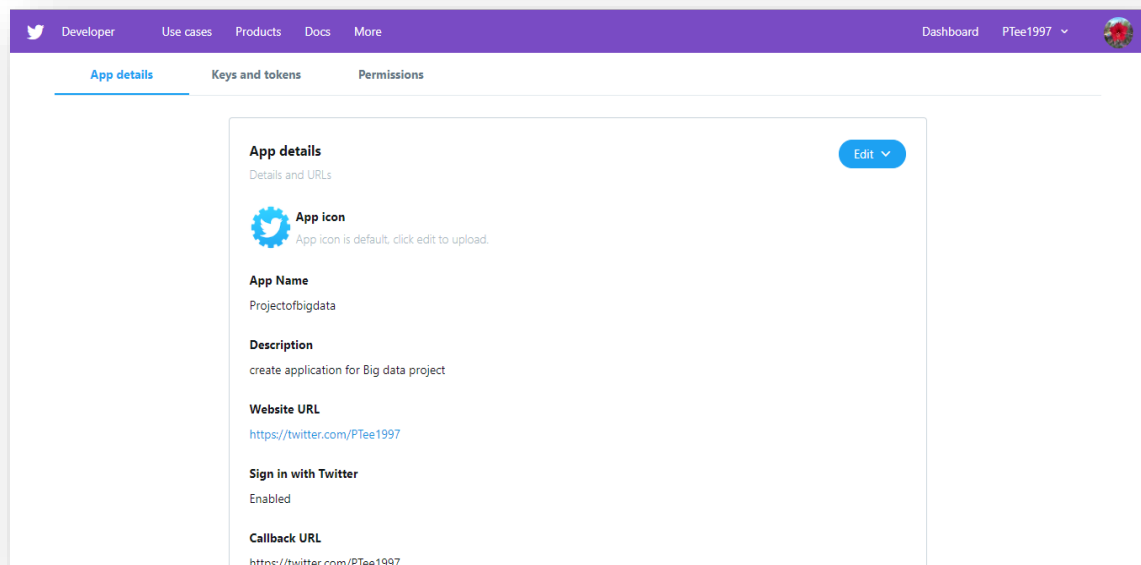
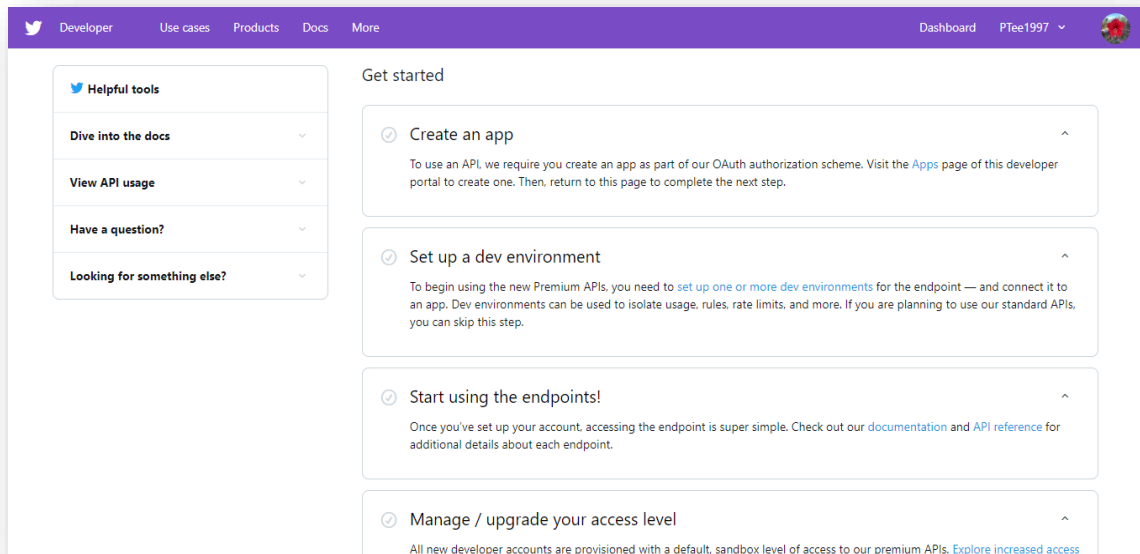




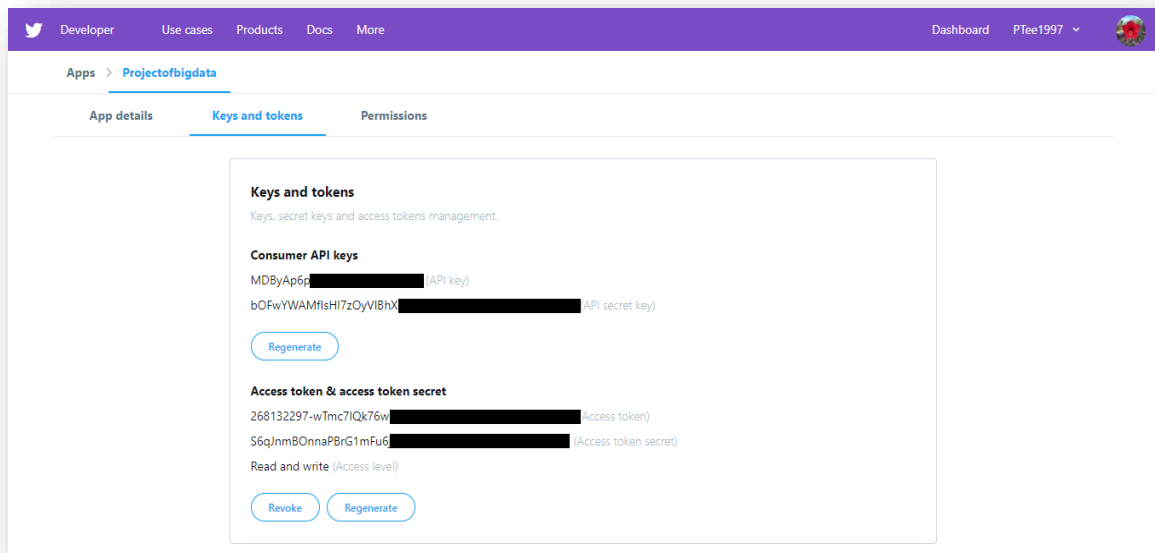
## 1. รับ API keys จากทวิตเตอร์

1.1 ) ขอ API จากทวิตเตอร์ จากลิ้งค์ <https://developer.twitter.com/en/apps>

1.2 ) ทวิตเตอร์จะส่งลิ้งค์เพื่อนำไปสู่การสร้าง App



1.3 ) เมื่อคลิกเข้าที่ Keys and token จะได้ Consumer API keys และ “Access token” and “Access token secret”



## 2. ใช้โปรแกรม R Studio ในการดึงข้อมูลและนำข้อมูลมาทำเป็น word cloud



### 2.1 ติดตั้ง package สำหรับเชื่อมต่อกับทวิตเตอร์

> `install.packages("twitter")` เป็นแพ็คเกจ R ที่ให้การเข้าถึง Twitter API รองรับการใช้งานส่วนใหญ่ของ API

```
> install.packages("twitter")
```

> `install.packages("RCurl")` เป็นแพ็คเกจเพื่อดึงข้อมูลกลับไปยังเครื่อง

```
> install.packages("RCurl")
```

> `require(twitter)` โหลดแพ็คเกจ twitter

> `require(RCurl)` โหลดแพ็คเกจ RCurl

```
> require(twitter)
Loading required package: twitter
> require(RCurl)
Loading required package: RCurl
Loading required package: bitops
>
```

## 2.2 ทำการเชื่อมต่อกับทวิตเตอร์ โดยนำ API ที่ได้ทำการขอจากทวิตเตอร์มาใช้

```
> consumer_key <- 'MDByAp6ptA1c[REDACTED]'
> consumer_secret <- 'boFwYwAMfIsHI7zOyVlBhxchB[REDACTED]'
> access_token <- '268132297-wTmc7IQk76wnjDT[REDACTED]'
> access_secret <- 's6qJnmBonnaPBrG1mFu6[REDACTED]'
> |
```

เรียกใช้ consumer\_key, consumer\_secret, access\_token, access\_secret  
โดยการใช้คำสั่ง

```
> setup_twitter_oauth(consumer_key, consumer_secret, access_token, access_secret)
[1] "Using direct authentication"
> |
```

เมื่อหน้า console ขึ้นว่า [1] "Using direct authentication" แสดงว่าเชื่อมต่อ Twitter ได้  
แล้ว

ลองดึงข้อมูลจาก twitter โดยใช้ Function : searchTwitter

```
> kpop <- searchTwitter("Junho+Nichkhun", lang="en", n=500, resultType="recent")
class(kpop)
```

- ฟังก์ชัน searchTwitter จะอยู่ในแพ็คเกจของ twitterR โดยจะทำการดึงข้อมูลคำว่า  
“Junho +(หมายความว่า and) Nichkhun” ส่วน lang="en" กำหนดเป็นภาษาอังกฤษ  
n=500 คือ เป็นจำนวนทวิตที่ต้องการจะดึงเข้ามา โดยเราจะค้นหาจำนวนทวิต500ทวิต  
ข้อมูลที่ถูกต้องจะเข้ามาเก็บในดาต้าที่ชื่อ kpop

จากคำค้นหาที่มีคำว่า Junho+Nichkhun มาเกี่ยวข้อง ที่เป็นภาษาอังกฤษ และเมื่อกด Enter  
จะใช้เวลาไม่กี่วินาทีและสิ่งที่มันทำคือการค้นหาใน Twitter และดึงทวิตที่เกี่ยวข้องกับคำค้นหา  
เหล่านี้

Name	Type	Value
--> id	character [1]	'1100020434349699072'
--> isRetweet	logical	TRUE
--> latitude	character [0]	
--> longitude	character [0]	
--> replyToSID	character [0]	
--> replyToSN	character [0]	
--> replyToUID	character [0]	
--> retweetCount	double [1]	2
--> retweeted	logical	FALSE
--> screenName	character [1]	'EHs2bXX9uPSQ1Za'
--> statusSource	character [1]	'<a href="http://twitter.com/download/android" rel="nofollow">Twitter for Android ...
--> text	character [1]	'RT @kimjeha90: 2PM is my 1st kpop boy group that made me Go Crazy into kpop wor ...
--> truncated	logical	FALSE

Environment	History	Connections
Global Environment		
Data		
kpop		
List of 357		
:Reference class 'status' [package "twitter"] with 17 fields		
..\$ text : chr "RT @Noona0125: #2PM are the Men of their words. if they said they will stay together that means they fucking wi"  __truncated__		
..\$ favorited : logi FALSE		
..\$ favoriteCount : num 0		
..\$ replyToSN : chr (0)		
..\$ created : POSIXct[1:1], format: "2019-02-25 13:16:16"		
..\$ truncated : logi FALSE		
..\$ replyToSID : chr (0)		
..\$ id : chr "1100021661997428736"		
..\$ replyToUID : chr (0)		
..\$ statusSource : chr "<a href='http://twitter.com/download/android' rel='nofollow'>Twitter for Android</a>"		
..\$ screenName : chr "Manutami"		
..\$ retweetCount : num 16		
..\$ isRetweet : logi TRUE		
..\$ retweeted : logi FALSE		
..\$ longitude : chr (0)		
..\$ latitude : chr (0)		
..\$ urls : 'data.frame': 0 obs. of 4 variables:		
..\$ url1 : chr (0)		
..\$ expanded_url : chr (0)		

## 2.3 ติดตั้งแพ็คเกจสำหรับดึงข้อมูลและแปลงข้อมูล

> install.packages("tm") แพ็คเกจคำสั่งให้อ่านข้อมูลแหล่งต่าง ๆ มาสร้างเป็นคลังข้อมูลได้  
\* tm --> (text mining)

```
> install.packages("tm")
```

> install.packages("wordcloud") เป็นแพ็คเกจที่ช่วยสร้างกราฟฟิคของคำตามขนาด  
ความถี่ของคำได้

```
> install.packages("wordcloud")
```

> require(tm) ใช้แพ็คเกจอ่านข้อมูลจากแหล่งต่าง ๆ มาสร้างเป็นคลังข้อมูล

```
> require(tm)
Loading required package: tm
Loading required package: NLP
```

> require(wordcloud) ใช้แพ็คเกจ wordcloud

```
> require(wordcloud)
Loading required package: wordcloud
Loading required package: RColorBrewer
```

> kpop\_text <- sapply(kpop, function(x) x\$get\_text())

- ฟังก์ชัน sapply เป็นฟังก์ชันปกติของ R อยู่แล้ว เพื่อแยก text จากทวีตให้อยู่ในรูป vector

- kpop เป็นข้อมูลที่ต้องการนำไปแปลงเป็น text โดยมีชื่อว่า kpop\_text

- function(x) x\$get\_text() คือ โดยส่วนใหญ่เราต้องการจะดึงเพียงข้อมูลเฉพาะจากทวีตที่มี  
ทั้งหมดจึงเรียกใช้ sapply

> str(kpop\_text) เพื่อเรียกดูข้อความจาก kpop\_text

- str() เป็นคำสั่งดูตัวอย่าง text ที่เกี่ยวข้องกับคำค้นหาที่เราต้องการแล้วจากโค้ดข้างบน

```
> kpop_text <- sapply(kpop, function(x) x$get_text())
> str(kpop_text)
chr [1:358] "RT @Noona0125: #2PM are the Men of their words. if they said they will stay together that means they fucking wi" | _truncated_ ...
```

ใช้คำสั่ง Corpus(VectorSource(ชื่อไฟล์)) ในการโหลดข้อมูลเพื่อให้ไฟล์เก็บไว้ใน  
directory เดียวกัน ซึ่ง Corpus จะอยู่ในส่วนของแพ็คเกจ tm

> pm\_corpus <- Corpus(VectorSource(kpop\_text)) โหลดข้อมูลเพื่อให้ไฟล์

kpop\_text เก็บไว้ใน directory เดียวกันที่ชื่อว่า pm\_corpus

> pm\_corpus เพื่อเรียกดู pm\_corpus ซึ่งจะเห็นว่ามี 358 documents ที่โหลดมาใน  
corpus นี้

```
> pm_corpus <- Corpus(VectorSource(kpop_text))
> pm_corpus
<<SimpleCorpus>>
Metadata: corpus specific: 1, document level (indexed): 0
Content: documents: 358
```

> inspect(pm\_corpus[ข้อมูลบรรทัดที่ต้องการดู]) คำสั่งที่ใช้ในการเรียกดูข้อมูลของไฟล์ในแต่ละบรรทัด

```
> inspect(pm_corpus[1])
<<SimpleCorpus>>
Metadata: corpus specific: 1, document level (indexed): 0
Content: documents: 1

[1] RT @Noona0125: #2PM are the Men of their words. if they said they will stay together that means they fucking will! #JUN_K #NICKKHUN #TAEYE...
> inspect(pm_corpus[100])
<<SimpleCorpus>>
Metadata: corpus specific: 1, document level (indexed): 0
Content: documents: 1

[1] RT @Elfiani-Phei: Nickkhun 1st Mini Album ME \u0001f60d + Junho Album TW0\#2PM #Nickkhun #Junho #투피엠 #닉쿤 #준호 https://t.co/zB5d2x6H1o
```

## 2.4 ทำการ cleaning เพื่อนำมาทำเป็น word cloud

คำที่เป็นไวยากรณ์ต่าง ๆ สามารถลบออก (เรียกว่า stop-words) รวมทั้งลบเครื่องหมายวรรคตอนก็สามารถทำได้โดยใช้คำสั่ง

> pm\_clean <- tm\_map(pm\_corpus, คำสั่งในการลบ)

- pm\_clean สร้าง object ใหม่เรียกว่า pm สร้างมาเพื่อเพื่อทำการ clean

- tm\_map() ใช้ฟังก์ชันของแพ็คเกจ tm และฟังก์ชัน tm ที่อยู่บน map

- pm\_corpus เอา corpus ที่เพิ่งสร้างมาเป็น argument

> pm\_clean <- tm\_map(pm\_corpus, removePunctuation)

removePunctuation สร้างคลังข้อมูลใหม่เพื่อที่จะลบเครื่องหมายวรรคตอนออกไป

> pm\_clean <- tm\_map(pm\_clean, content\_transformer(tolower))

content\_transformer(tolower) สร้างคลังข้อมูลใหม่โดยแปลงพิมพ์ใหญ่เป็นพิมพ์เล็กทั้งหมด

> pm\_clean <- tm\_map(pm\_clean, removeWords, stopwords("en"))

สร้างคลังข้อมูลใหม่โดยลบคำออก โดยการใส่ฟังก์ชัน stopwords() ซึ่ง stopwords() ไม่เอาคำซึ่งเป็นคำที่ไม่น่าสนใจ เช่น and,the, so ที่มีอยู่ในภาษาอังกฤษ

> pm\_clean <- tm\_map(pm\_clean, removeNumbers)

สร้างคลังข้อมูลใหม่โดยลบตัวเลขออก

> pm\_clean <- tm\_map(pm\_clean, stripWhitespace)

สร้างคลังข้อมูลใหม่โดยลบช่องว่างหรือ Whitespace ออก เนื่องจากเมื่อมีการลบส่วนอื่น ๆ ออกหมดแล้วก็มีช่องว่างมาก เพื่อเก็บช่องว่างที่เกิดขึ้นจึงใช้คำสั่งนี้

```
> pm_clean <- tm_map(pm_corpus, removePunctuation)
warning message:
In tm_map.SimpleCorpus(pm_corpus, removePunctuation) :
  transformation drops documents
> pm_clean <- tm_map(pm_clean, content_transformer(tolower))
warning message:
In tm_map.SimpleCorpus(pm_clean, content_transformer(tolower)) :
  transformation drops documents
> pm_clean <- tm_map(pm_clean, removeWords, stopwords("en"))
warning message:
In tm_map.SimpleCorpus(pm_clean, removeWords, stopwords("en")) :
  transformation drops documents
> pm_clean <- tm_map(pm_clean, removeNumbers)
warning message:
In tm_map.SimpleCorpus(pm_clean, removeNumbers) :
  transformation drops documents
> pm_clean <- tm_map(pm_clean, stripWhitespace)
warning message:
In tm_map.SimpleCorpus(pm_clean, stripWhitespace) :
  transformation drops documents
```

```
> pm_clean <- tm_map(pm_clean, removewords, c("nickkhun", "junho", "dick", "https"))
warning message:
In tm_map.SimpleCorpus(pm_clean, removewords, c("nickkhun", "junho", :
  transformation drops documents
```

```
> wordcloud(pm_clean, random.order = F, max.words=600, scale=c(3,0.5),
colors=brewer.pal(8,"Dark2") )
```

- random.order = F ทำให้คำที่มีความถี่สูงจะปรากฏอยู่กลางภาพ
- max.words=600 คำที่ต้องการให้ปรากฏบนภาพจำนวน 600 คำ
- scale=c(3,0.5) กำหนดขนาดตัวอักษรขนาดใหญ่ที่สุดและขนาดตัวอักษรเล็กที่สุด
- colors=brewer.pal(8,"Dark2") กำหนดสีให้ตัวอักษรที่อยู่บนภาพ

[illegible]



## Source Code

```
1 install.packages("twitter")
2 install.packages("RCurl")
3 require(twitter)
4 require(RCurl)
5
6
7
8 #twitter API authorization process
9 consumer_key <- 'MDByAp6ptA1qfwgg12JxOWiAN'
10 consumer_secret <- 'boFwYwAMfIsHI7zOyVlBhxchBVckbjRovwgrbCojkuRg4Y7rnp'
11 access_token <- '268132297-wTmc7IQk76wnjDTk0LH1SyQQZzaq9dnuvokjXqiu'
12 access_secret <- 'S6qJnmBonnaPBrGlmFu6jwmM4hzoZB6aw95vQONpx3u0G'
13 setup_twitter_oauth(consumer_key, consumer_secret, access_token, access_secret)
14
15
16 #search examples from function help file
17 kpop <- searchTwitter("Junho+Nichkhun", lang="en", n=500, resultType="recent")
18 class(kpop)
19
20
21 #convert list to vector
22 kpop_text <- sapply(kpop, function(x) x$text)
23 str(kpop_text)
24
25
26 install.packages("tm")
27 install.packages("wordcloud")
28 require(tm)
29 require(wordcloud)
30
31 kpop_text <- sapply(kpop, function(x) x$text)
32 str(kpop_text)
33
34 #create corpus from vector of tweets
35 pm_corpus <- Corpus(VectorSource(kpop_text))
36 pm_corpus
37 inspect(pm_corpus[1])
38 inspect(pm_corpus[100])
39
40 #lower cases, remove numbers, cut out stopwords, remove punctuation, strip whitespace
41 pm_clean <- tm_map(pm_corpus, removePunctuation)
42 pm_clean <- tm_map(pm_clean, content_transformer(tolower))
43 pm_clean <- tm_map(pm_clean, removeWords, stopwords("en"))
44 pm_clean <- tm_map(pm_clean, removeNumbers)
45 pm_clean <- tm_map(pm_clean, stripWhitespace)
46
47 #you may want to remove search words these will obviously be very frequent
48 pm_clean <- tm_map(pm_clean, removeWords, c("nichkhun", "junho", "dick", "https"))
49
50 #wordcloud, play with parameters
51 wordcloud(pm_clean, random.order = F, max.words=600, scale=c(3,0.5), colors=brewer.pal(8,"dark2"))
52
53
54:1 wordcloud, play with parameters
R Markdown
```