 프라임 새로운 두께감과 맛	또군공간	20170807	http://blog.naver.com/ddo_gun?Redirect=Log&logNo=221068329912	오늘은 수미칩 프라임에 대해서 포스팅을 하려합니다. 평소 감자 과자 중에서는 수미칩이 참 맛있다고 생각하는 사람 중 한 명인데, 이번에 수미칩 프라임이라고 나왔더라구요 평소 수미칩과는 전혀 다른...
수미칩 허니 머스타드 맛	피부 새로고침 ♥ 아우름클리닉	20170809	http://blog.naver.com/jeon6367?Redirect=Log&logNo=221070392923	그 뒤로 너무 맛있어서 자주 즐겨먹고 있답니다 ㅎㅎ 한때 허니버터칩 대체품으로 나온 것 같은데 호불호는 갈리겠지만 저는 수미칩이 더

달콤해	bloggername	postdate	link	만들라고요 ㅎㅎ 수미칩이 허니버터칩보다 몇백원 비싸지만 양도 많고...
수미칩 허니머스타드에 밀맥주 한잔 좋아요~	상크미의 해피월드 ★	20170619	http://hanpost33.tistory.com/1198	좋아하는 수미칩과 제일 좋아하는 맥주 파울라너 후기를 남겨봐요 ㅎ 독일 바이에른 뮌헨의 대표적인 밀맥주 파울라너 허페바이스 그리고 농심 수미칩 허니머스타드, 이건 허니버터칩이 인기몰이하면서 같이...
커피 디저트 수미칩 아메리카노와 꿀조합	쌍오	20170324	http://blog.naver.com/foren00n?Redirect=Log&logNo=220957445349	수미칩이 사로워졌어요 여러분 전에 알던 수미칩 아냐 브랜뉴 사운 수미칩이 더 두꺼워져서 프라임으로 태어난거 다들 아시나요!? 평상시 커피 디저트를 아타게 찾던 제가 이 수미칩 프라임을 맛보고...
수미칩 프라	지구인 이럿저		http://blog.naver.com/gkmovingon?	수미칩프라임 초콜릿이었습니다. 마트에서 보자마자 바로 카트에 담았습니다. 와 이프님께 선... 수미칩

title	bloggername	postdate	link	description
초콜릿과로이스감자칩	은은과 어은서	20170516	http://blog.naver.com/gkmovingon?Redirect=Log&logNo=221006609870	스...ㅜㅜ입초콜릿으로 좋아하는거 먹는다는 기분이라도 느끼셨으면 좋겠네요. 자 뜯어보겠습니다 과연...
수미칩 프라임 초코, 제 점수는요?	SLOWLY	20170512	http://blog.naver.com/slowly__?Redirect=Log&logNo=221004314650	수미칩 프라임 초코, 제 점수는요? 냥히랑 전화하는데 토이스토리 초콜릿얘기하다가 꼬북칩얘기하다가 수미칩은 도대체 어디있니 던 그녀. 내 눈앞에 있는데요? 그래서 샀다. 로이스감자칩을 떠올리며! (힝...

```
'data.frame': 834 obs. of  5 variables:
 $ title      : Factor w/ 793 levels "'<b>수미칩</b>' 프라임..
감자칩의 격을 높이다!",..: 177 347 356 629 382 178 444 29 457 1
83 ...
 $ bloggername: Factor w/ 790 levels "-블로그쟁이가 된 조대리",
..: 354 746 477 529 671 195 725 135 260 265 ...
 $ postdate   : Factor w/ 325 levels " <b>수미칩</b> 프라임 시
식후기,치노의 래빗하우
스,20170715,http://itboxtis.tistory.com/2019,편의점에 가니 수미
칩"| __truncated__,..: 318 319 313 279 302 300 301 312 295 2
82 ...
 $ link       : Factor w/ 834 levels " <b>수미칩</b> 허니머스타
드 비교 ( +꿀꽈배기),예그리나 chocolatier♡,20150624,http://blog.
naver.com/fonti_?Redi"| __truncated__,..: 168 332 802 227 24
5 640 152 823 401 154 ...
 $ description: Factor w/ 834 levels "","' 수미칩 허니버터칩 '
음 크기는 굉장히 컸어요 ! ㅋㅋㅋㅋ 기대감도 커졌지요 .. 생감자가 91%
들어가고, 용량은 저"| __truncated__,..: 574 131 664 512 514 38
0 670 699 638 572 ...
```

x

쓰로
술꾼
칩수미백술도
ㅋㅋㅋㅋㅋ
옛날
별명이네요보고

```
'data.frame': 235 obs. of  1 variable:
 $ x: Factor w/ 203 levels "<U+653C><U+3E64>愼<U+3E30><U+623
C><U+3E64><U+653C><U+3E64><U+623C><U+3E34>愼<U+3E62><U+653C>
<U+3E64>愼<U+3E30><"| __truncated__,..: 189 104 169 172 130
75 171 47 161 12 ...
Backup was just finished!
370957 words dictionary was built.
List of 6
 $ i       : int [1:17714] 1 1 1 1 1 1 1 1 1 1 ...
 $ j       : int [1:17714] 1 2 3 4 5 6 7 8 9 10 ...
 $ v       : num [1:17714] 1 1 1 1 1 1 1 1 1 2 ...
 $ nrow    : int 834
 $ ncol    : int 8171
 $ dimnames:List of 2
  ..$ Docs : chr [1:834] "1" "2" "3" "4" ...
  ..$ Terms: chr [1:8171] "감자" "과자" "나왔더라구요" "다른" ..
.
 - attr(*, "class")= chr [1:2] "DocumentTermMatrix" "simple_
triplet_matrix"
 - attr(*, "weighting")= chr [1:2] "term frequency" "tf"
```

```
Best guess: EUC-KR (100% confident)
Warning message in stringi::stri_conv(x, from = from):
"input data \xfffffffc2 in current source encoding could not
be converted to Unicode"Warning message in stringi::stri_con
v(x, from = from):
"input data \xfffffffc2 in current source encoding could not
be converted to Unicode"Warning message in stringi::stri_con
v(x, from = from):
"the Unicode codepoint \U0000008b cannot be converted to des
tination encoding"Warning message in stringi::stri_conv(x, f
rom = from):
"the Unicode codepoint \U0000009a cannot be converted to des
tination encoding"Warning message in stringi::stri_conv(x, f
rom = from):
"input data \xfffffffa0 in current source encoding could not
be converted to Unicode"Warning message in stringi::stri_con
v(x, from = from):
"input data \xfffffffd9 in current source encoding could not
be converted to Unicode"Warning message in stringi::stri_con
v(x, from = from):
"input data \xfffffffa0 in current source encoding could not
```

```
be converted to Unicode"Warning message in stringi::stri_con
v(x, from = from):
"input data \xffffffe4 in current source encoding could not
be converted to Unicode"Warning message in stringi::stri_con
v(x, from = from):
"input data \xffffffe7 in current source encoding could not
be converted to Unicode"Warning message in stringi::stri_con
v(x, from = from):
"the Unicode codepoint \U00000089 cannot be converted to des
tination encoding"Warning message in stringi::stri_conv(x, f
rom = from):
"the Unicode codepoint \U00000091 cannot be converted to des
tination encoding"Warning message in stringi::stri_conv(x, f
rom = from):
"input data \xffffffa1 in current source encoding could not
be converted to Unicode"Warning message in stringi::stri_con
v(x, from = from):
"the Unicode codepoint \U00000092 cannot be converted to des
tination encoding"Warning message in stringi::stri_conv(x, f
rom = from):
"input data \xffffff8e in current source encoding could not
be converted to Unicode"Warning message in stringi::stri_con
v(x, from = from):
"input data \xffffffd4 in current source encoding could not
be converted to Unicode"Warning message in stringi::stri_con
v(x, from = from):
"input data \xffffff8e in current source encoding could not
be converted to Unicode"Warning message in stringi::stri_con
v(x, from = from):
"input data \xffffffc4 in current source encoding could not
be converted to Unicode"Warning message in stringi::stri_con
v(x, from = from):
"input data \xffffffc4 in current source encoding could not
be converted to Unicode"Warning message in stringi::stri_con
v(x, from = from):
"input data \xffffffee in current source encoding could not
be converted to Unicode"Warning message in stringi::stri_con
v(x, from = from):
"the Unicode codepoint \U0000009f cannot be converted to des
tination encoding"Warning message in brewer.pal(9, "Set2"):
"n too large, allowed maximum for palette Set2 is 8
Returning the palette you asked for with that many colors
"
```

In [65]:

```
# 허니버터칩 wordcloud
```

```
wordcloud(wname.rel, freq = wcount.rel,rot.per=0.35,random.o
rder=FALSE,colors = pal)
wordcloud(names(result_wordcount), freq=result_wordcount,
          scale=c(4,0.5), min.freq=12, random.order=F, rot.p
er=0.35,
          colors=pal, family="malgun")
```

```
Warning message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in wordcloud(names(result_wordcount), freq = result_
wordcount, scale = c(4, :
"<U+653C><U+3E64>愼<U+3E30><U+623C><U+3E63><U+653C><U+3E64><
U+623C><U+3E64>愼<U+3E66>크크크크 could not be fit on page
```

```
It will not be plotted."Warning message in strwidth(words[i]
, cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"
```

* 허니버터칩 **wordcloud** 비교

--네이버 블로그 : '만들다'는 단어와 '편의점', '드디어', '오늘'이란 단어가 많이 보인다

-- 트위터 : '편의점'이란 단어가 많이 보이고, 다른 단어들은 어떤 연관성이 있는지 알기 어려워 보인다

비교 : 허니버터칩과 편의점은 공통 분모를 가지고 있는 것으로 보인다

In [67]:

```
# 포카칩 wordcloud

wordcloud(wname.rel, freq = wcount.rel,rot.per=0.35,random.o
rder=FALSE,colors = pal)
wordcloud(names(result_wordcount), freq=result_wordcount,
          scale=c(4,0.5), min.freq=12, random.order=F, rot.p
er=0.35,
          colors=pal, family="malgun")
```

```
Warning message in strwidth(words[i], cex = size[i], ...):
```

"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning message in text.default(x1, y1, words[i], cex = size[i], offset = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning message in text.default(x1, y1, words[i], cex = size[i], offset = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning message in text.default(x1, y1, words[i], cex = size[i], offset = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning message in text.default(x1, y1, words[i], cex = size[i], offset = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning message in wordcloud(names(result_wordcount), freq = result_wordcount, scale = c(4, :
"떡밥만즐기는짓은안하시죠에너제틱한워너블스밍파란색악개나야 could not be fit on page. It will not be plotted."Warning message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning message in text.default(x1, y1, words[i], cex = size[i], offset = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning message in text.default(x1, y1, words[i], cex = size[i], offset = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning message in text.default(x1, y1, words[i], cex = size[i], offset = 0, srt = rotWord * :

```
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
```

```
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트패밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트패밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트패밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트패밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트패밀리입니다"
```

* 포카칩 **wordcloud** 비교

--네이버 블로그 : '맛과 관련된 검색어가 많이 보인다.(스윗치즈, 구운김맛,불닭맛 등등), 이외에도 다른 감자칩도 연관검색어로 많이 보인다.(수미칩,허니버터칩) 허니버터칩과는 달리 제조업체인 '오리온'이 연관 단어로 보인다는 것도 확인할 수 있다

-- 트위터 : 역시 맛과 관련된 검색어가 보이고, 단어 정제가 잘 안되어서 제대로 관계를 확인하기는 어려워 보인다.

비교 :포카칩은 맛과 관련된 단어가 많은 비중을 차지한다. 포카칩이 다양한 맛을 출시하기 때문인걸로 보인다

In [69]:

```
# 수미칩 wordcloud

wordcloud(wname.rel, freq = wcount.rel,rot.per=0.35,random.order=FALSE,colors = pal)
wordcloud(names(result_wordcount), freq=result_wordcount,
          scale=c(4,0.5), min.freq=12, random.order=F, rot.per=0.35,
          colors=pal, family="malgun")
```

```
Warning message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning message in text.default(x1, y1, words[i], cex = size[i], offset = 0, srt = rotWord * :
```

```
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in wordcloud(names(result_wordcount), freq = result_
wordcount, scale = c(4, :
"<U+653C><U+3E64>慎<U+3E30><U+623C><U+3E64><U+653C><U+3E64><
U+623C><U+3E34>慎<U+3E62><U+653C><U+3E64>慎<U+3E30><U+623C><
U+3E64><U+653C><U+3E64><U+623C><U+3E38>慎<U+3E30> could not
be fit on page. It will not be plotted."Warning message in w
ordcloud(names(result_wordcount), freq = result_wordcount, s
cale = c(4, :
"<U+653C><U+3E64>慎<U+3E30><U+623C><U+3E64><U+653C><U+3E64><
U+623C><U+3E38><U+383C><U+3E34><U+653C><U+3E64>慎<U+3E30><U+
623C><U+3E64><U+653C><U+3E64><U+623C><U+3E38><U+383C> could
not be fit on page. It will not be plotted."Warning message
in wordcloud(names(result_wordcount), freq = result_wordcoun
t, scale = c(4, :
"<U+653C><U+3E64>慎<U+3E30><U+623C><U+3E64><U+653C><U+3E64><
U+623C><U+3E38><U+383C><U+3E62><U+653C><U+3E64>慎<U+3E30><U+
623C><U+3E64><U+653C><U+3E64><U+623C><U+3E38><U+383C><U+3E62
><U+653C><U+3E64>慎<U+3E30><U+623C><U+3E64><U+653C><U+3E64><
U+623C><U+3E38><U+383C><U+3E62><U+653C><U+3E64>慎<U+3E30><U+
623C><U+3E64><U+653C><U+3E64><U+623C><U+3E38><U+383C><U+3E62
><U+653C><U+3E64>慎<U+3E30><U+623C><U+3E64><U+653C><U+3E64><
U+623C><U+3E38><U+383C><U+3E62><U+653C><U+3E64>慎<U+3E30><U+
623C><U+3E64><U+653C><U+3E64><U+623C><U+3E38><U+383C><U+3E62
> could not be fit on page. It will not be plotted."Warning
```

```
message in wordcloud(names(result_wordcount), freq = result_
wordcount, scale = c(4, :
"<U+653C><U+3E64>愼<U+3E30><U+623C><U+3E64><U+653C><U+3E64><
U+623C><U+3E38>愼 could not be fit on page. It will not be p
lotted."Warning message in text.default(x1, y1, words[i], ce
x = size[i], offset = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
```

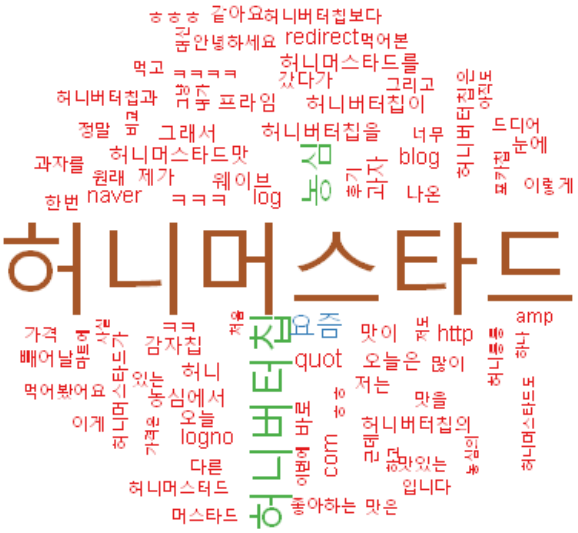
```
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
```

```
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
```

```
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
```

```
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in wordcloud(names(result_wordcount), freq = result_
wordcount, scale = c(4, :
"트러플이라는미쳣냐감자에다무슨짓이죠놈심으존나고급과자가잇는데 could
not be fit on page. It will not be plotted."Warning message
in text.default(x1, y1, words[i], cex = size[i], offset = 0,
srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
```

```
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in wordcloud(names(result_wordcount), freq = result_
wordcount, scale = c(4, :
"허니머스타드마싯서ㅜ could not be fit on page. It will not be
plotted."Warning message in text.default(x1, y1, words[i], c
ex = size[i], offset = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in strwidth(words[i], cex = size[i], ...):
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"Warning
message in text.default(x1, y1, words[i], cex = size[i], off
set = 0, srt = rotWord * :
"윈도우즈 폰트데이터베이스에서 찾을 수 없는 폰트페밀리입니다"
```



*** 수미칩 wordcloud 비교**

--네이버 블로그 : 수미칩의 연관 단어로 가장 많이 보이는 건 '허니'관련된 단어들이다. 수미칩의 히트 제품인 허니머스타드 맛 때문인 걸로 보인다. 제조업체인 '농심'이 보이지만 포카칩의 오리온 만큼 연관성이 크게 나타나진 않아 보인다.

-- 트위터 : 맛과 관련된 단어들도 보이고, 다른 과자들(포카칩, 꼬깔콘)도 보인다.

비교 :수미칩의 데이터가 포카칩이나 허니버터칩보다 더 많은 정보량을 담고 있는 것으로 보이는데, 그 이유는 아마도 수미칩이 상대적으로 최근에 나온(허니버터칩의 대항마) 감자 과자 제품이기 때문에, 인터넷을 자주 하는 세대들에게 더 어필 할 수 있었던 것으로 보인다

3- (2) Word Frequency 비교

트위터 데이터가 없는 레이즈 감자칩과 이마트 노브랜드 감자칩은 **wordcloud** 대신 **barplot**으로 어떤 단어가 가장 많이 관련되어서 나왔는지를 확인하자.

In [72]:

```
#데이터를 불러오자
dat_tmp<-read.csv('C:/Users/hanbum/Desktop/Data/Visualizatio
n Task/snack/emart_blog.csv') #네이버 블로그 데이터

# 사전을 불러오자
useSejongDic()

# tm package is based on Eng. Addition option is required
text = dat_tmp[,5]
cps = Corpus(VectorSource(text))
dtm = tm::DocumentTermMatrix(cps,
            control = list(tokenize = extractNoun,
                                      removeNumber = T,
                                      removePunctuation = T)
)

# matrix class
rmat <- as.matrix(dtm)

rmat <-spMatrix(dtm$nrow,dtm$ncol, i=dtm$i, j=dtm$j, x=dtm$v
)
wcount<-colSums(rmat)
wname <- dtm$dimnames$Terms
wname <- repair_encoding(dtm$dimnames$Terms)
colnames(rmat)<- wname

sort.var <- sort(wcount,decreasing = T)[100]
idx <- !( grepl('감자칩', wname)| (wcount<=sort.var) )
wname.rel <- wname[idx]
wcount.rel <- wcount[idx]
wcount<-sort(wcount.rel,decreasing = T)[1:20]
wname<-wname.rel[1:20]

dfcount <-data.frame(wcount,wname)
```

```
Backup was just finished!
370957 words dictionary was built.
```

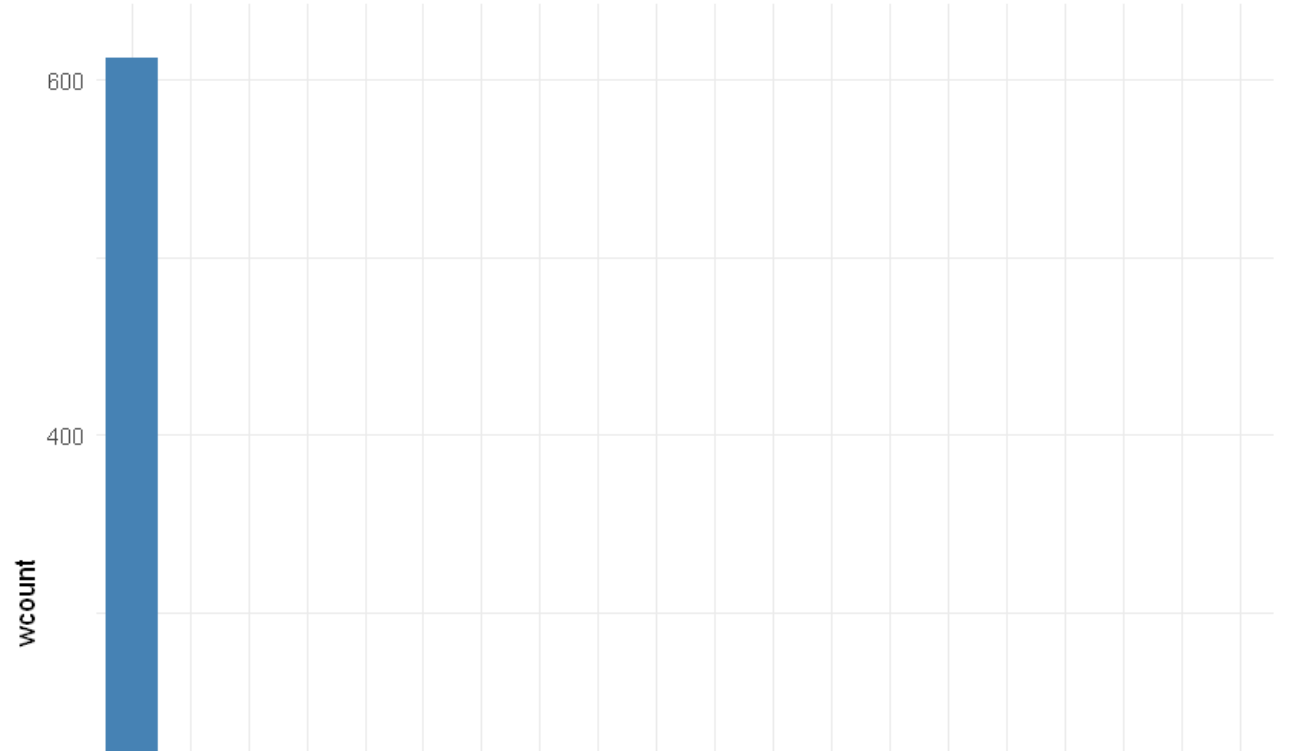
```
Best guess: EUC-KR (100% confident)
Warning message in stringi::stri_conv(x, from = from):
"the Unicode codepoint \U0000008b cannot be converted to des
tination encoding"Warning message in stringi::stri_conv(x, f
rom = from):
"the Unicode codepoint \U0000009a cannot be converted to des
```

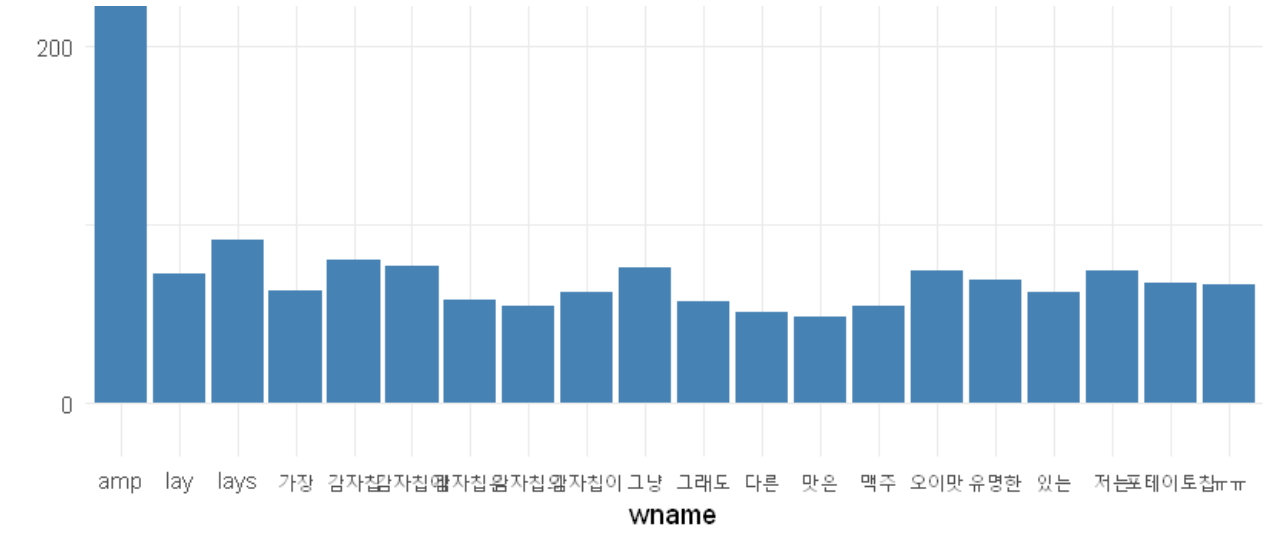
```
tination encoding"Warning message in stringi::stri_conv(x, f
rom = from):
"input data \xffffffbe in current source encoding could not
be converted to Unicode"Warning message in stringi::stri_con
v(x, from = from):
"the Unicode codepoint \U0000008b cannot be converted to des
tination encoding"Warning message in stringi::stri_conv(x, f
rom = from):
"the Unicode codepoint \U0000009a cannot be converted to des
tination encoding"Warning message in stringi::stri_conv(x, f
rom = from):
"input data \xffffffac in current source encoding could not
be converted to Unicode"Warning message in stringi::stri_con
v(x, from = from):
"the Unicode codepoint \U0000009e cannot be converted to des
tination encoding"Warning message in stringi::stri_conv(x, f
rom = from):
"input data \xffffffa5\xffffffbf in current source encoding
could not be converted to Unicode"Warning message in stringi
::stri_conv(x, from = from):
"input data \xffffffbe in current source encoding could not
be converted to Unicode"Warning message in stringi::stri_con
v(x, from = from):
"the Unicode codepoint \U00000089 cannot be converted to des
tination encoding"Warning message in stringi::stri_conv(x, f
rom = from):
"the Unicode codepoint \U0000fffd cannot be converted to des
tination encoding"Warning message in stringi::stri_conv(x, f
rom = from):
"input data \xffffffa5 in current source encoding could not
be converted to Unicode"Warning message in stringi::stri_con
v(x, from = from):
"the Unicode codepoint \U00000092 cannot be converted to des
tination encoding"Warning message in stringi::stri_conv(x, f
rom = from):
"the Unicode codepoint \U0000009f cannot be converted to des
tination encoding"Warning message in stringi::stri_conv(x, f
rom = from):
"input data \xffffffa7 in current source encoding could not
be converted to Unicode"Warning message in stringi::stri_con
v(x, from = from):
"input data \xffffffa5 in current source encoding could not
be converted to Unicode"Warning message in stringi::stri_con
v(x, from = from):
"the Unicode codepoint \U00000089 cannot be converted to des
tination encoding"Warning message in stringi::stri_conv(x, f
rom = from):
"input data \xffffffff6 in current source encoding could not
be converted to Unicode"Warning message in stringi::stri_con
v(x, from = from):
"the Unicode codepoint \U0000009f cannot be converted to des
```

```
tination encoding"Warning message in stringi::stri_conv(x, f
rom = from):
"input data \xffffffe5 in current source encoding could not
be converted to Unicode"Warning message in stringi::stri_con
v(x, from = from):
"the Unicode codepoint \U0000009f cannot be converted to des
tination encoding"Warning message in stringi::stri_conv(x, f
rom = from):
"input data \xffffffcc in current source encoding could not
be converted to Unicode"Warning message in stringi::stri_con
v(x, from = from):
"the Unicode codepoint \U00000090 cannot be converted to des
tination encoding"Warning message in stringi::stri_conv(x, f
rom = from):
"input data \xffffffa5 in current source encoding could not
be converted to Unicode"Warning message in stringi::stri_con
v(x, from = from):
"the Unicode codepoint \U00000092 cannot be converted to des
tination encoding"Warning message in stringi::stri_conv(x, f
rom = from):
"input data \xffffff8e in current source encoding could not
be converted to Unicode"Warning message in stringi::stri_con
v(x, from = from):
"input data \xffffffc2 in current source encoding could not
be converted to Unicode"
```

In [71]:

```
p<-ggplot(data=dfcount, aes(x=wname, y=wcount)) +
  geom_bar(stat="identity", fill="steelblue")+
  theme_minimal()
p
```



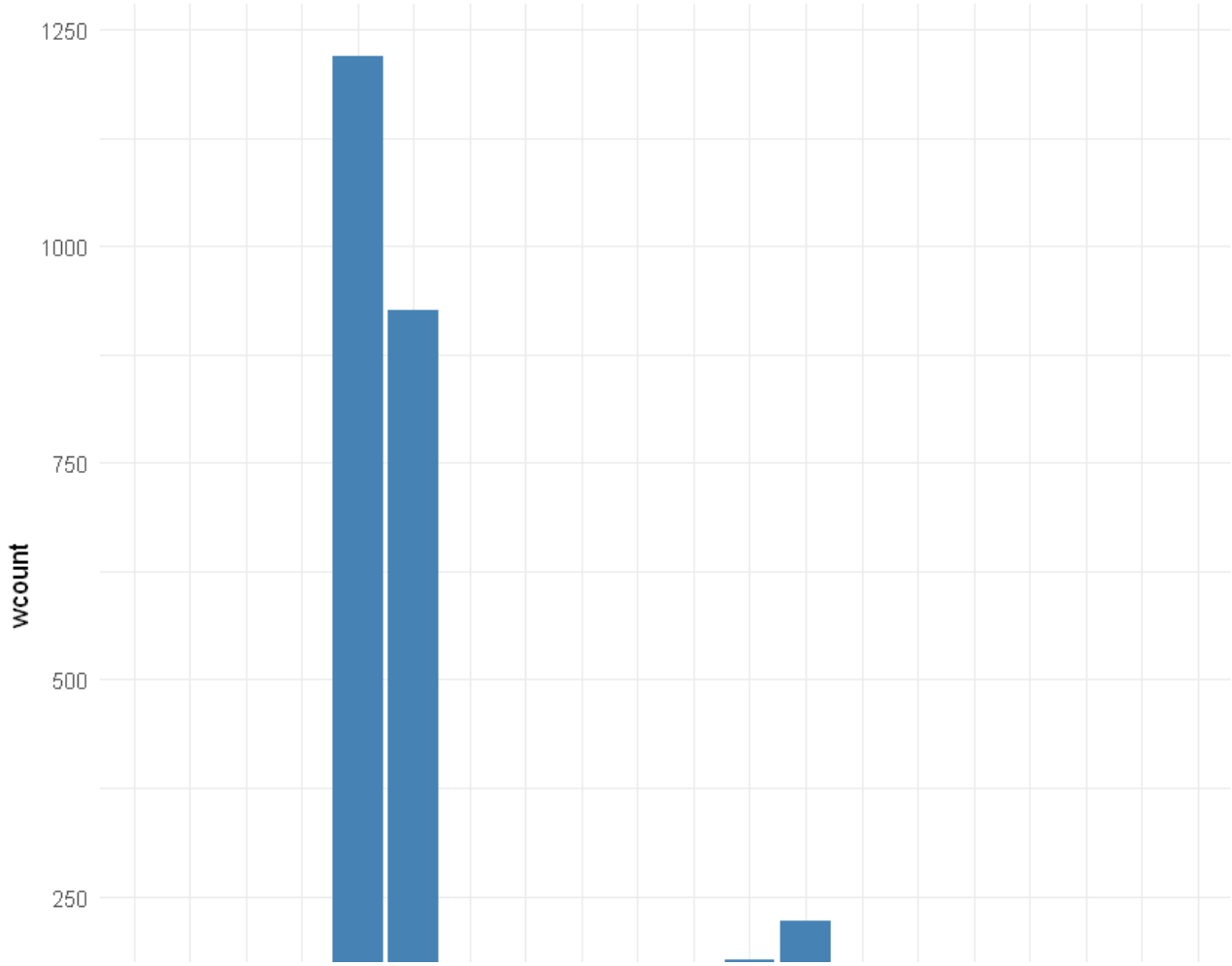


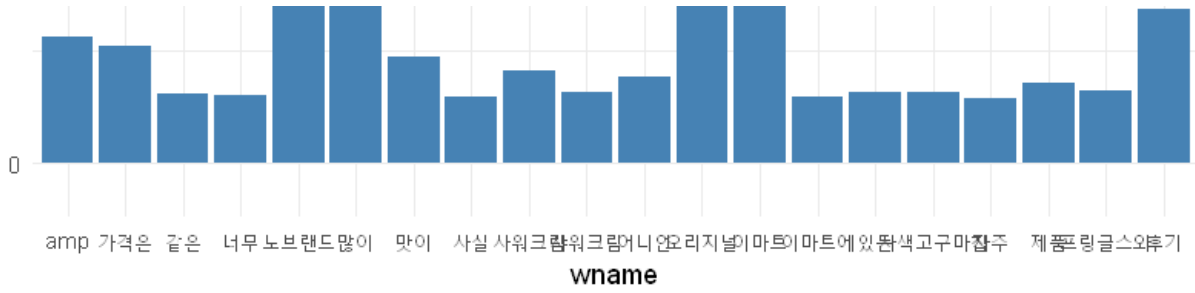
*** 레이즈 감자칩 word freqeuncy 결과**

--네이버 블로그 : amp란 단어가 가장 많이 나오는데, 이는 레이즈 감자칩과는 관련 없는 단어로 보인다. '오이맛'은 레이즈 감자칩이 갖고 있는 독특한 맛종류이고, '유명한'이란 연관단어는 해외에서 더 유명한 제품이어서 연관검색어로 나온듯하다.

In [73]:

```
p<-ggplot(data=dfcount, aes(x=wname, y=wcount)) +
  geom_bar(stat="identity", fill="steelblue")+
  theme_minimal()
p
```





* 이마트 노브랜드 감자칩 **word freqeuncy** 결과

--네이버 블로그 : '가격', '많이' 등은 노브랜드 (pb)제품의 특징을 보여주는 연관단어이다. 가성비를 중시하는 현재의 분위기를 잘 보여주는 연관단어이다.

4. Text Mining - 상관분석

4- (1) 상관관계 단어들 비교

단어들간의 상관관계를 확인해보고, 높은 순서대로 정렬해보자

In [23]:

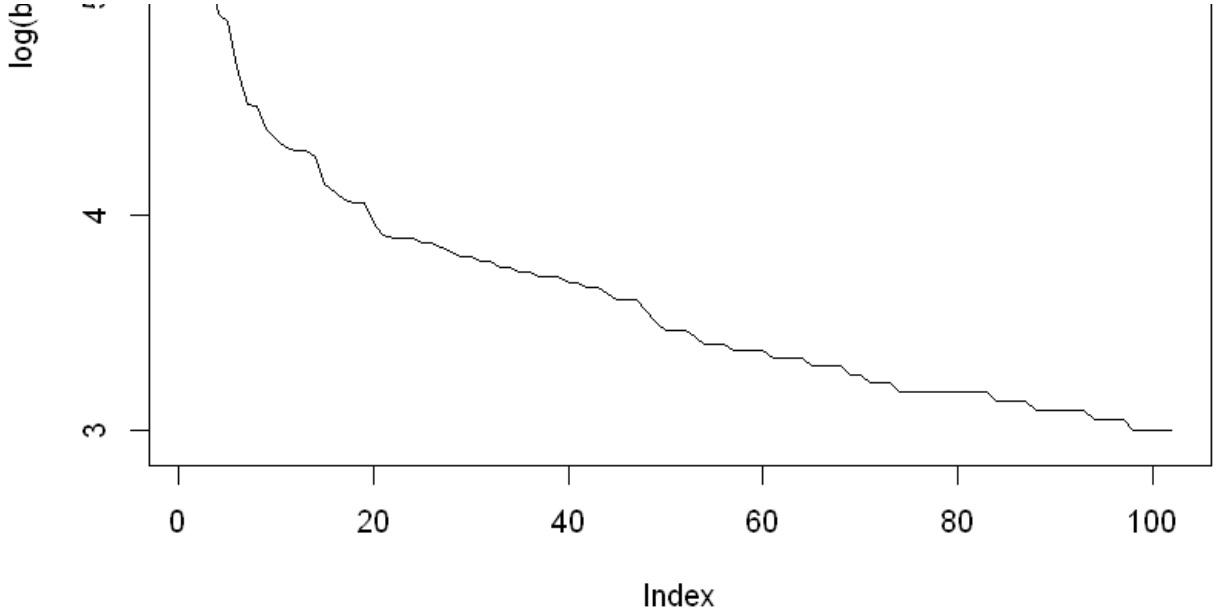
```
#데이터를 불러오자
dat_tmp<-read.csv('C:/Users/hanbum/Desktop/Data/Visualizatio
n Task/snack/sumi_blog.csv') #네이버 블로그 데이터
text = dat_tmp[,5]
cps = Corpus(VectorSource(text))
dtm = tm::DocumentTermMatrix(cps,
             control = list(tokenize = extractNoun,
                                       removeNumber = T,
                                       removePunctuation = T)
)
# matrix class
rmat <- as.matrix(dtm)

# 빈도수 확인
bb <- rmat
bb.freq <- sort(colSums(bb), decreasing = T)

# 빈도수가 많은 단어 필터링
bb.freq <- bb.freq[bb.freq>quantile(bb.freq,0.99)]
idx <- match(names(bb.freq),  colnames(bb))
bb.r <- bb[,idx]
bb.r <- as.matrix(bb.r)
cor.mat <- cor(bb.r)

# 상관관계순으로 1~10개만 추려내었다.
sort_word <- sort(cor.mat[1,], decreasing = T)[1:10]
```

```
Warning message in scan(file = file, what = what, sep = sep,
quote = quote, dec = dec, :
"따옴표로 묶인 문자열내에 EOF가 있습니다"
```

In [77]:

```
#허니버터칩 빈도수와 상관관계
plot(log(bb.freq), pch = 19, type = 'l')
sort_word
```

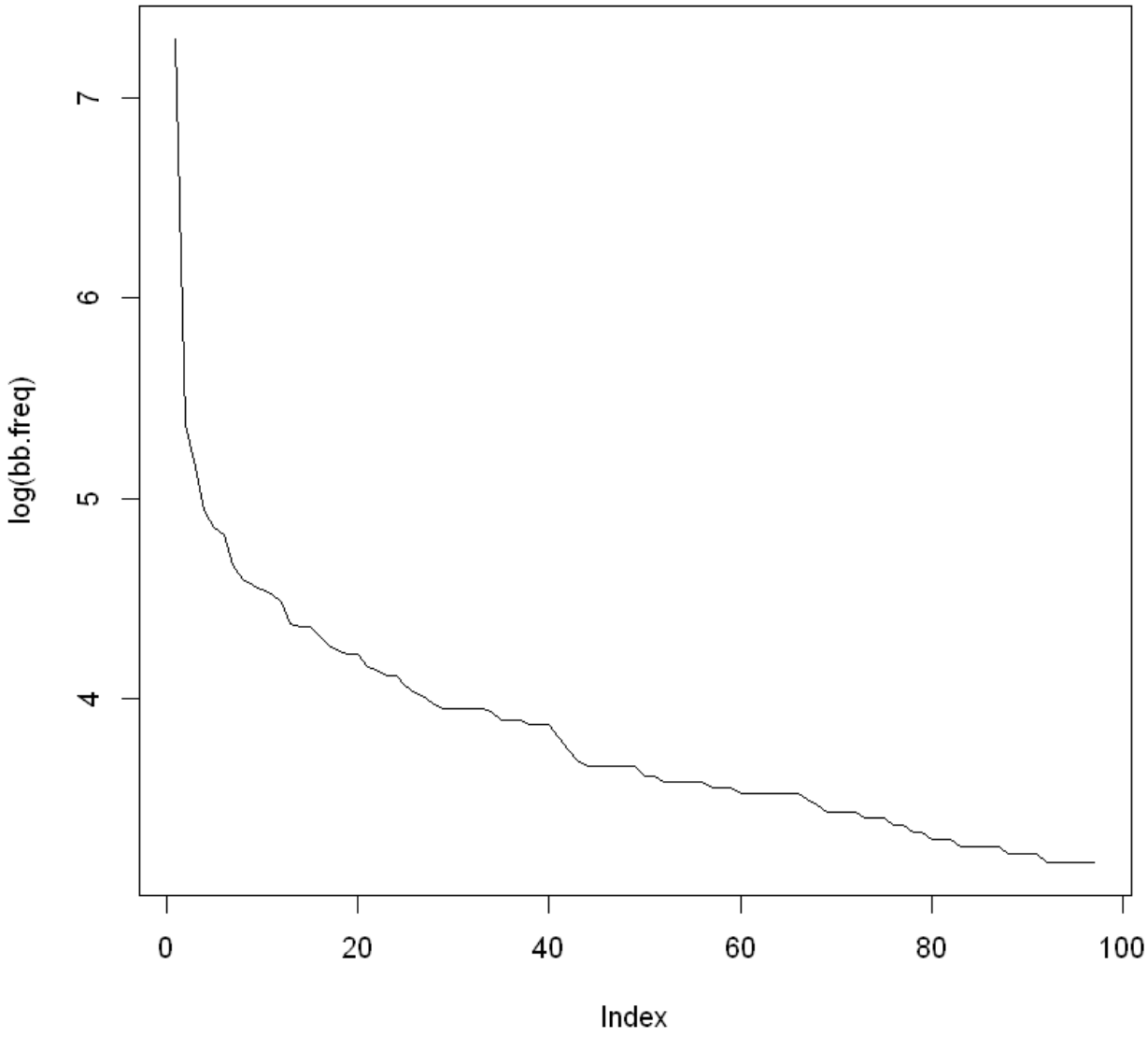
허니버터칩
1
만들기
0.298050661256059
파는곳
0.142234369723456
다들
0.103672526808325
집에서
0.098578939124114
정말
0.0810811406884686
핫한
0.0777915019778848
없어서
0.0587105802246496
오늘
0.055566681918312
요즘
0.0547436750012508



In [79]:

```
#포카칩 빈도수와 상관관계
plot(log(bb.freq), pch = 19, type = 'l')
sort_word
```

포카칩
1
스윗치즈
0.159883153756516
스윗치즈맛
0.152157392573614
불닭맛
0.13759750918275
신상
0.130119574955866
갈릭쉬림프맛
0.113584063722749
수미칩
0.11199878475608
오리온
0.111074139989042
라임페퍼맛
0.110116641749004
신제품
0.107395764837995

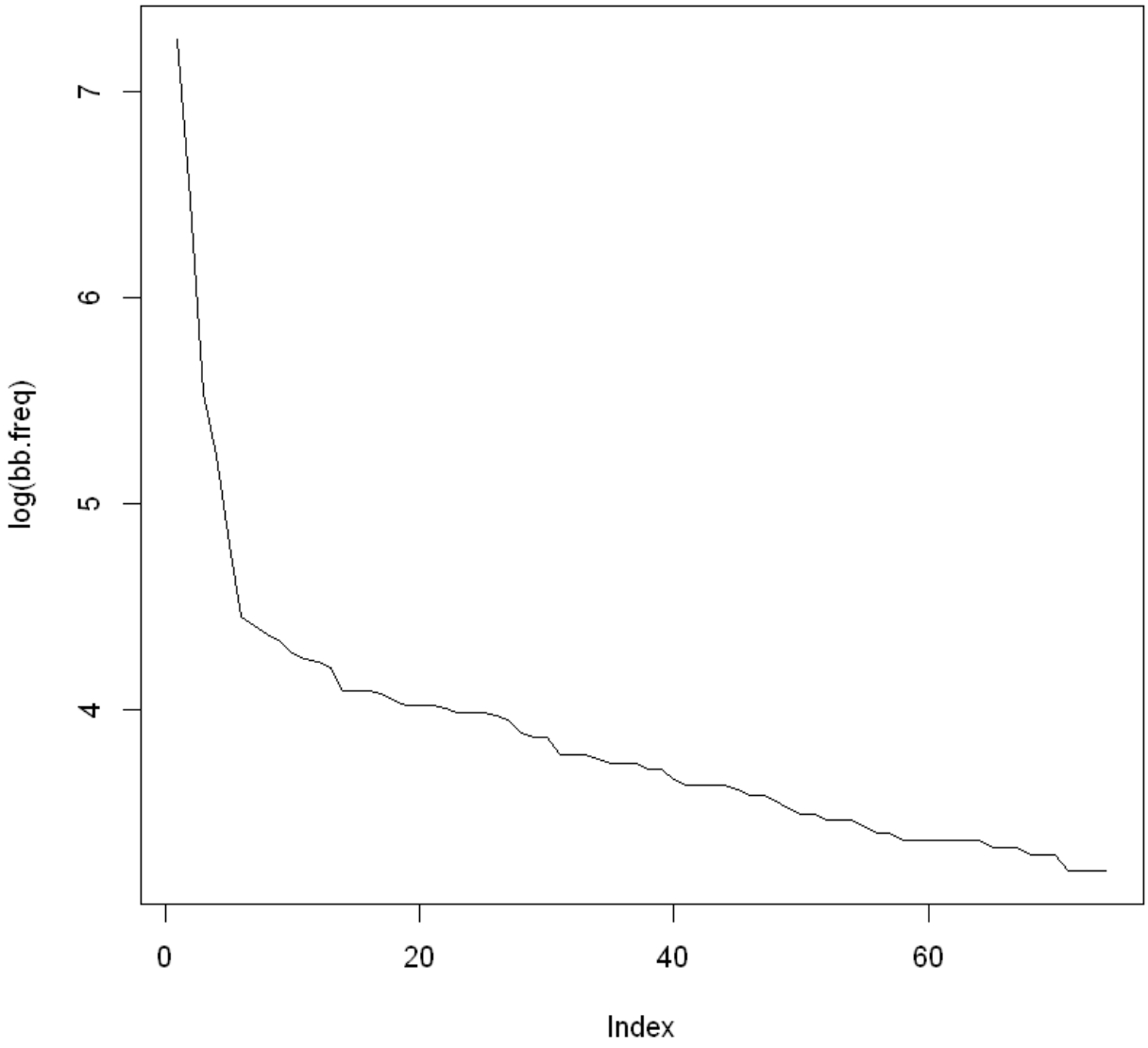


In [25]:

```
#수미칩 빈도수와 상관관계
plot(log(bb.freq), pch = 19, type = 'l')

# 수미칩과 연관있는 단어들을 상관관계 순으로 정렬하면 1~8개는 모두 ht
tp, blog, com 등 연관 없는 단어들이다.
#따라서 9등부터 보았다.

sort_word <- sort(cor.mat[1,], decreasing = T)[9:17]
sort_word
```

허니머스타드
0.574626600535823
허니버터칩
0.444823628750596
감자칩
0.355463781139011
포카칩
0.340963341567472
허니머스타드를

허니머스타드를
0.328562906264164
허니머스터드
0.294376793681439
농심
0.281354061293965
이게
0.264352496422562
정말
0.237841576195453



*** 허니버터칩, 포카칩, 수미칩의 단어들 상관관게 분석 결과**

--허니버터칩 : '만들기', '집에서'라는 단어와의 상관관계가 높은 것으로 나타났다. 집에서 만들어 먹을 수 있는 허니버터칩 레시피 같은 것이 있는 것으로 생각한다.

--포카칩 : '스윗치즈', '불닭맛', '신상'같은 단어와의 상관관계가 높은 것으로 나타났

다. 포카칩의 판매전략은 새로운 맛을 출시하여 소비자들의 관심을 끄는 것인 것 같다.

--수미칩 : '허니머스타드' 맛이 화제인 것을 알 수 있고, 포카칩과의 상관관계도 높은 것으로 보아 두 제품간에는 대체제 관계가 성립될 것 같다.

---

* 레이즈 감자칩과, 이마트 노브랜드 감자칩 데이터는 **network graph**를 그려보자

In [22]:

```
#데이터를 불러오자
dat_tmp<-read.csv('C:/Users/hanbum/Desktop/Data/Visualizatio
n Task/snack/emart_blog.csv') #네이버 블로그 데이터
text = dat_tmp[,5]
cps = Corpus(VectorSource(text))
dtm = tm::DocumentTermMatrix(cps,
             control = list(tokenize = extractNoun,
                                       removeNumber = T,
                                       removePunctuation = T)
)
# matrix class
rmat <- as.matrix(dtm)

# 빈도수 확인
bb <- rmat
bb.freq <- sort(colSums(bb), decreasing = T)

# 빈도수가 많은 단어 필터링
bb.freq <- bb.freq[bb.freq>quantile(bb.freq,0.99)]
idx <- match(names(bb.freq),  colnames(bb))
bb.r <- bb[,idx]
bb.r <- as.matrix(bb.r)
```

In [23]:

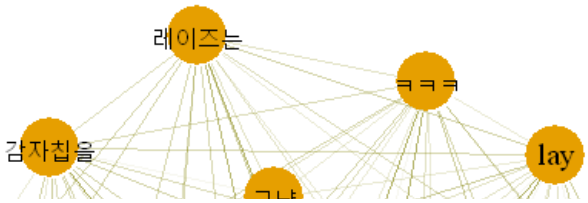
```
#bb.t matrix를 graph로 그릴 수 있게  작업이 필요하다. 우선 Boolea
n 로 바꿔 주자
bb.r <-t(bb.r)
bb.r[bb.r>=1] <- 1

# term-term adjacency matrix로 변환해보자
bb.t <- bb.r %*% t(bb.r )

#변환된 matrix를 확인해보자
head(bb.t)

#20개만 뽑아보자
bb.t<-bb.t[1:20,1:20]
```

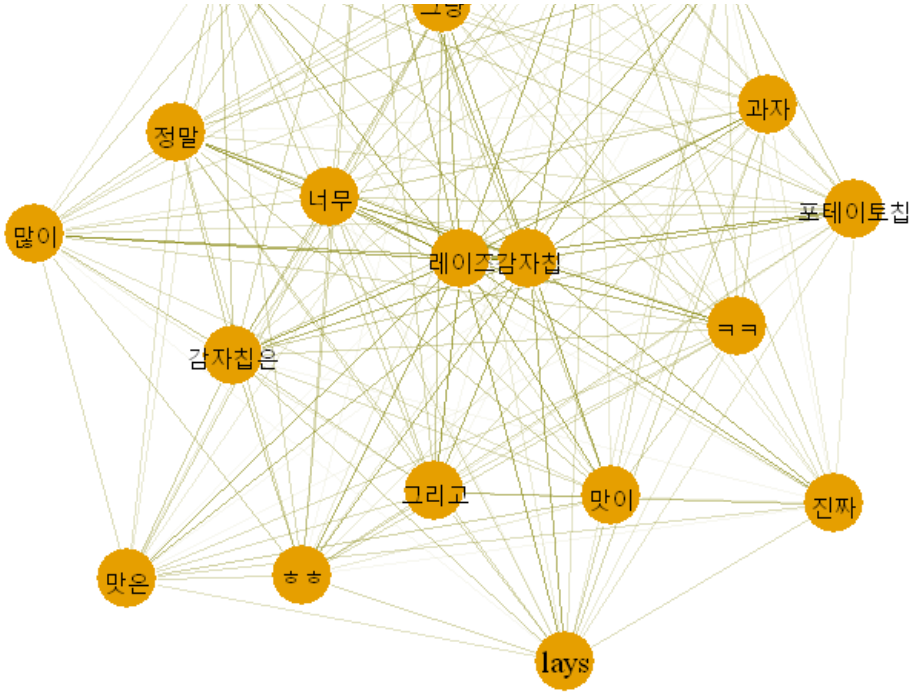
	노브랜드	이마트	감자칩	amp	어니언	오리지널	사워크림	요즘	이마트에서	과자	...	눈에	유명한	추천	자주	저렴하고	이마트노브랜드	하지만
노브랜드	847	633	590	179	151	133	111	107	101	74	...	23	23	22	24	20	9	20
이마트	633	712	499	152	122	120	83	95	38	67	...	17	17	20	21	21	7	17
감자칩	590	499	662	162	141	133	100	86	77	69	...	15	21	18	13	13	11	11
amp	179	152	162	197	143	37	111	18	19	15	...	4	5	3	4	4	3	5
어니언	151	122	141	143	166	29	92	15	15	13	...	3	3	4	2	4	3	3
오리지널	133	120	133	37	29	152	18	23	14	23	...	1	3	5	0	2	4	4

In [212]:

```
g <- graph.adjacency(bb.t, weighted=T, mode='undirected')

# loops를 없앤다
g <- simplify(g)

V(g)$label.cex <- 2.2 * V(g)$degree / max(V(g)$degree)+ .2
V(g)$label.color <- rgb(0, 0, .2, .8)
V(g)$frame.color <- NA
egam <- (log(E(g)$weight)+.4) / max(log(E(g)$weight)+.4)
E(g)$color <- rgb(.5, .5, 0, egam)
E(g)$width <- egam

plot(g, layout=layout1,vertex.label.color="black")
```

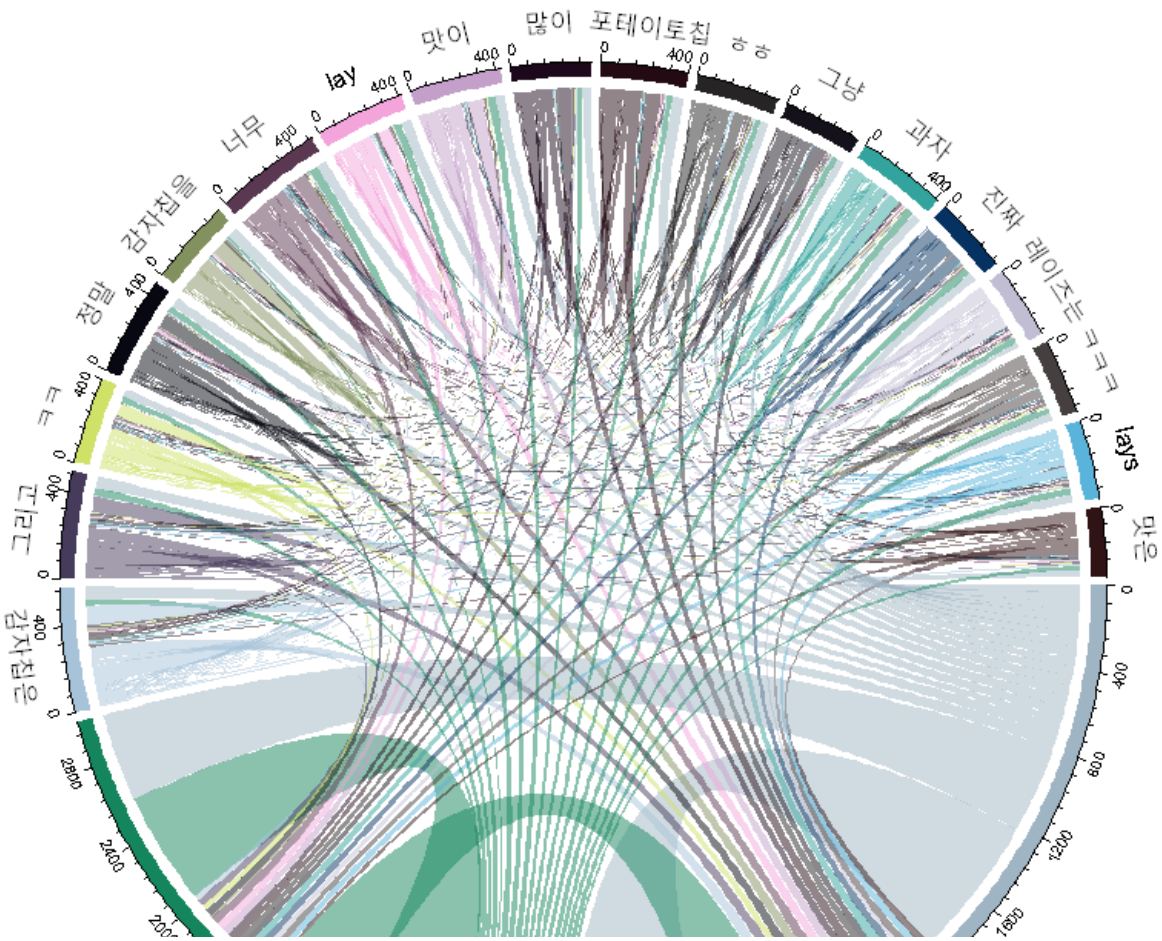
```
Warning message in max(V(g)$degree):
"max에 전달되는 인자들 중 누락이 있어 -Inf를 반환합니다"
```

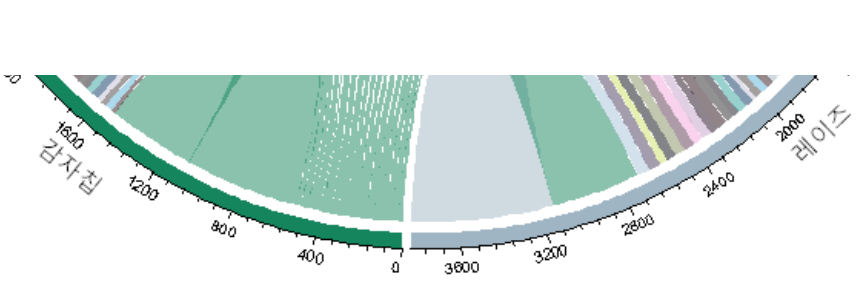




In [110]:

```
chordDiagram(bb.t, transparency = 0.5)
```



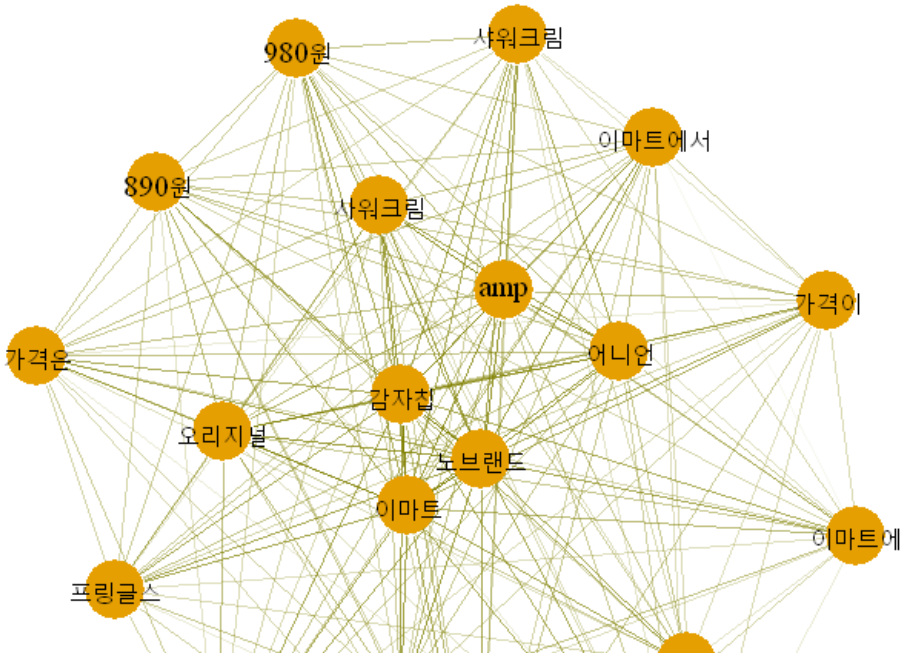


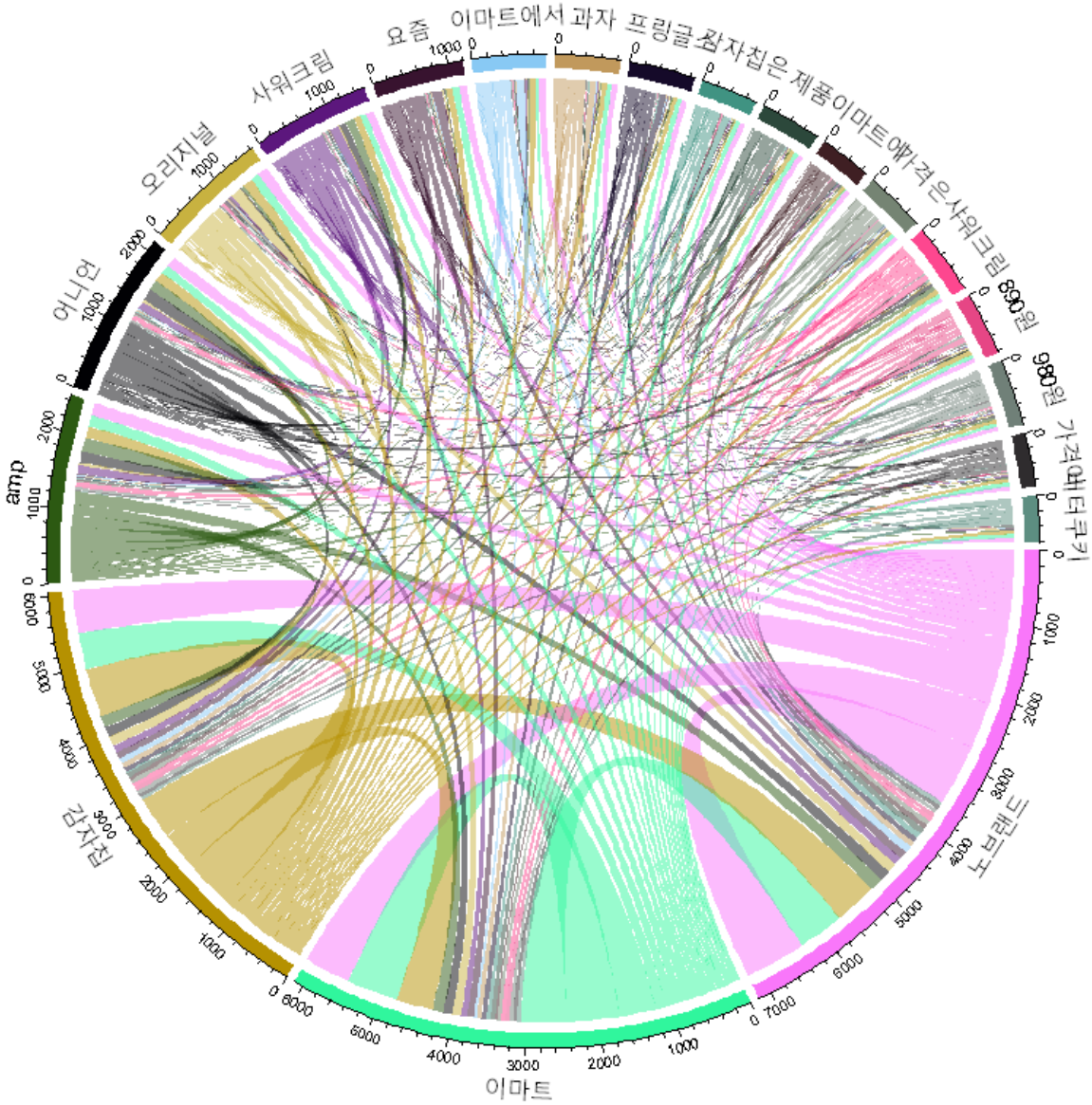
- 레이즈 감자칩 : 의미있는 결과를 해석하기가 어렵다. 그래프는 예쁘지만, 유용한 결과를 얻기엔 어려워 보인다.

In [28]:

```
g <- graph.adjacency(bb.t, weighted=T, mode='undirected')

# loops를 없앤다
g <- simplify(g)

V(g)$label.cex <- 2.2 * V(g)$degree / max(V(g)$degree)+ .2
V(g)$label.color <- rgb(0, 0, .2, .8)
V(g)$frame.color <- NA
egam <- (log(E(g)$weight)+.4) / max(log(E(g)$weight)+.4)
E(g)$color <- rgb(.5, .5, 0, egam)
E(g)$width <- egam

plot(g,vertex.label.color="black")
chordDiagram(bb.t, transparency = 0.5)
```

```
Warning message in max(V(g)$degree):
"max에 전달되는 인자들 중 누락이 있어 -Inf를 반환합니다"
```





- 이마트 노브랜드 감자칩 : 그래프에서 '가격'이란 단어의 비중이 작은 것은 예상과는 다른 결과였다.위와 마찬가지로 그래프가 예쁘지만, 해석하기는 어렵다.

4- (2) Graphical Lasso 단어 상관관계 분석

단어들간의 상관 관계를 보기 위해 추가 변수의 효과를 제어하는 동시에 두 변수간 선형 관계를 설명하는 **Partial correlation**을 활용한다.

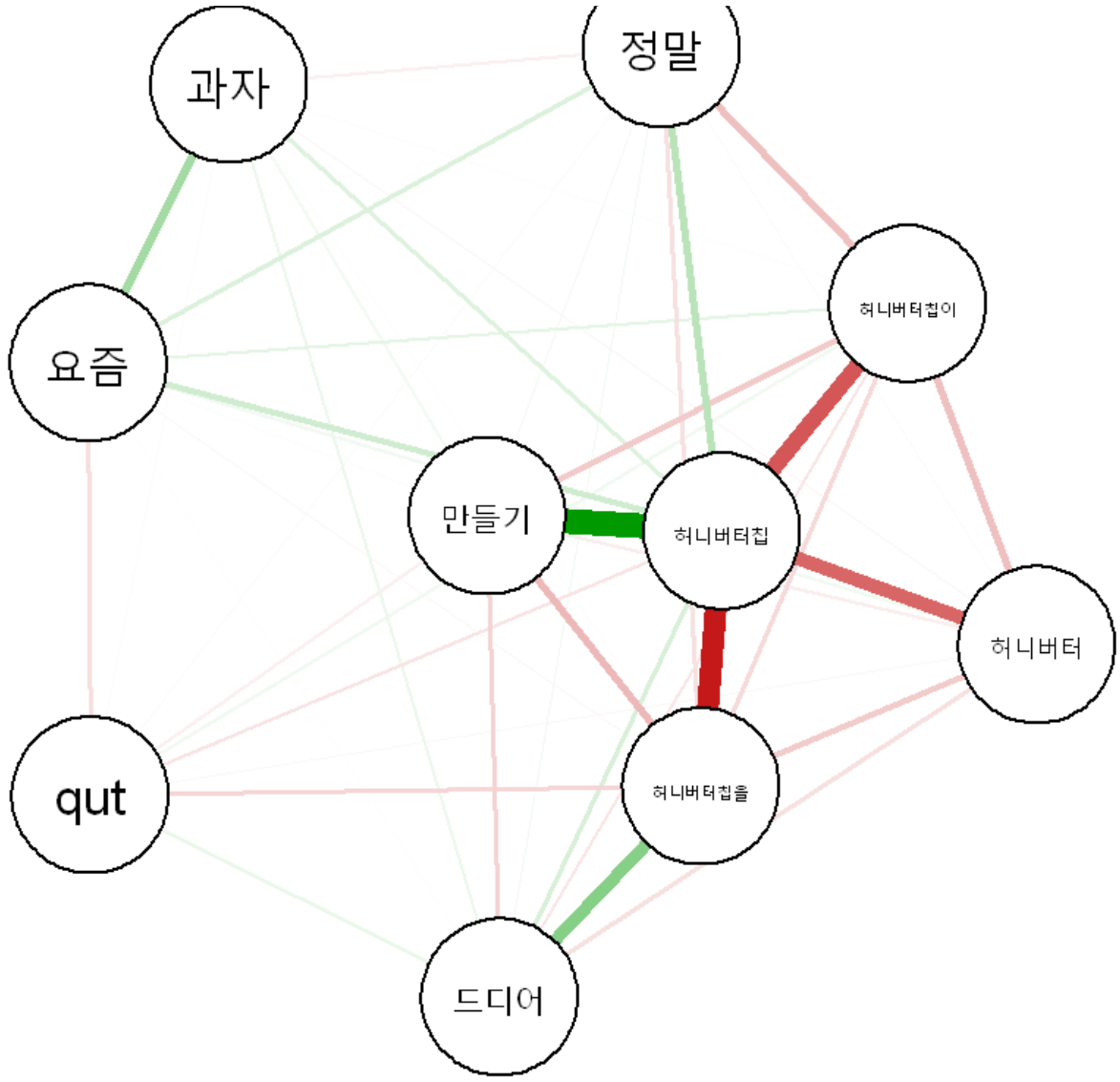
In [62]:

```
#데이터를 불러오자
dat_tmp<-read.csv('C:/Users/hanbum/Desktop/Data/Visualizatio
n Task/snack/lays_blog.csv') #네이버 블로그 데이터
text = dat_tmp[,5]
cps = Corpus(VectorSource(text))
dtm = tm::DocumentTermMatrix(cps,
             control = list(tokenize = extractNoun,
                                       removeNumber = T,
                                       removePunctuation = T)
)
# matrix class
rmat <- as.matrix(dtm)

# 빈도수 확인
bb <- rmat
bb.freq <- sort(colSums(bb), decreasing = T)

# 빈도수가 많은 단어 필터링
bb.freq <- bb.freq[bb.freq>quantile(bb.freq,0.99)]
idx <- match(names(bb.freq),  colnames(bb))
bb.r <- bb[,idx]
bb.r <- as.matrix(bb.r)

cor.mat <- cor_auto(bb.r,detectOrdinal = FALSE)
```

In [53]:

```
# 허니버터칩 glasso 그래프를 그려 보자
qgraph(cor.mat[1:10,1:10], method='glasso',
       vsize=12, maximum=.05, border.width=2,layout = "sprin
g")
```

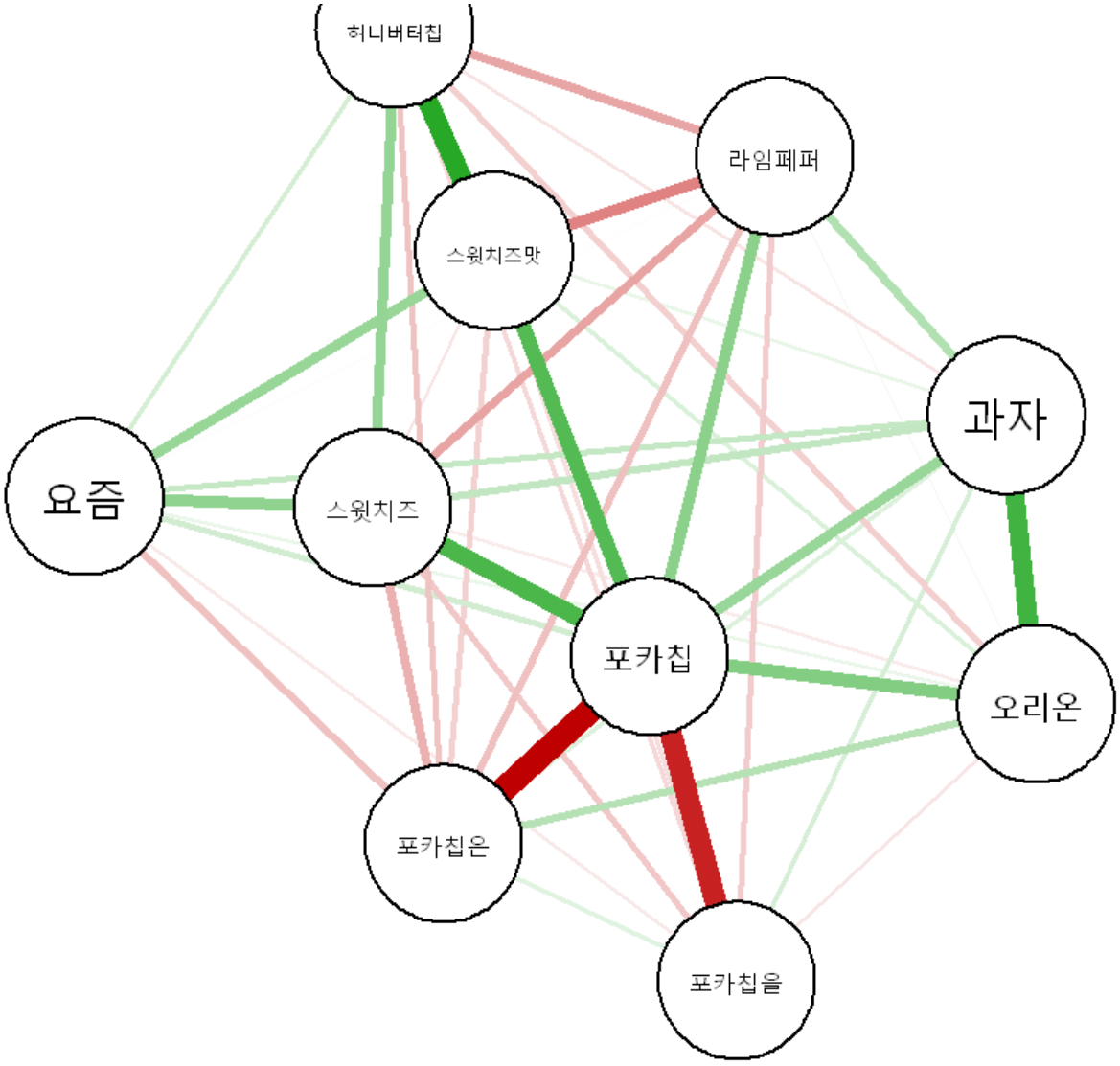
```
Warning message in qgraph(cor.mat[1:10, 1:10], method = "gla
sso", vsize = 12, maximum = 0.05, :
"The following arguments are not documented and likely not a
rguments of qgraph and thus ignored: method"Warning message
in abbreviate(colnames(input), 3):
"아스키 문자가 아닌 것을 사용한 약어입니다"Warning message in abbr
eviate(colnames(input), 3):
"아스키 문자가 아닌 것을 사용한 약어입니다"Warning message in abbr
eviate(colnames(input), 3):
"아스키 문자가 아닌 것을 사용한 약어입니다"
```



In [56]:

```
# 포카칩 glasso 그래프를 그려 보자
qgraph(cor.mat[1:10,1:10], method='glasso',
       vsize=12, maximum=.05, border.width=2,layout = "spring")
```

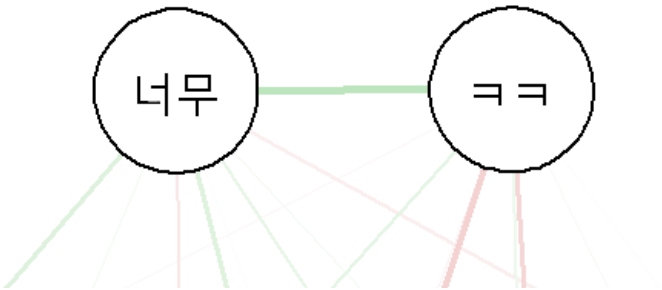
```
Warning message in qgraph(cor.mat[1:10, 1:10], method = "glasso", vsize = 12, maximum = 0.05, :
"The following arguments are not documented and likely not arguments of qgraph and thus ignored: method"Warning message in abbreviate(colnames(input), 3):
"아스키 문자가 아닌 것을 사용한 약어입니다"Warning message in abbreviate(colnames(input), 3):
"아스키 문자가 아닌 것을 사용한 약어입니다"Warning message in qgraph(cor.mat[1:10, 1:10], method = "glasso", vsize = 12, maximum = 0.05, :
"Some labels where not abbreviatable."
```

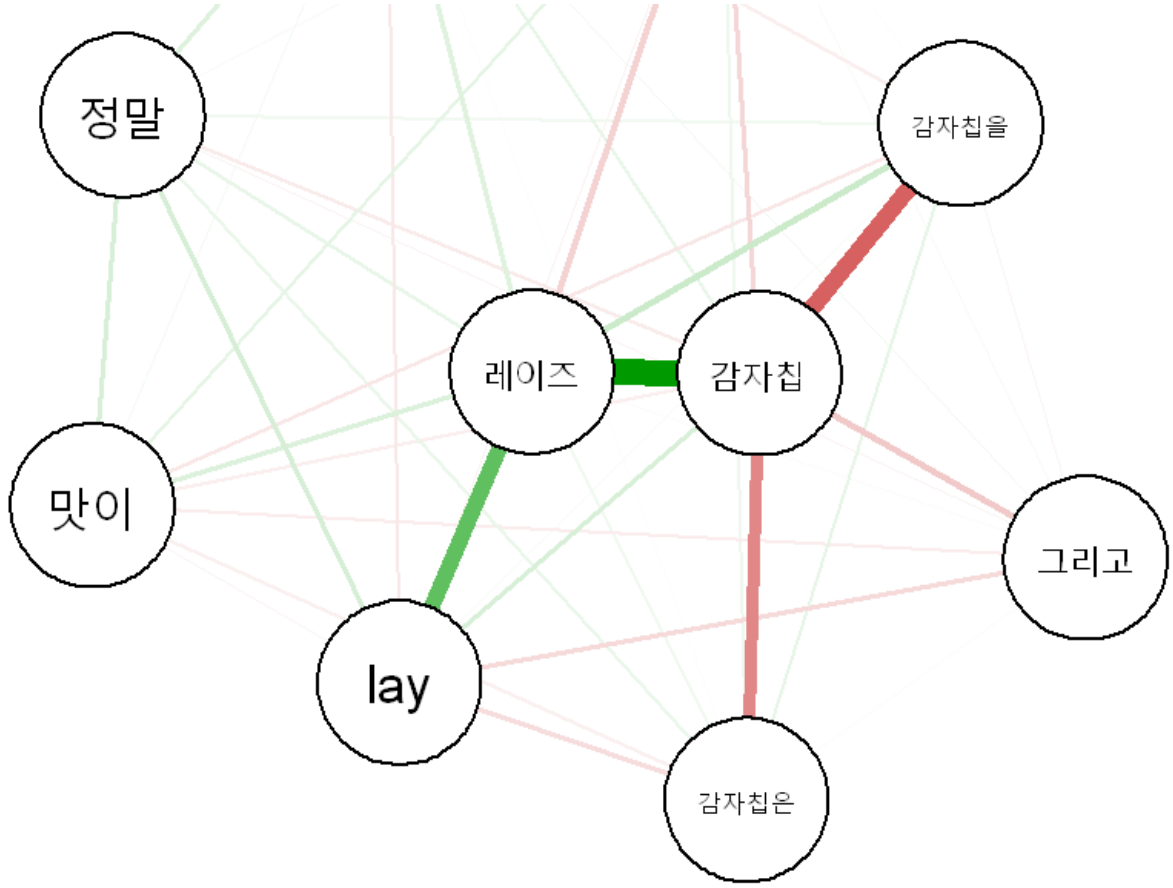


In [63]:

```
# 레이즈 감자칩 glasso 그래프를 그려 보자
qgraph(cor.mat[1:10,1:10], method='glasso',
       vsize=12, maximum=.05, border.width=2,layout = "spring")
```

```
Warning message in qgraph(cor.mat[1:10, 1:10], method = "glasso", vsize = 12, maximum = 0.05, :
"The following arguments are not documented and likely not arguments of qgraph and thus ignored: method"Warning message in abbreviate(colnames(input), 3):
"아스키 문자가 아닌 것을 사용한 약어입니다"
```

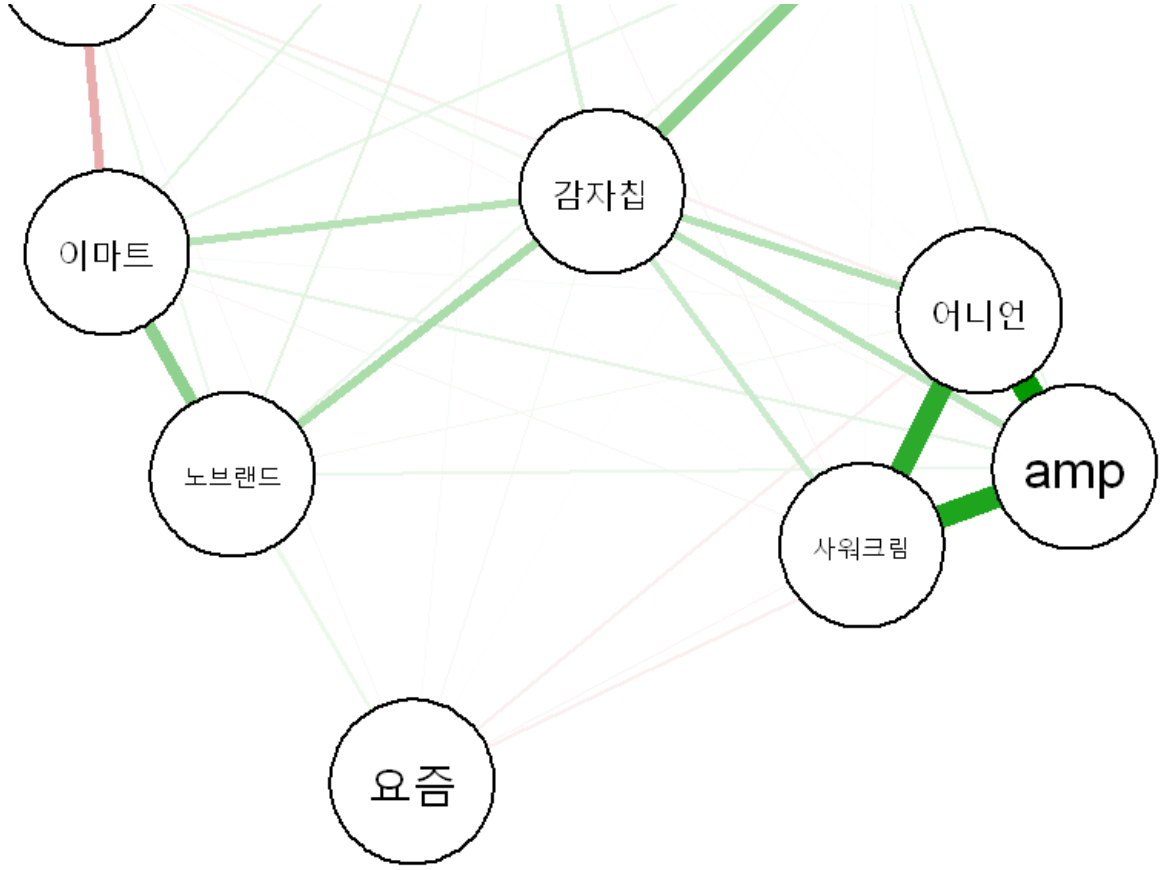




In [61]:

```
# 이마트 노브랜드 감자칩 glasso 그래프를 그려 보자
qgraph(cor.mat[1:10,1:10], method='glasso',
       vsize=12, maximum=.05, border.width=2,layout = "sprin
g")
```

```
Warning message in qgraph(cor.mat[1:10, 1:10], method = "gla
sso", vsize = 12, maximum = 0.05, :
"The following arguments are not documented and likely not a
rguments of qgraph and thus ignored: method"Warning message
in abbreviate(colnames(input), 3):
"아스키 문자가 아닌 것을 사용한 약어입니다"Warning message in abbr
eviate(colnames(input), 3):
"아스키 문자가 아닌 것을 사용한 약어입니다"Warning message in qgra
ph(cor.mat[1:10, 1:10], method = "glasso", vsize = 12, maxim
um = 0.05, :
"Some labels where not abbreviatable."
```



*** Glasso summary**

- 상관계수가 높았던 단어들이 **graphical lasso** 그래프에서도 많이 보이지만, **partial correlation**을 활용하였기에 단어들 사이의 **link**가 끊어진 것을 확인 할 수 있다

5. Text Mining - 연관규칙

*사용한 데이터는 네이버 블로그에서 모은 허니버터칩과 포카칩이다.

5- (1) 연관규칙 생성

In [32]:

```
#데이터를 불러오자
dat_tmp<-read.csv('C:/Users/hanbum/Desktop/Data/Visualization Task/snack/poca_blog.csv') #네이버 블로그 데이터
text = dat_tmp[,5]
cps = Corpus(VectorSource(text))
dtm = tm::DocumentTermMatrix(cps,
             control = list(tokenize = extractNoun,
                                       removeNumber = T,
                                       removePunctuation = T)
)
```

```
# matrix class
rmat <- as.matrix(dtm)

# 빈도수 확인
bb <- rmat
bb.freq <- sort(colSums(bb), decreasing = T)

# 빈도수가 많은 단어 필터링
bb.freq <- bb.freq[bb.freq>quantile(bb.freq,0.99)]
idx <- match(names(bb.freq),  colnames(bb))
bb.r <- bb[,idx]
bb.tt<-bb.r[1:20,1:20]
bb.df<-as.data.frame(as.table(bb.tt))

#연관규칙생성을 위한 transaction class 변환
snack.list<-split(bb.df$Terms, bb.df$Freq)
snack.transaction<-as(snack.list, "transactions")
snack.transaction
```

```
Warning message in asMethod(object):
"removing duplicated items in transactions"
```

```
transactions in sparse format with
 5 transactions (rows) and
 20 items (columns)
```

In [30]:

```
#허니버터칩 연관규칙 생성
rules = apriori(snack.transaction, parameter=list(support=0.
001, confidence=0.6))
summary(rules)
```

```
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime s
upport minlen
        0.6    0.1    1 none FALSE            TRUE       5
0.001      1
 maxlen target   ext
     10  rules FALSE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 0

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[20 item(s), 4 transaction(s)] done [0.0
0s].
sorting and recoding items ... [20 item(s)] done [0.00s].
```

```
sorting and recoding items ... [20 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 5 6 7 8 9 10
```

```
Warning message in apriori(snack.transaction, parameter = list(support = 0.001, :
"Mining stopped (maxlen reached). Only patterns up to a length of 10 returned!"
```

```
 done [0.65s].
writing ... [5188164 rule(s)] done [2.92s].
creating S4 object  ... done [7.33s].
```

```
set of 5188164 rules

rule length distribution (lhs + rhs):sizes
      1       2       3       4       5       6       7
8       9      10
      4     280    2874   17378   72515  223551  530628  995
748 1502631 1842555

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   8.000   9.000   8.753  10.000  10.000

summary of quality measures:
    support         confidence          lift             coun
t
 Min.   :0.2500   Min.   :0.6667   Min.   :0.8889   Min.   :
1.00
 1st Qu.:0.2500   1st Qu.:1.0000   1st Qu.:2.0000   1st Qu.:
1.00
 Median :0.2500   Median :1.0000   Median :2.0000   Median :
1.00
 Mean   :0.2525   Mean   :1.0000   Mean   :2.5347   Mean   :
1.01
 3rd Qu.:0.2500   3rd Qu.:1.0000   3rd Qu.:4.0000   3rd Qu.:
1.00
 Max.   :1.0000   Max.   :1.0000   Max.   :4.0000   Max.   :
4.00

mining info:
              data ntransactions support confidence
 snack.transaction             4   0.001        0.6
```

In [34]:

```
#포카칩 연관규칙 생성
rules = apriori(snack.transaction, parameter=list(support=0.001, confidence=0.6))
summary(rules)
```

```
Apriori

Parameter specification:
```

```
Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime s
upport minlen
        0.6    0.1    1 none FALSE            TRUE       5
0.001      1
 maxlen target   ext
     10  rules FALSE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 0

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[20 item(s), 5 transaction(s)] done [0.0
0s].
sorting and recoding items ... [20 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 5 6 7 8 9 10
```

```
Warning message in apriori(snack.transaction, parameter = li
st(support = 0.001, :
"Mining stopped (maxlen reached). Only patterns up to a leng
th of 10 returned!"
```

```
 done [0.59s].
writing ... [5239676 rule(s)] done [2.58s].
creating S4 object  ... done [7.49s].
```

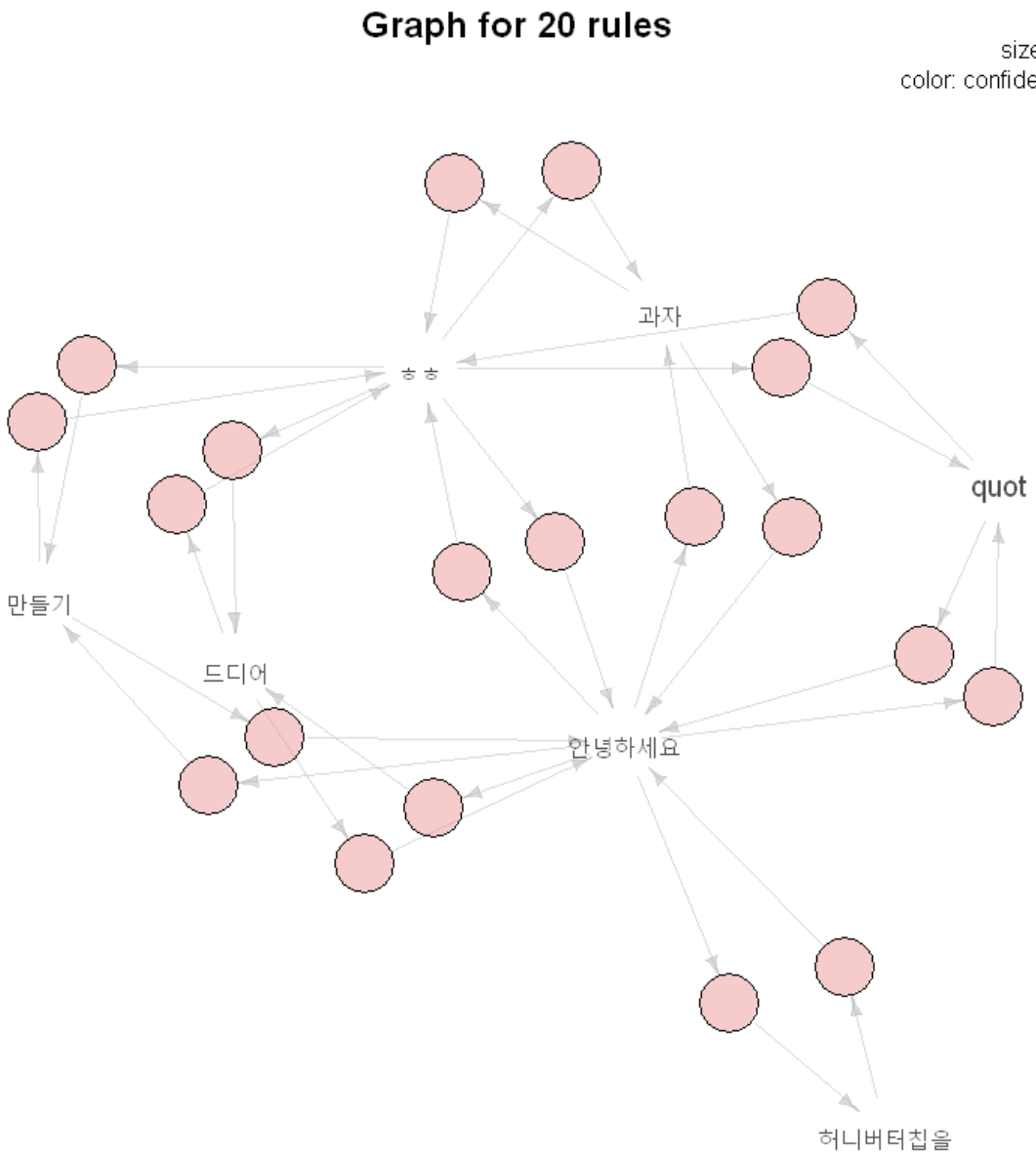
```
set of 5239676 rules

rule length distribution (lhs + rhs):sizes
      1       2       3       4       5       6       7
8       9      10
      3     265    3039   18661   76664  231888  542304 1007
664 1511628 1847560

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1.00    8.00    9.00    8.74   10.00   10.00

summary of quality measures:
    support         confidence        lift            count
 Min.   :0.2   Min.   :0.6   Min.   :1.000   Min.   :1
 1st Qu.:0.2   1st Qu.:1.0   1st Qu.:2.500   1st Qu.:1
 Median :0.2   Median :1.0   Median :5.000   Median :1
 Mean   :0.2   Mean   :1.0   Mean   :3.716   Mean   :1
 3rd Qu.:0.2   3rd Qu.:1.0   3rd Qu.:5.000   3rd Qu.:1
 Max.   :1.0   Max.   :1.0   Max.   :5.000   Max.   :5

mining info:
              data ntransactions support confidence
 snack.transaction             5   0.001        0.6
```

5- (2) 연관규칙 시각화

- 위에서 구한 연관규칙을 시각화하여 표현해보자

In [9]:

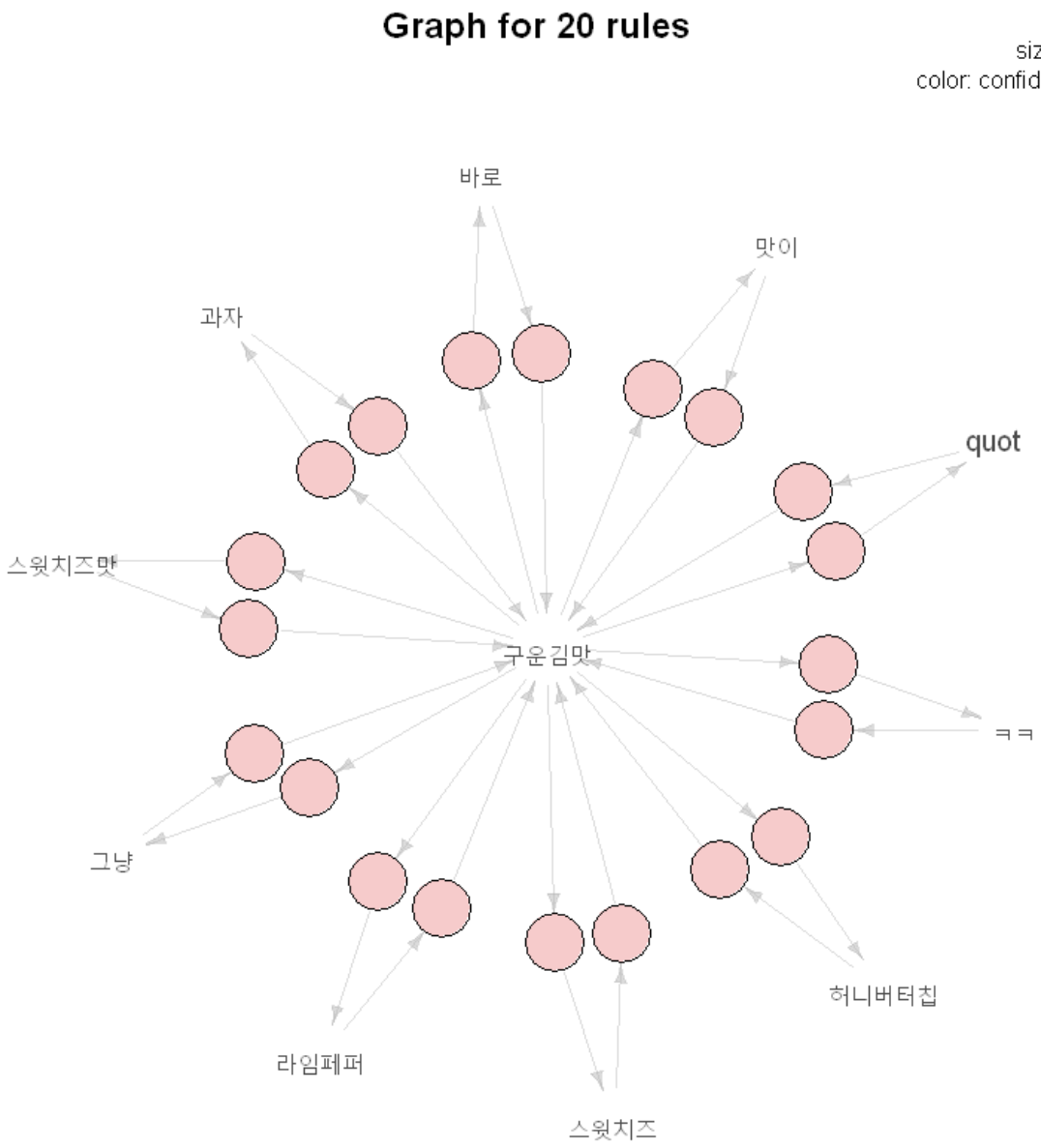
```
# 허니버터칩 연관규칙 시각화
subrules <- head(sort(rules, by="lift"), 20)
plot(subrules, method="graph",measure = 'lift',shading = 'co
nfidence')
```



- 허니버터칩 연관규칙 : 과자와 관련된 단어보다 블로그에서 흔히 볼 수 있는 인사말이 많이 나온것으로 보아 단어 정제 과정이 더 필요했던 것 같다.

In [35]:

```
# 포카칩 연관규칙 시각화
subrules <- head(sort(rules, by="lift"), 20)
plot(subrules, method="graph",measure = 'lift',shading = 'co
nfidence')
```



- 포카칩 연관규칙 : 맛을 중심으로 관계가 형성되어 있다.

6. Time Series Analysis

- 검색어를 바탕으로 시계열 분석을 해보자
- 사용한 데이터는 네이버 블로그에서 모은 허니버터칩, 이마트 노브랜드 감자칩이다.

6- (1) 일일 검색어 분석

In [19]:

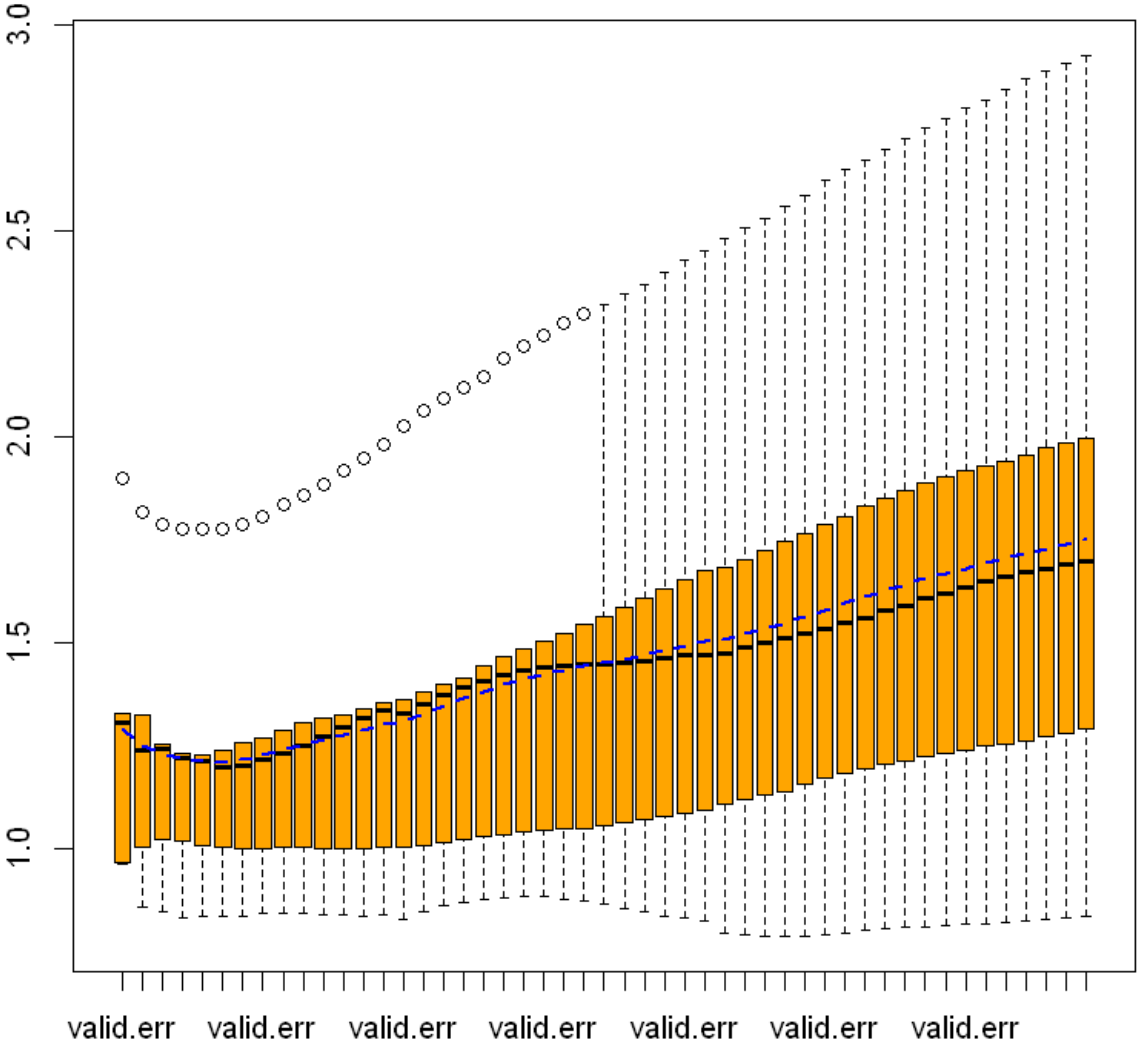
```
#데이터를 불러오자
dat_tmp<-read.csv('C:/Users/hanbum/Desktop/Data/Visualizatio
n Task/snack/emart_blog.csv') #네이버 블로그 데이터
text = dat_tmp[,5]
cps = Corpus(VectorSource(text))
dtm = tm::DocumentTermMatrix(cps,
            control = list(tokenize = extractNoun,
                                      removeNumber = T,
                                      removePunctuation = T)
)
# matrix class
rmat <- as.matrix(dtm)

# 빈도수 확인
bb <- rmat
bb.freq <- sort(colSums(bb), decreasing = T)

# 빈도수가 많은 단어 필터링
bb.freq <- bb.freq[bb.freq>quantile(bb.freq,0.99)]
idx <- match(names(bb.freq),  colnames(bb))
bb.r <- bb[,idx]

# top posting blogger and link
tb <- table(dat_tmp[,2])
top.blogger<- sort(tb, decreasing = T)[1:4]
tmp <- dat_tmp %>% select(bloggername, link) %>%
filter(bloggername %in% names(top.blogger) )

#post date Analysis(문자열을 날짜로 변환하기 위한 작업)
tb <- table(dat_tmp$postdate)
x <-as.Date(names(tb), format = "%Y%m%d")
y <- as.numeric(tb)

xx <- as.Date(as.integer(min(x)):as.integer(max(x)),
        origin = "1970-01-01")
yy <- rep(0, length(xx))
yy[xx%in%x] <-y

xint <- as.integer(xx)
rdata = data.frame(y = yy, x = xint)
fit<-loess(y~x,data = rdata, span = 0.1, normalize = FALSE)

# 모형 복잡도 선택
k.fold = 5
set.seed(1)
idx <-sample(1:k.fold, length(xint), replace = TRUE)
k = 1
rdata.tr <- rdata[idx != k, ]
rdata.va <- rdata[idx == k, ]
```

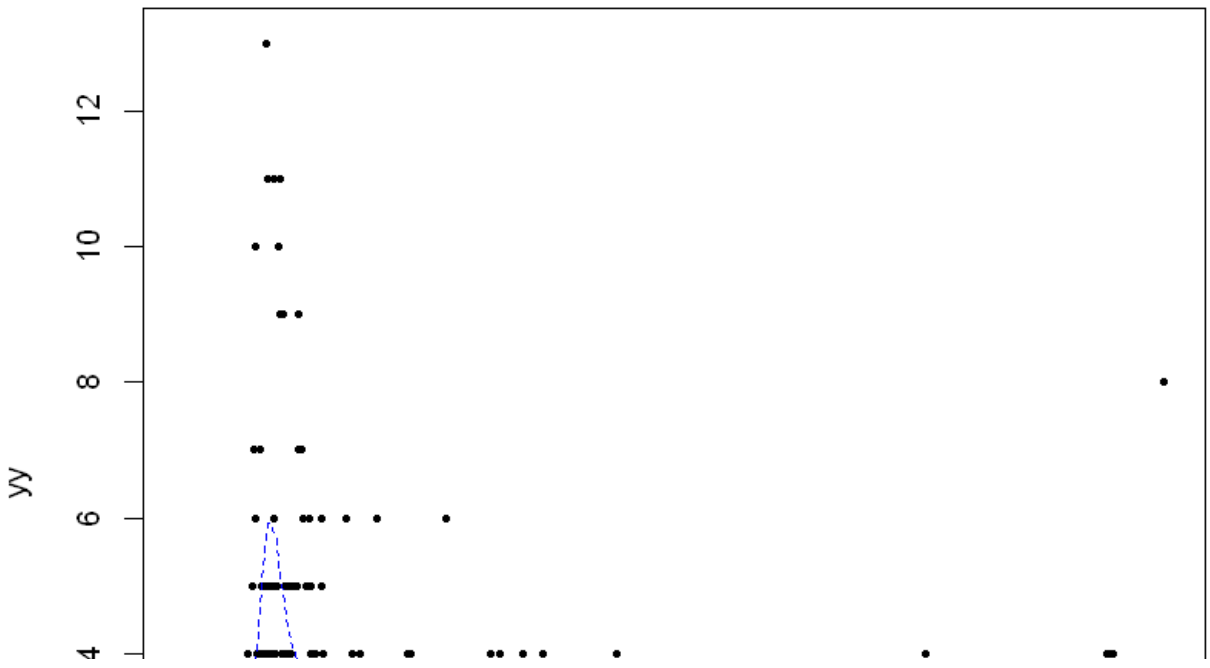
```
valid.err <- c()
span.var <- seq(0.02, 0.5, by  = 0.01)
valid.mat <- NULL

for (j in 1:length(span.var))
{
  valid.err <- c()
  for (k in 1:k.fold)
  {
    rdata.tr <- rdata[idx != k, ]
    rdata.va <- rdata[idx == k, ]
    fit<-loess(y~x,data = rdata.tr,
               span = span.var[j], normalize = FALSE)
    fit.y<-predict(fit, newdata = rdata.va)
    valid.err[k] <- mean((fit.y-rdata.va$y)^2, na.rm = T)
 }
      valid.mat <- cbind(valid.mat, valid.err)
}

# model decision
span.par<- span.var[which.min(colMeans(valid.mat))]
fit<-loess(y~x,data = rdata,
           span = span.par, normalize = FALSE)
```

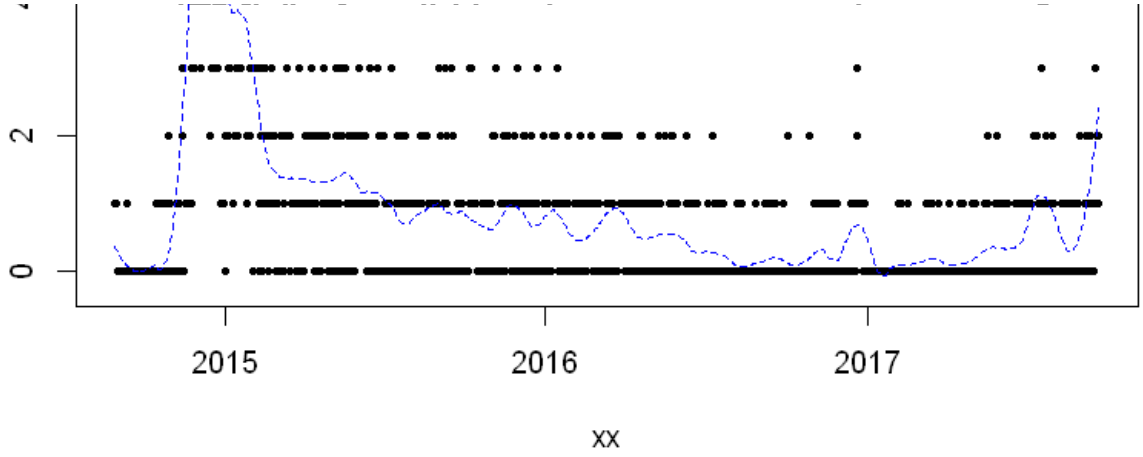
```
Warning message in simpleLoess(y, x, w, span, degree = degre
e, parametric = parametric, :
"k-d tree limited by memory. ncmax= 659"Warning message in s
impleLoess(y, x, w, span, degree = degree, parametric = para
metric, :
"k-d tree limited by memory. ncmax= 646"Warning message in s
impleLoess(y, x, w, span, degree = degree, parametric = para
metric, :
"k-d tree limited by memory. ncmax= 646"Warning message in s
impleLoess(y, x, w, span, degree = degree, parametric = para
metric, :
"k-d tree limited by memory. ncmax= 653"Warning message in s
impleLoess(y, x, w, span, degree = degree, parametric = para
metric, :
"k-d tree limited by memory. ncmax= 656"
```

In [3]:

```
#허니버터칩 검색어
boxplot(valid.mat, col='orange')
lines(colMeans(valid.mat), col = "blue", lty = 2, lwd = 2)

plot(xx,yy,  pch = 19, cex = 0.5)
points(xx,fit$fitted, type = 'l', lty = 2, lwd = 1.5, col =
'blue')
```

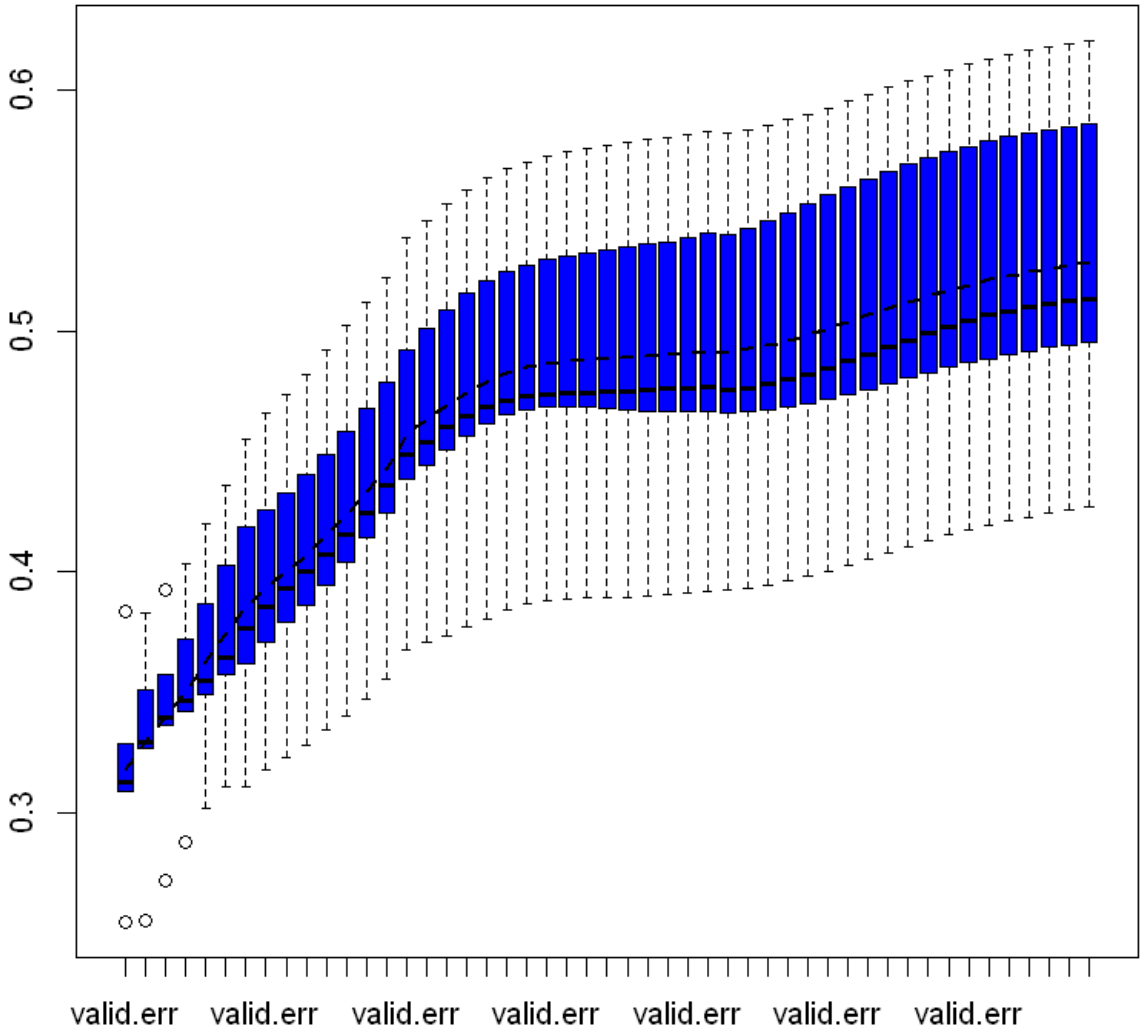

3.0
2.5
2.0
1.5
1.0
valid.err
valid.err
valid.err
valid.err
valid.err
valid.err
valid.err

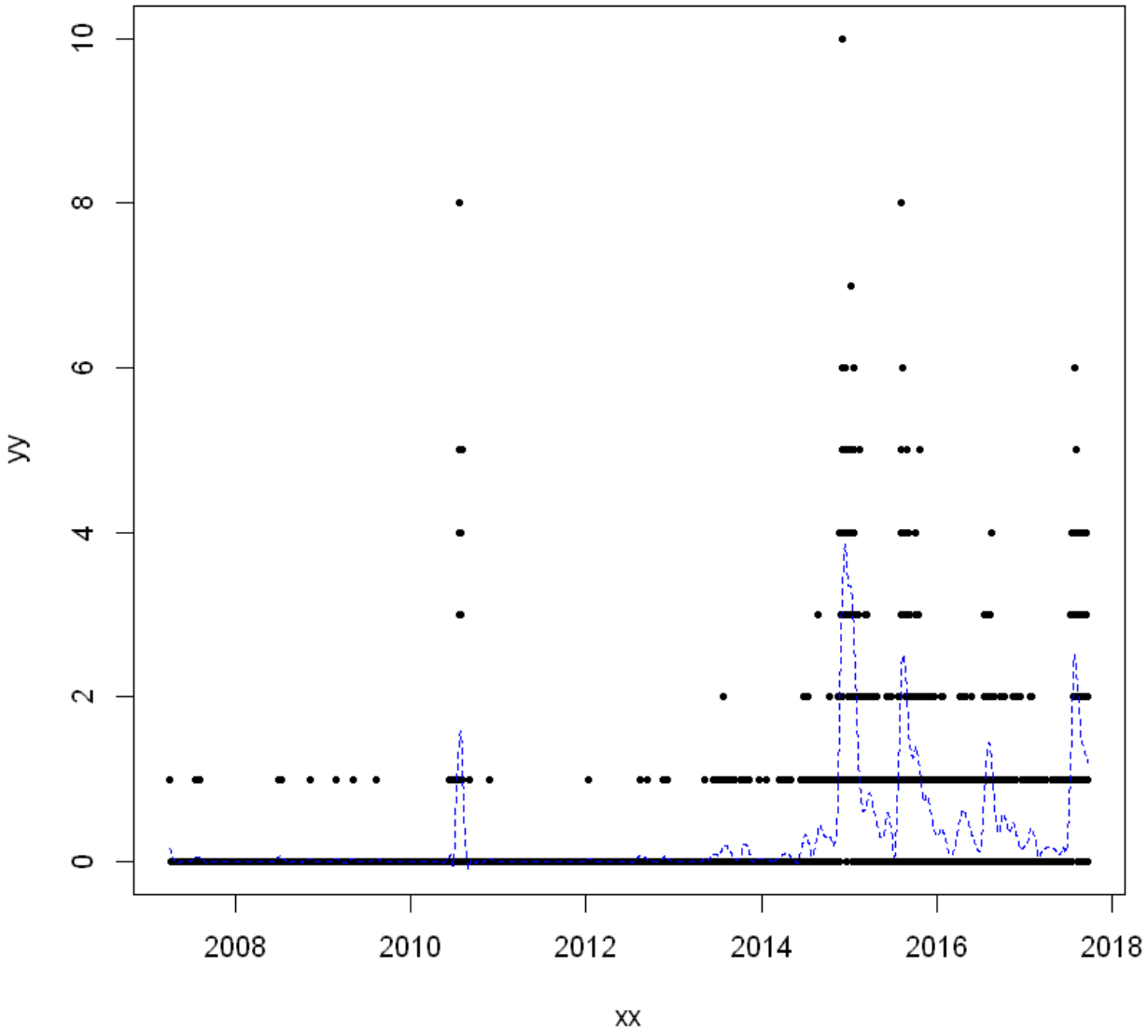

yy
12
10
8
6
4



In [54]:

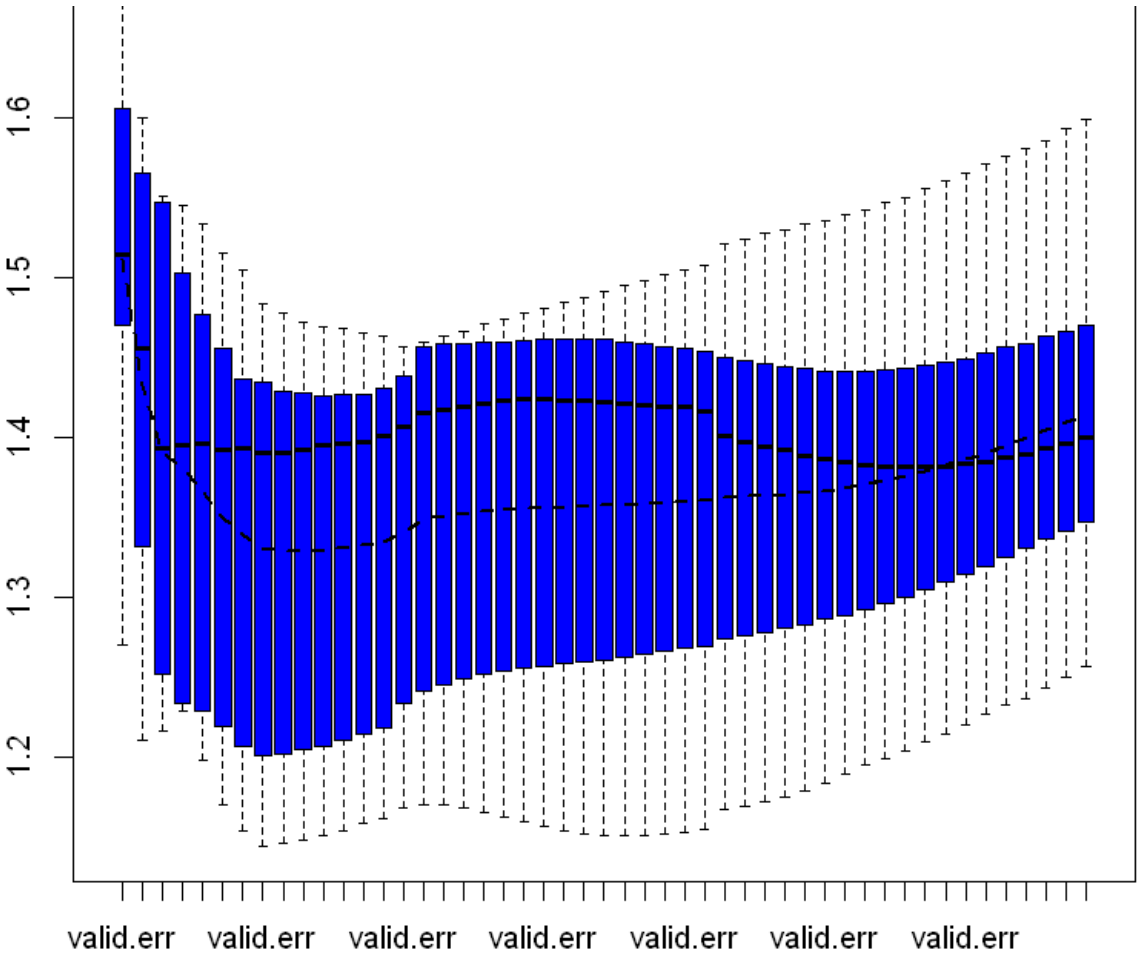
```
#포카칩 검색어
boxplot(valid.mat, col='blue')
lines(colMeans(valid.mat), col = "black", lty = 2, lwd = 2)

plot(xx,yy,  pch = 19, cex = 0.5)
points(xx,fit$fitted, type = 'l', lty = 2, lwd = 1.5, col =
'blue')
```

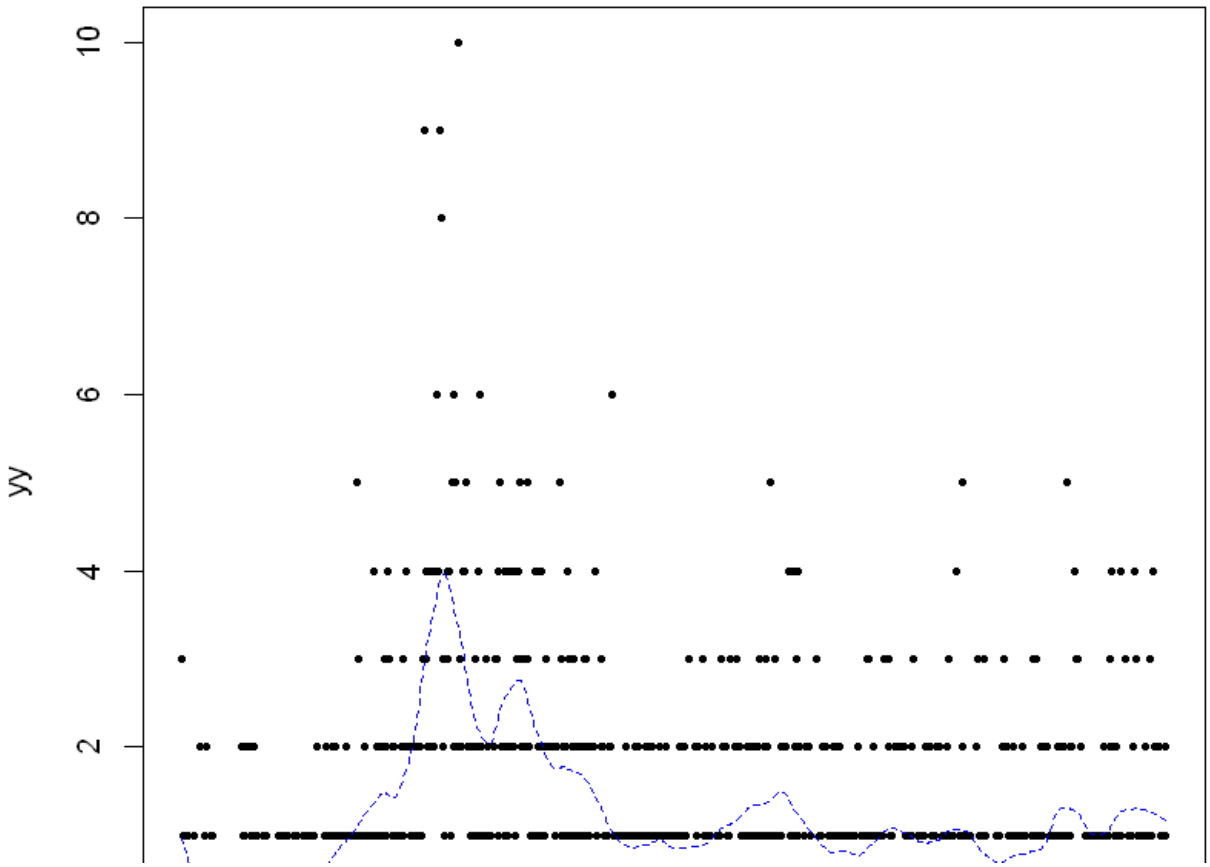


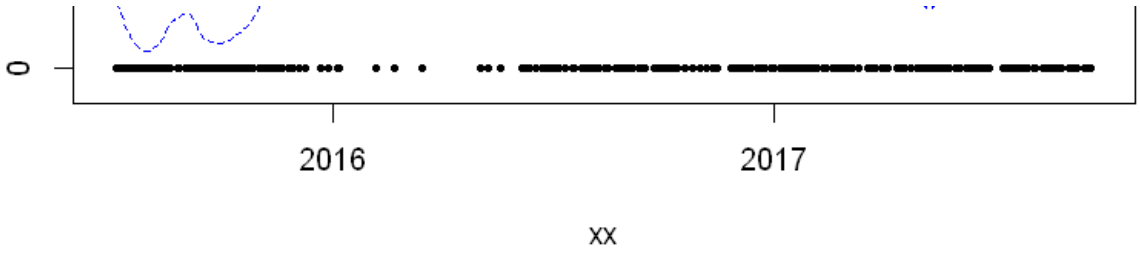


In [56]:

```
#이마트 노브랜드 감자칩 검색어
boxplot(valid.mat, col='blue')
lines(colMeans(valid.mat), col = "black", lty = 2, lwd = 2)

plot(xx,yy,  pch = 19, cex = 0.5)
points(xx,fit$fitted, type = 'l', lty = 2, lwd = 1.5, col =
'blue')
```


1.2
1.3
1.4
1.5
1.6
valid.err
valid.err
valid.err
valid.err
valid.err
valid.err
valid.err


yy
2
4
6
8
10

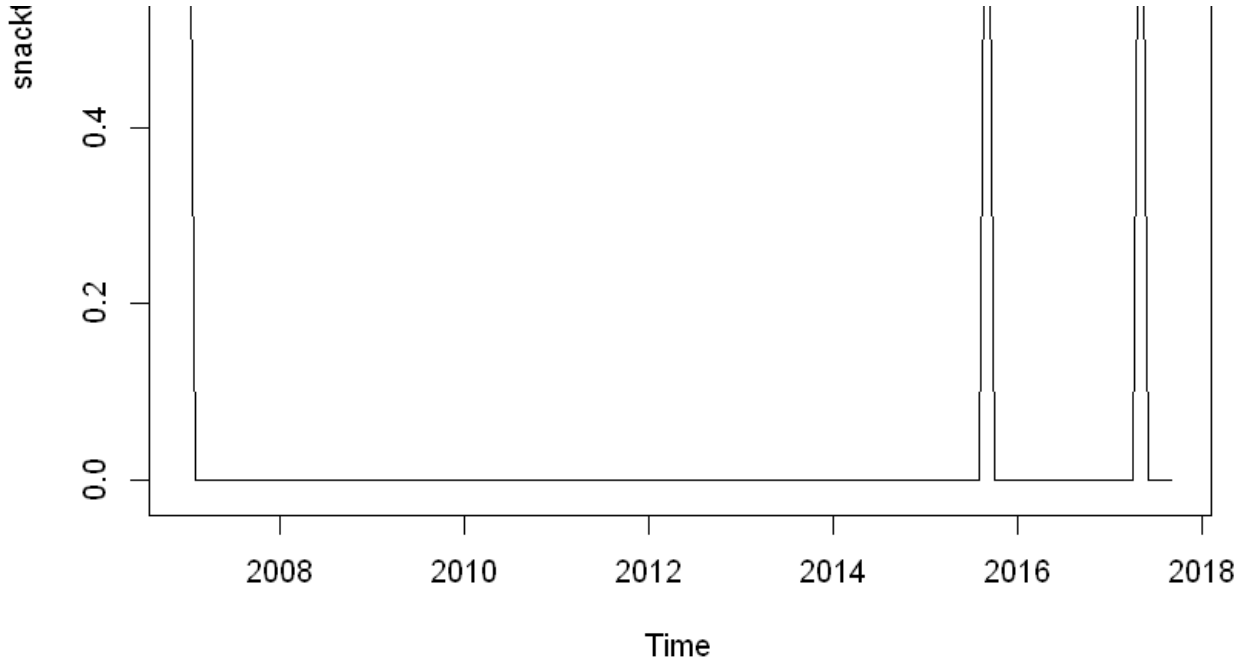


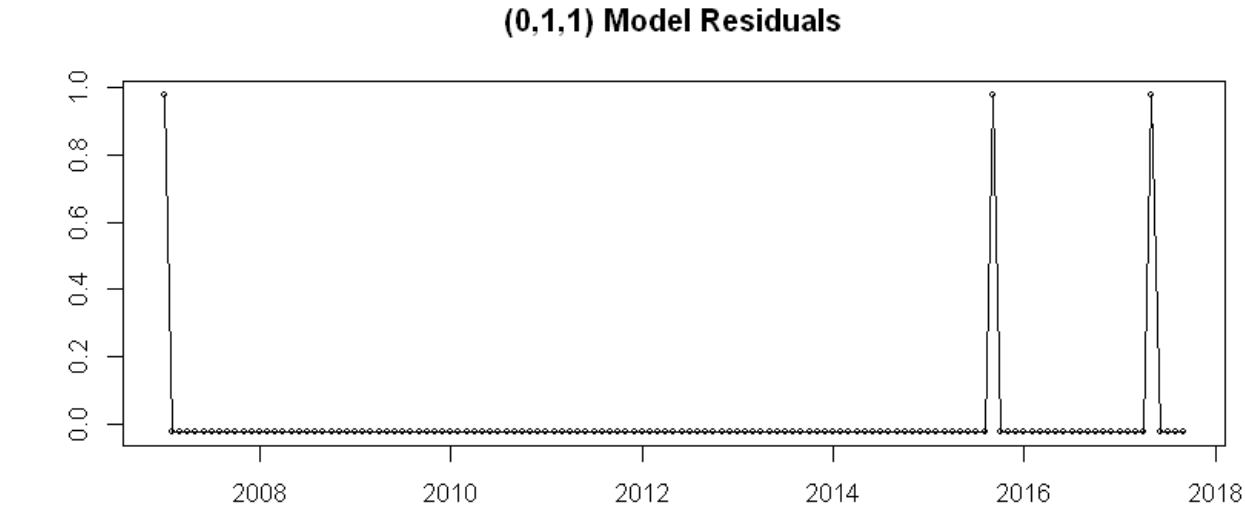
6- (2) 검색어 시계열 예측

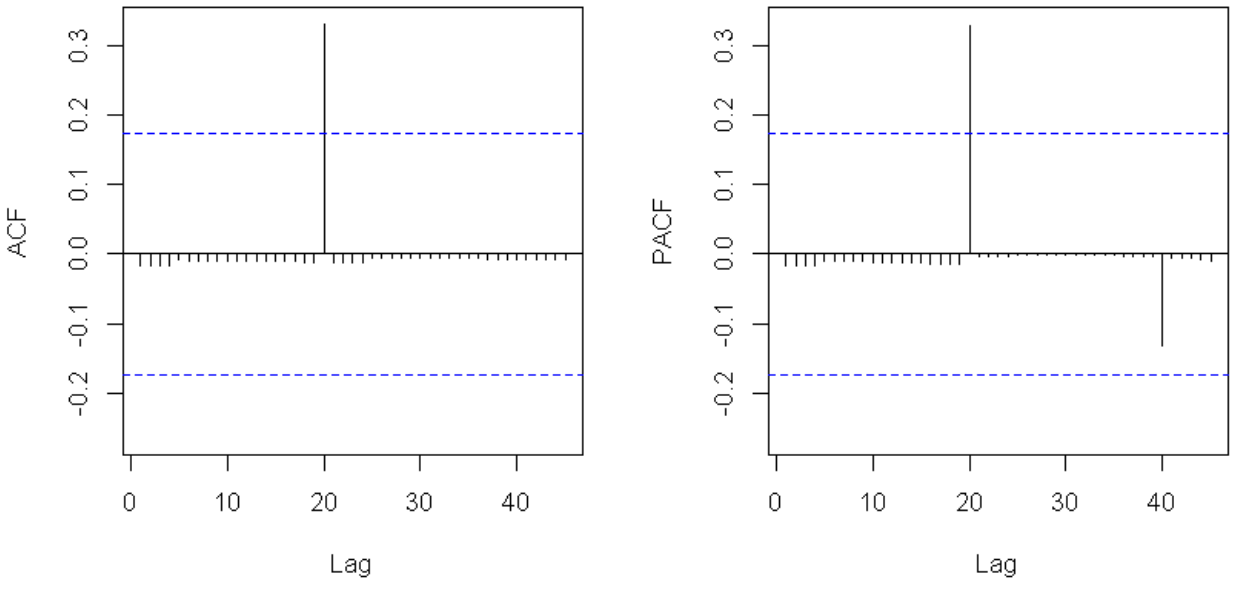
In [11]:

```
#시계열 자료로 바꾸어 준다.
snackts <- ts(yy, start=c(2007, 1), end=c(2017, 09), frequency=12)
plot(snackts)

#auto.arima 함수를 이용해보자
auto.arima(snackts, seasonal=FALSE)

# 적합시키면서 최적의 값을 찾아보자해보자
fit<-auto.arima(snackts, seasonal=FALSE)
tsdisplay(residuals(fit), lag.max=45, main='(0,1,1) Model Residuals')
```

```
Series: snackts
ARIMA(0,0,0) with non-zero mean

Coefficients:
         mean
      0.0233
s.e.  0.0133

sigma^2 estimated as 0.02289:  log likelihood=61.07
AIC=-118.14   AICc=-118.05   BIC=-112.42
```

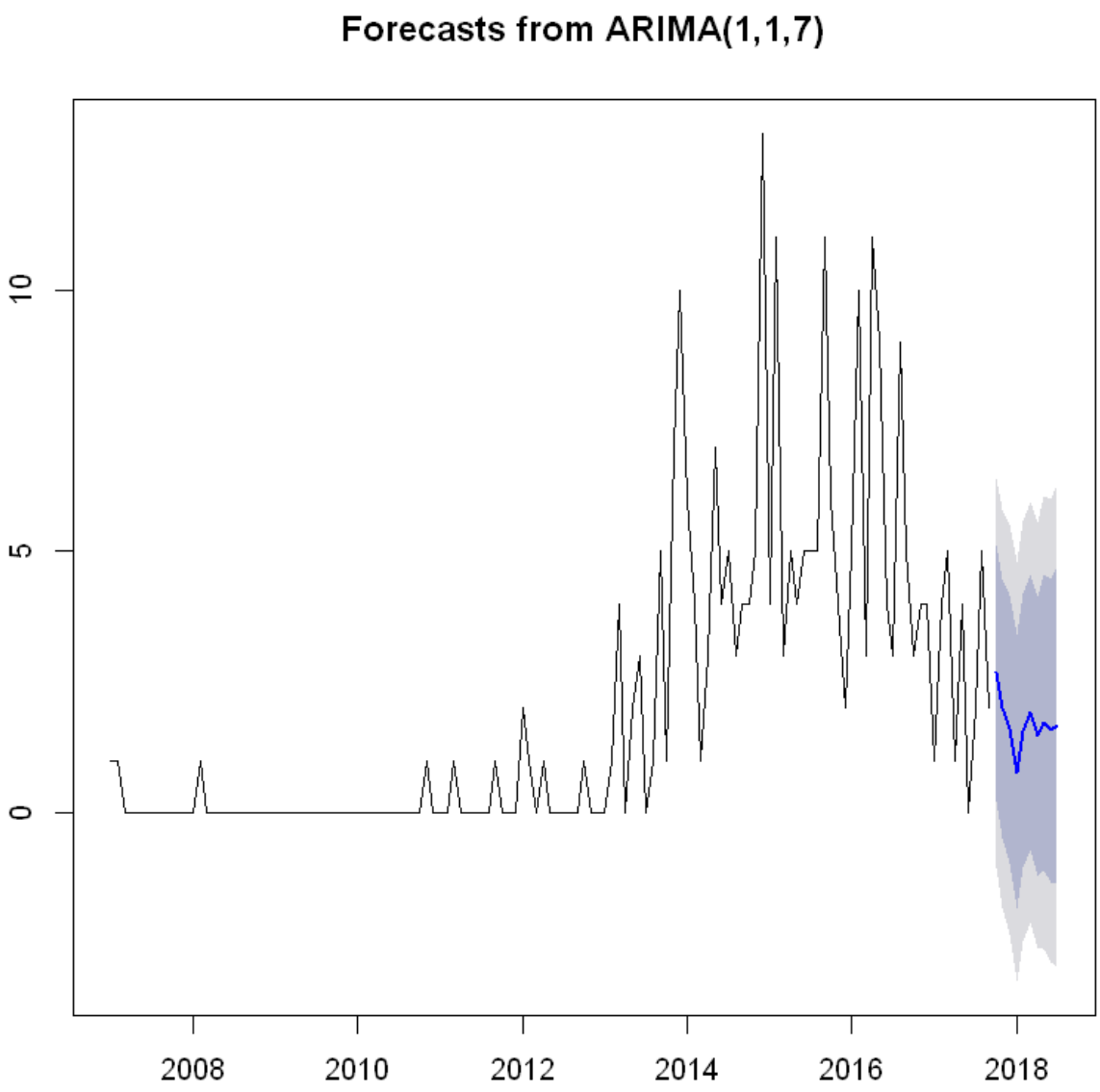

snackt
0.0
0.2
0.4
2008
2010
2012
2014
2016
2018
Time


(0,1,1) Model Residuals
0.0
0.2
0.4
0.6
0.8
1.0
2008
2010
2012
2014
2016
2018


ACF
PACF
-0.2
-0.1
0.0
0.1
0.2
0.3
0
10
20
30
40
Lag
Lag

In [6]:

```
#허니버터칩 시계열 예측
snack_arima <- arima(snackts, order = c(1,1,7))
snack_fcast <- forecast(snack_arima, h = 10)
plot(snack_fcast)
```

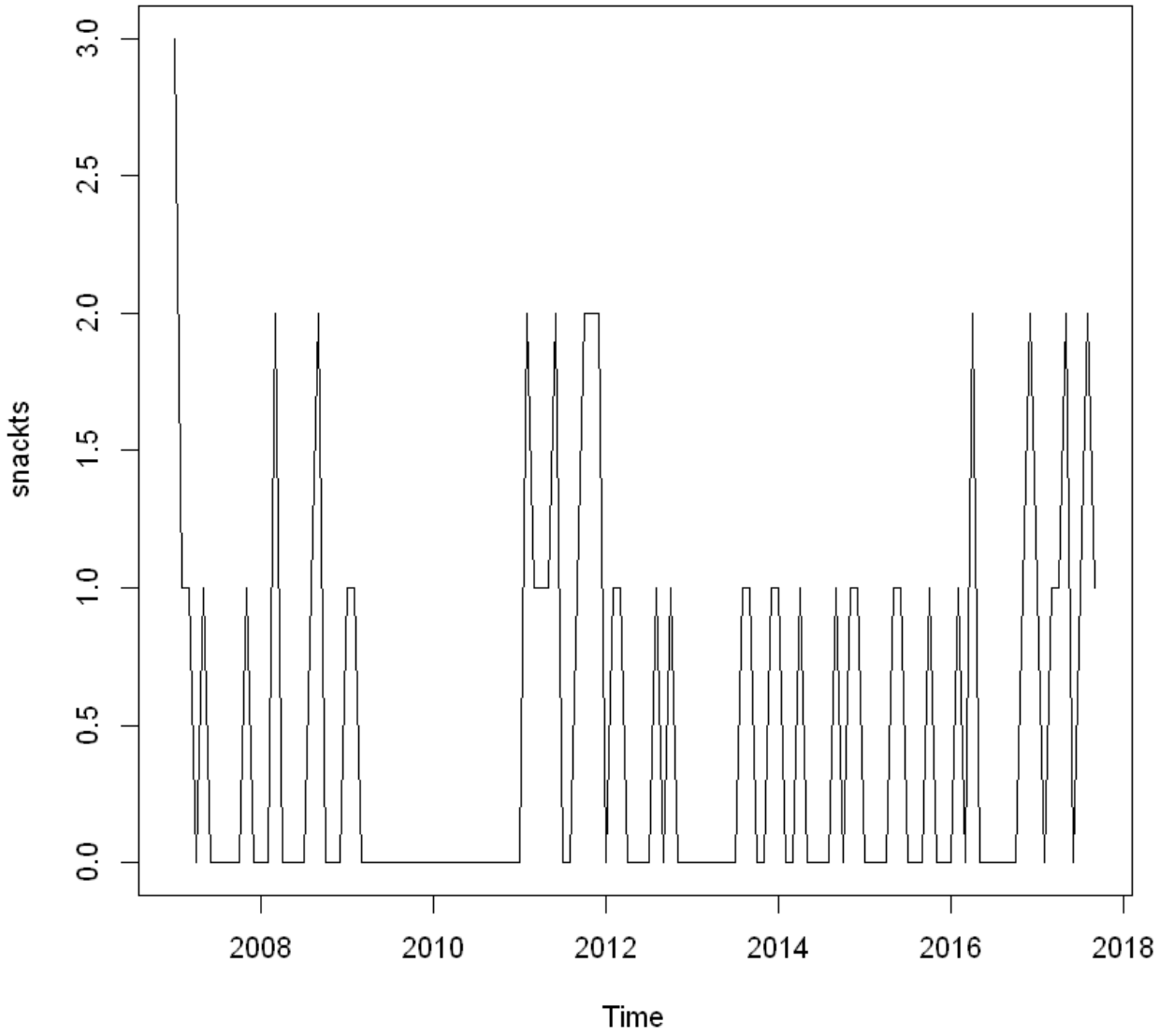


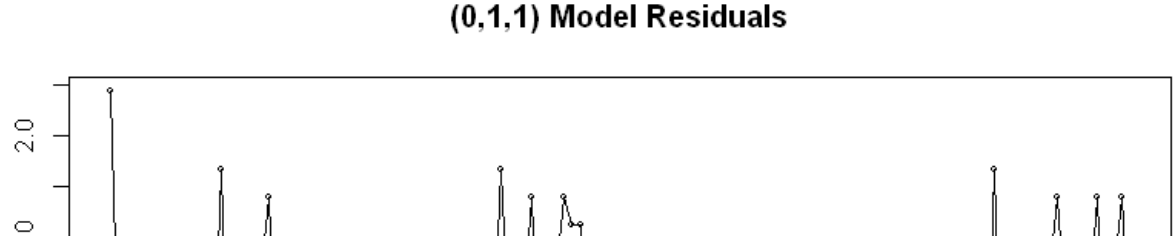
In [20]:

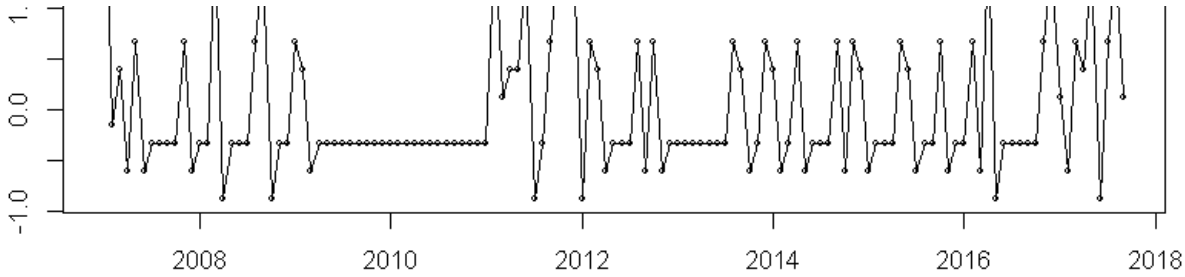
```
#시계열 자료로 바꾸어 준다.
snackts <- ts(yy, start=c(2007, 1), end=c(2017, 09), frequen
cy=12)
plot(snackts)

#auto.arima 함수를 이용해보자
auto.arima(snackts, seasonal=FALSE)

# 적합시키면서 최적의 값을 찾아보자해보자
fit<-auto.arima(snackts, seasonal=FALSE)
tsdisplay(residuals(fit), lag.max=45, main='(0,1,1) Model Re
```

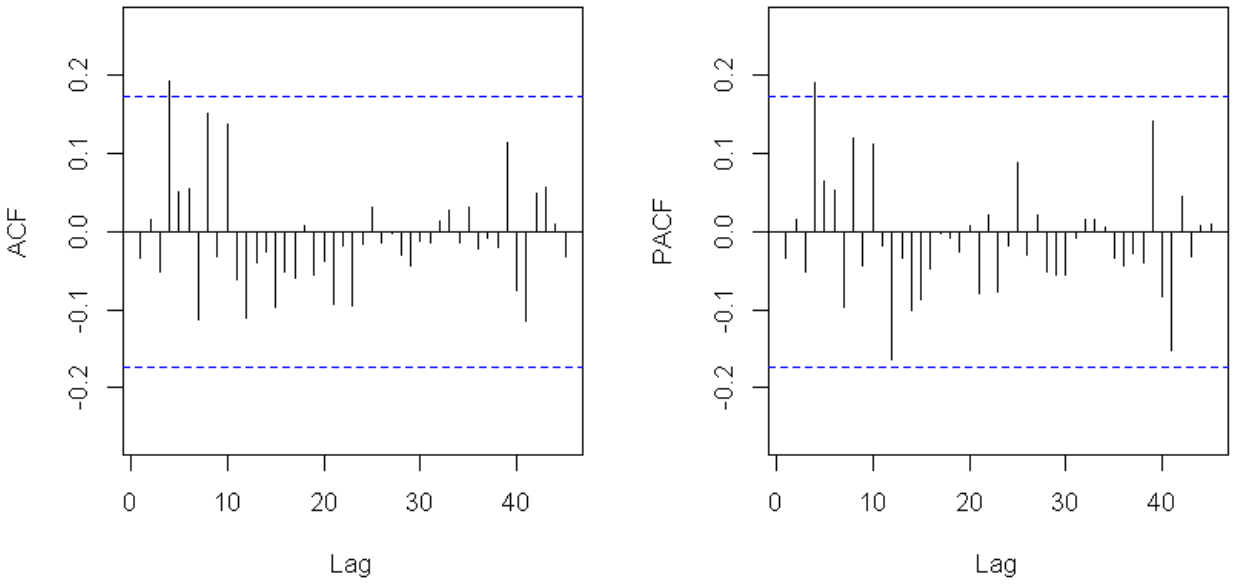
```
siduals')
```

```
Series: snackts
ARIMA(1,0,0) with non-zero mean

Coefficients:
         ar1    mean
      0.2713  0.4585
s.e.  0.0899  0.0794

sigma^2 estimated as 0.4405:  log likelihood=-129.2
AIC=264.4   AICc=264.59   BIC=272.98
```



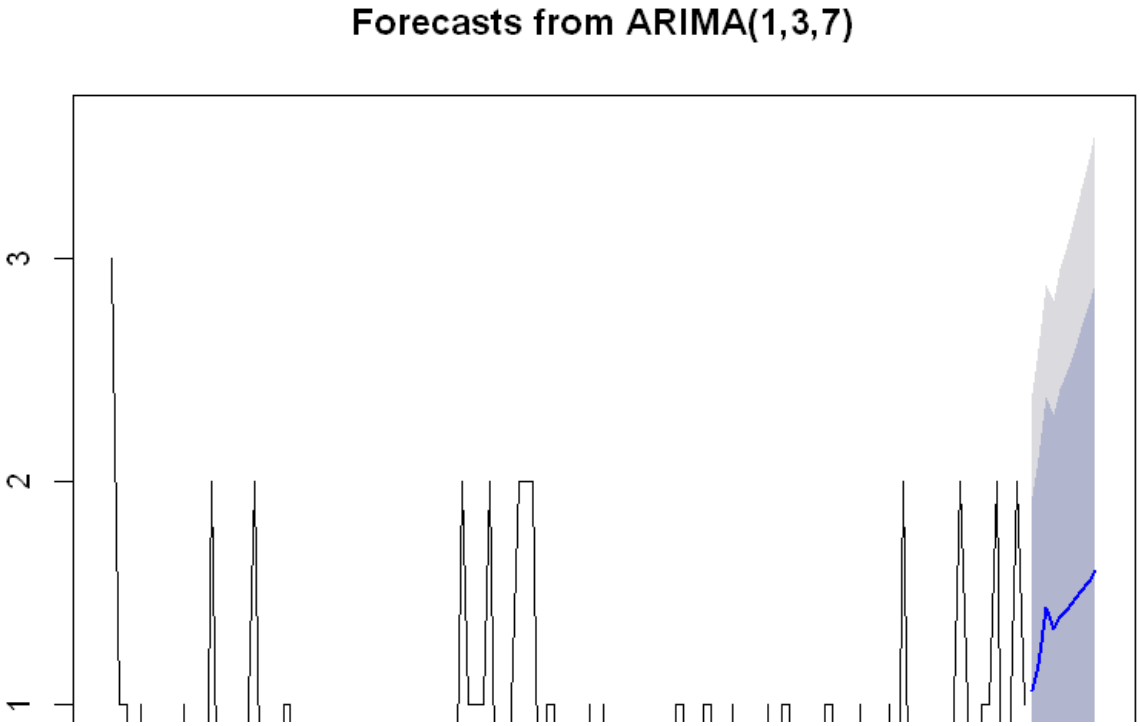


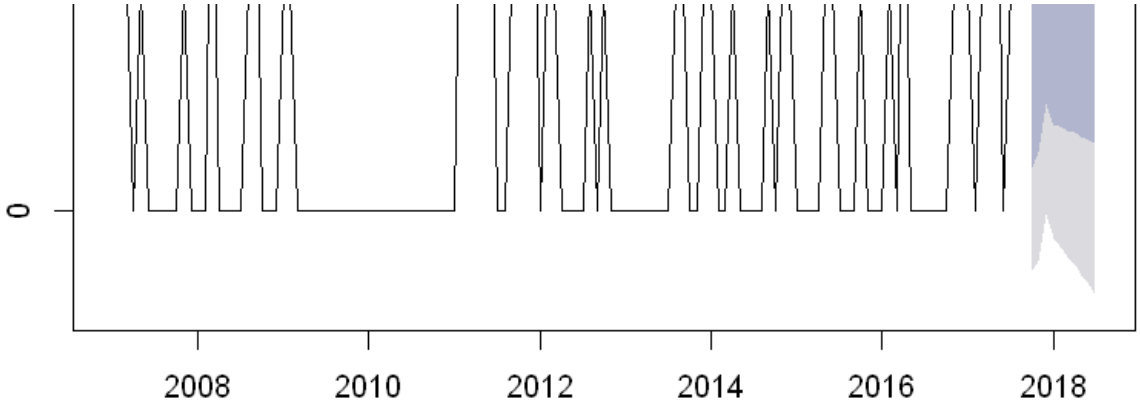




In [21]:

```
#이마트 노브랜드 감자칩 시계열 예측
snack_arima <- arima(snackts, order = c(1,3,7))
snack_fcast <- forecast(snack_arima, h = 10)
plot(snack_fcast)
```





* 시계열 예측 **summary**

- 허니버터칩은 검색어가 줄어드는 경향을 보이고, 이마트 노브랜드 감자칩의 검색어가 증가하는 추세를 보인다

7. Summary

Summary

- 가장 큰 특징을 보이는 것은 허니버터칩의 '만들기'와 포카칩'맛과 관련된' 단어들이다. 이 두개의 감자칩은 소비자들에게 확실한 브랜드 이미지를 각인 시킨 것으로 보인다.

- 같은 분석 방법을 시도하더라도, 수집된 출처 별로 다른 결과가 나올 것이라 생각했다. 하지만, 비슷한 결과가 나왔다.
- 다양한 분석들을 시도해 보았지만, 분석 방법의 이해가 부족하여 제대로 활용하지 못한 것 같다.
- 네이버 블로그 크롤링에서 **description**이 아닌 전체 내용을 받을 수 있는 방법을 찾아보았지만, 해결하지 못했다.
- 단어 정제과정도 부족하여, 과자와 관련된 단어들보다 블로그에서 흔히 볼 수 있는 인사말이 많이 나왔다.
