

# 데이터과학



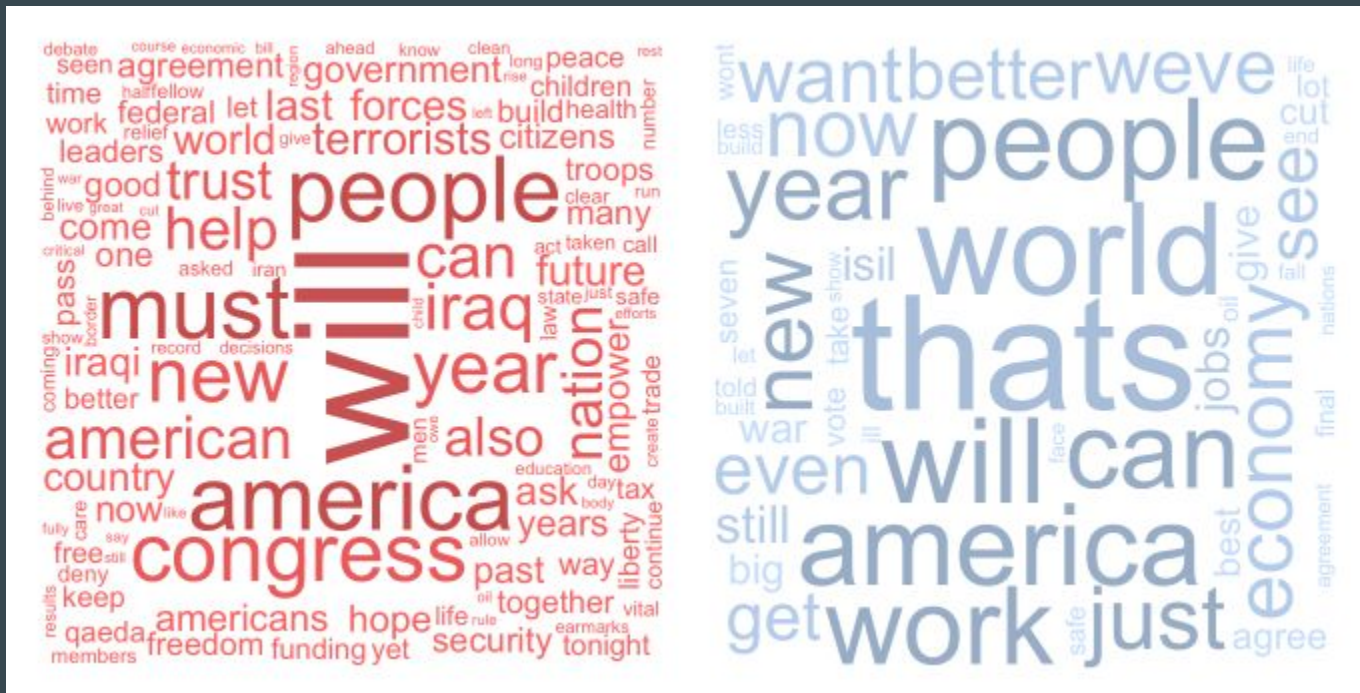
[7주차] WordCloud

# 목표

- 웹 페이지에 있는 Text 형태의 데이터를 수집하기
- 간단한 WordCloud 그리기
- 여러 Text에 대한 WordCloud를 통해 빈도수 비교
  - 공통단어
  - 빈도수 차이 비교

# WordCloud

- R에서 WordCloud 그리기
- 미국 대통령의 연설 비교
  - 부시, 오바마



# Data

- State of the Union
  - <http://stateoftheunion.onetwothree.net/>
- 미국 대통령들의 연설문이 Text형태로 존재

## STATE OF THE UNION

State of the UnionEssay

### State of the Union Address

< PreviousNext >

**Barack Obama**  
*January 27, 2010*

Madame Speaker, Vice President Biden, Members of Congress, distinguished guests, and fellow Americans:

Our Constitution declares that from time to time, the President shall give to Congress information about the state of our union. For two hundred and twenty years, our leaders have fulfilled this duty. They have done so during periods of prosperity and tranquility. And they have done so in the midst of war and depression; at moments of great strife and great struggle.

It's tempting to look back on these moments and assume that our progress was inevitable -- that America was always destined to succeed. But when the Union was turned back at Bull Run and the Allies first landed at Omaha Beach, victory was very much in doubt. When the market crashed on Black Tuesday and civil rights marchers were beaten on Bloody Sunday, the future was anything but certain. These were times that tested the courage of our convictions, and the strength of our union. And despite all our divisions and disagreements; our hesitations and our fears; America prevailed because we chose to move forward as one nation, and one people.

# 필요한 Library

- XML : 데이터 수집
- tm, dplyr, xtable : 데이터 가공
- wordcloud, RColorBrewer : 가시화

```
library(XML)
library(tm)
library(dplyr)
library(xtable)
library(wordcloud)
library(RColorBrewer)
```

# 데이터 가져오기

- onetwothree.net에 역대 대통령들의 연설문이 존재
- 웹사이트 URL 형태가 연/월/일 형태로 구성되어있기 때문에 입력을 연/월/일로 받아 해당되는 연설문을 추출하여 저장
- 2008년 연설문, 2016년 연설문을 추출하여 비교

```
speechtext <- function(ymd){  
  sotu <- data.frame(matrix(nrow=1,ncol=3))  
  colnames(sotu) = c("speechtext", "year", "date")  
  for(i in 1:length(ymd)){  
    year <- substr(ymd[i],1,4)  
    url <- paste0('http://stateoftheunion.onetwothree.net/texts/',ymd[i],'.html')  
    # 데이터가지고 오기  
    doc.html = htmlTreeParse(url, useInternal = TRUE)  
  
    # P태그에 존재하는 텍스트만 추출  
    doc.text = unlist(xpathApply(doc.html, '//p', xmlValue))  
  
    # 빈칸으로 구성된 것 또는 의미없는 newline제거  
    doc.text = gsub('WWn', '', doc.text)  
    doc.text = gsub('WW', '', doc.text)  
  
    doc.text = paste(doc.text, collapse = '')  
  
    # 연설문, 연도, 입력받은 data를 columns으로 설정하여 data.frame생성  
    x <- data.frame(doc.text, year, ymd[i], stringsAsFactors = FALSE)  
    names(x) <- c("speechtext", "year", "date")  
    sotu <- rbind(sotu, x)  
    # speechtext가 비어있으면(NA) 필터링  
    sotu <- sotu[!is.na(sotu$speechtext), ]  
  }  
  return(sotu)  
}  
  
sotu <- speechtext(c("20080128", "20160112"))
```

# 데이터 가공

- 웹에서 가져온 데이터는 raw 데이터 이므로 가공 필요
  - 세미콜론과 같은 특수문자(구두점) 제거
  - 숫자제거
  - 빈도수를 계산하기 위해 소문자로 통일
  - 조사 제거 (a, an, the....)
  - 빈칸제거
- 데이터 가공 이후 단어 빈도수를 Bush, Obama로 나누어서 하나의 matrix로 만듭니다.
- tm 과 dply 를 사용

```
docs <- Corpus(VectorSource(sotu$speechtext)) %>%
  tm_map(removePunctuation) %>%
  tm_map(removeNumbers) %>%
  tm_map(tolower) %>%
  tm_map(removeWords, stopwords("english")) %>%
  tm_map(stripWhitespace) %>%
  tm_map(PlainTextDocument)

tdm <- TermDocumentMatrix(docs) %>%
  as.matrix()
colnames(tdm) <- c("Bush", "Obama")

head(tdm)
```

```
##      Docs
## Terms  Bush Obama
## abandon    1    0
## ability     4    0
## abroad      2    0
## acceptable  1    0
## accepts     1    0
## access      2    0
```

# 데이터 가공

- 부시 대통령 column을 별도로 추출하여 내림차순으로 정렬

```
bushsotu <- as.matrix(tdm[,1])  
bushsotu <- as.matrix(bushsotu[order(bushsotu, decreasing=TRUE),])  
head(bushsotu)
```

```
##      [,1]  
## will    54  
## america 30  
## people  30  
## must    29  
## congress 27  
## new     25
```



# 데이터 가공

- 오바마대통령에 대해서도 동일하게 내림차순으로 정렬

```
obamasotu <- as.matrix(tdm[,2])  
obamasotu <- as.matrix(obamasotu[order(obamasotu, decreasing=TRUE),])  
head(obamasotu)
```

```
##      [,1]  
## thats  30  
## world  24  
## will   22  
## america 21  
## people 21  
## can    20
```

# Simple Word Cloud

- 두개 연설문에 대해 각각 Word Cloud 그리기

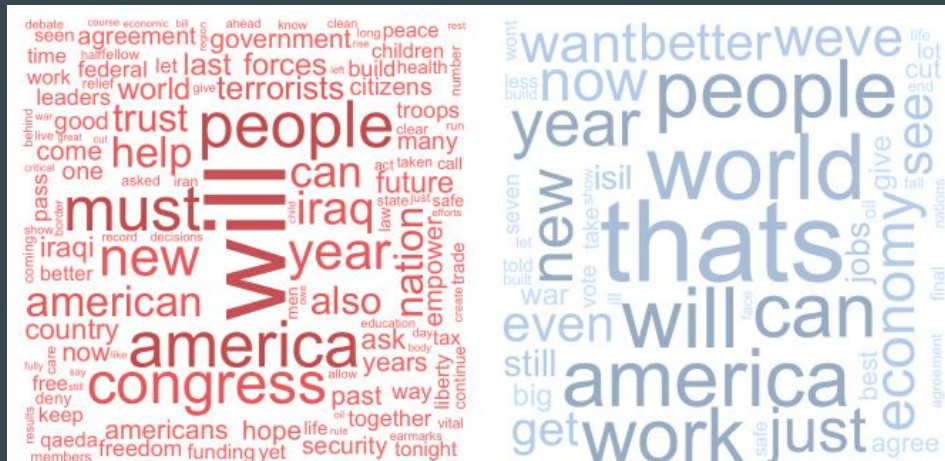
```
#Create Bush and Obama word clouds and plot them side-by-side
```

```
#Create two panels to add the word clouds to
```

```
par(mfrow=c(1,2))
```

```
wordcloud(rownames(bushsotu), bushsotu, min.freq =3, scale=c(5, .2), random.order = FALSE, random.color = FALSE, colors= c("indianred1","indianred2","indianred3","indianred"))
```

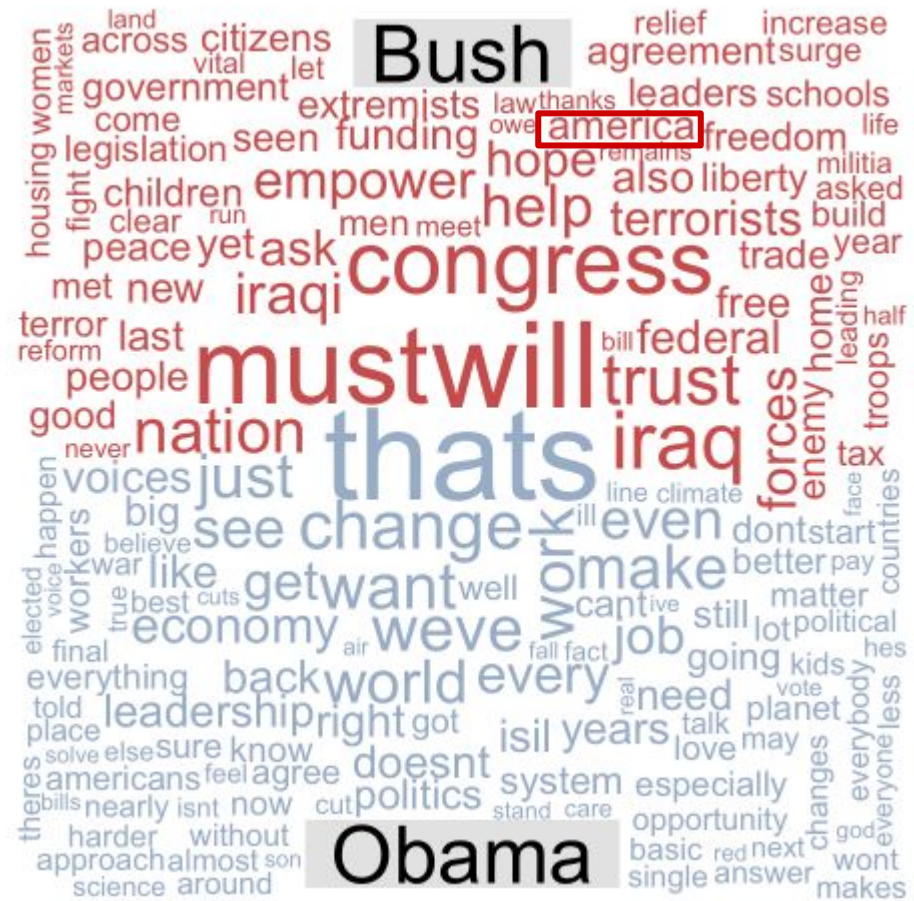
```
wordcloud(rownames(obamasotu), obamasotu, min.freq =3, scale=c(5, .2), random.order = FALSE, random.color = FALSE, colors= c("lightsteelblue1","lightsteelblue2","lightsteelblue3","lightsteelblue"))
```



# Comparision Cloud

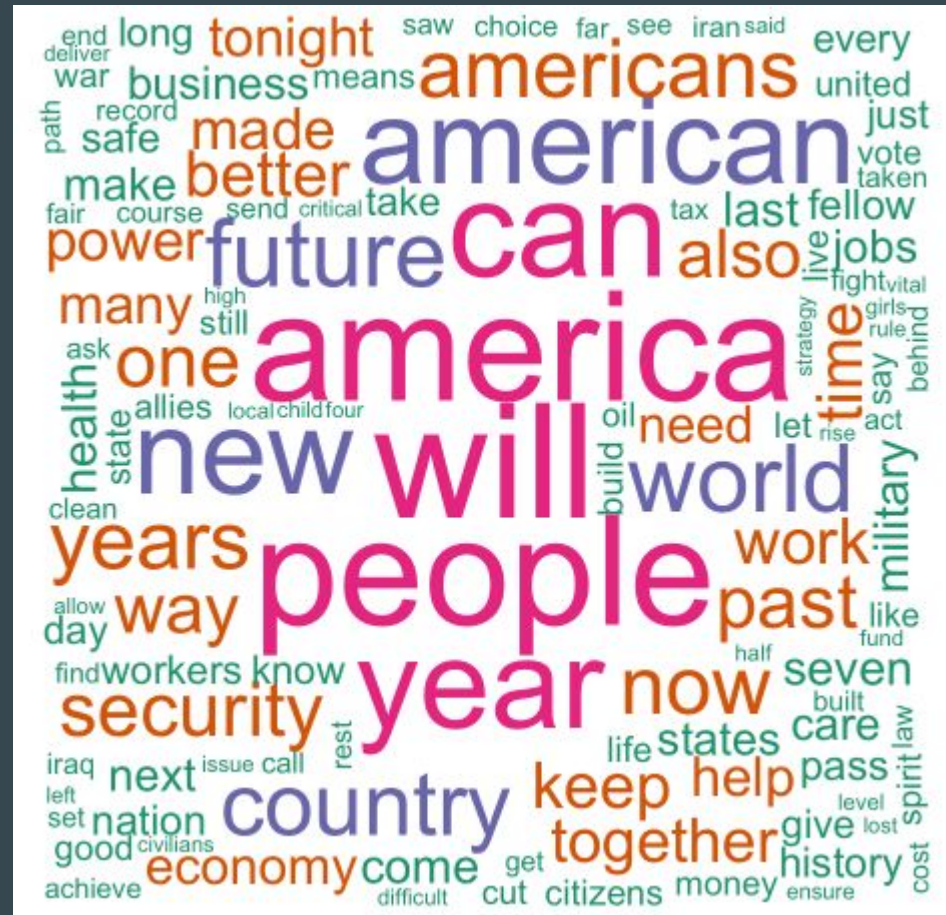
```
par(mfrow=c(1,1))
comparison.cloud(tdm, random.order=FALSE, colors = c("indianred3", "lightsteelblue3"),
                 title.size=2.5, max.words=400)
```

- 두개 혹은 두개이상의 문서를 비교할때 사용
- 각각의 row(단어)별 더 높은 빈도수를 가진 쪽에만 가시화시킴
- 부시대통령과 오바마대통령 모두 america라는 단어를 사용해 모두 출력되었으나 비교 그래프에서는 부시대통령쪽에만 존재
- 부시대통령이 america라는 단어를 더 많이 사용했음을 알 수 있음



# Commonality Cloud

- comparison cloud와 반대
- 두 문서에 공통으로 들어가있는 단어들에 대해서 만 가시화



```
commonality.cloud(tdm, random.order=FALSE, scale=c(5, .5), colors = brewer.pal(4, "Dark2"), max.words=400)
```

# 실습

- 앞선 과정을 모두 따라하면서 WordCloud 그리기
- 3개의 WordCloud모두 그린 후 스크린샷으로 제출
- 제출로 출석인정

# 웹 페이지에서 보기

- [https://hyunsik-yoo.github.io/Data\\_Analysis/wordcloud](https://hyunsik-yoo.github.io/Data_Analysis/wordcloud)