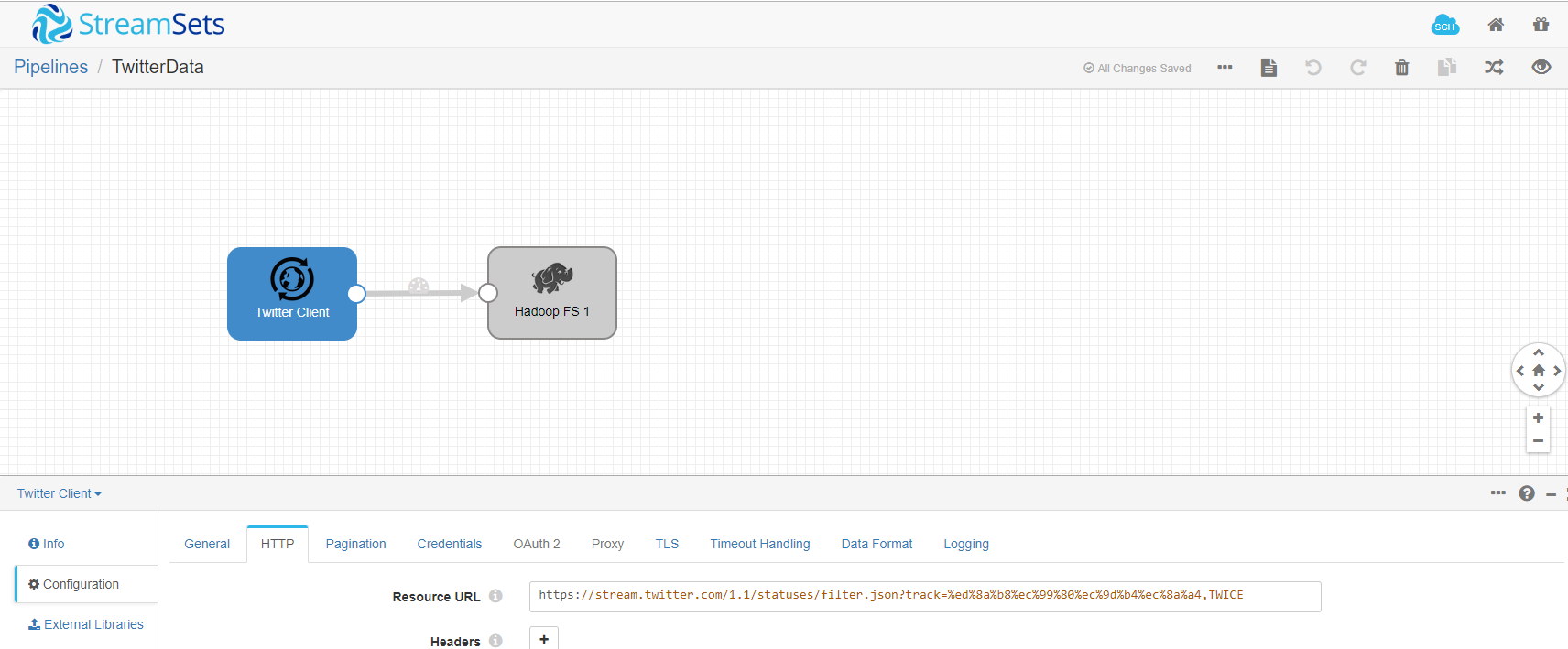
Twitter Data를 이용한 감성분석

(이삭엔지니어링 인턴 김형근 fnfn9087@gmail.com)  
(StreamSets, Hive, Kibana 모두 CDH 5.15에서 진행.)

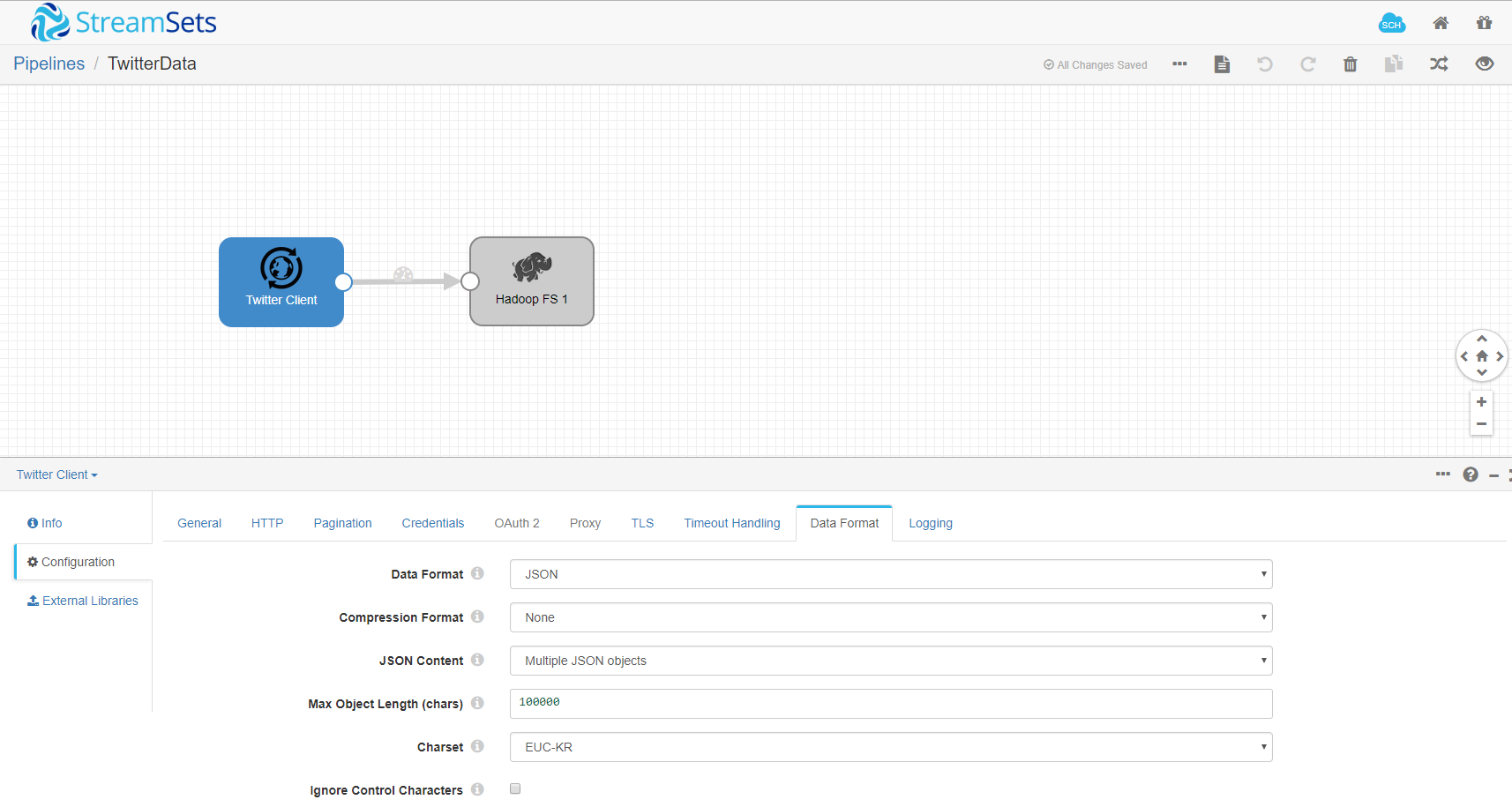
**Index**

1. **StreamSets로 트위터 데이터 받기**
2. **Hue에서 Hive Table 만들기  
   (1) Hue에서 Hive Table 만들기  
   (2) Timestamp type이 포함된 Hive Table 만들기**
3. **Hive Table을 Kibana에서 분석  
   (1) Kibana에서 사용할 external 테이블을 생성  
   (2) Dashboard 생성**
4. **Excel 3D Map에서 분석  
   (1) Hive Table 가져오기 : “데이터 탭” -> “새 쿼리” -> “기타 원본에서” - > “ODBC에서”  
   (2) Hive Table 가져오기 : “데이터 탭” -> “기타 원본에서” - > “데이터 연결 마법사에서”  
   (3) Excel 3D Map 사용  
   + 부록 1,2**

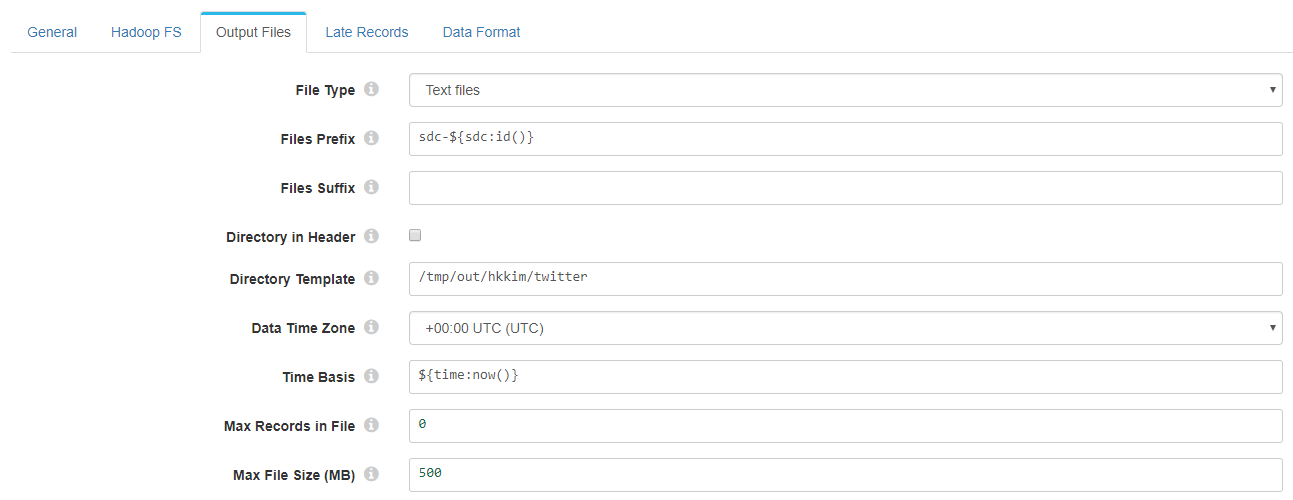
1. **Streamsets로 트위터 데이터 받기( Topic : TWICE(걸그룹) )**



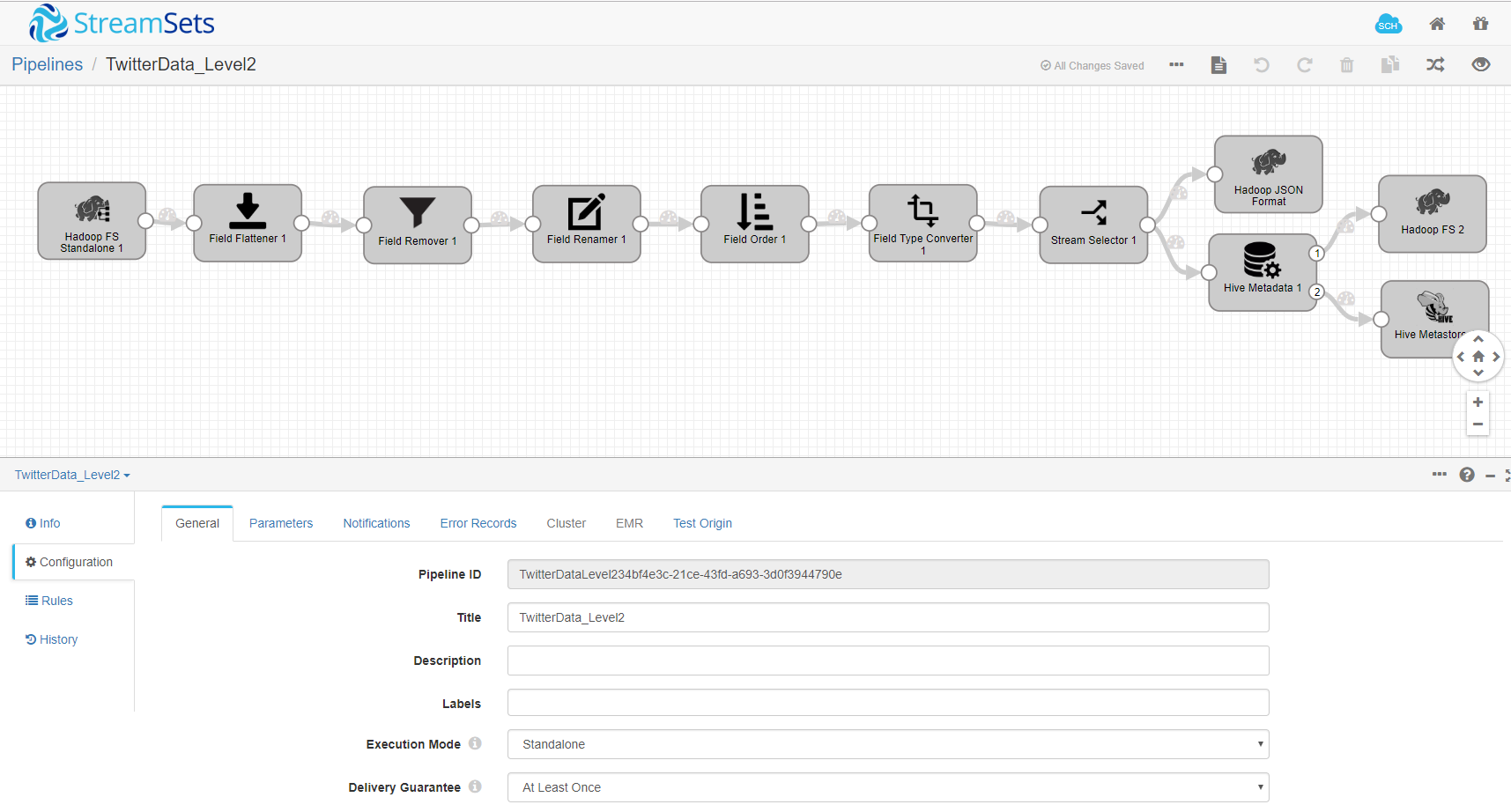
Resource : https://stream.twitter.com/1.1/statuses/filter.json?track=%ed%8a%b8%ec%99%80%ec%9d%b4%ec%8a%a4,TWICE



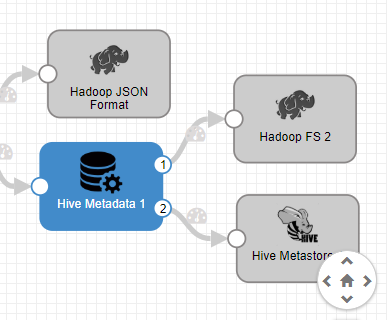
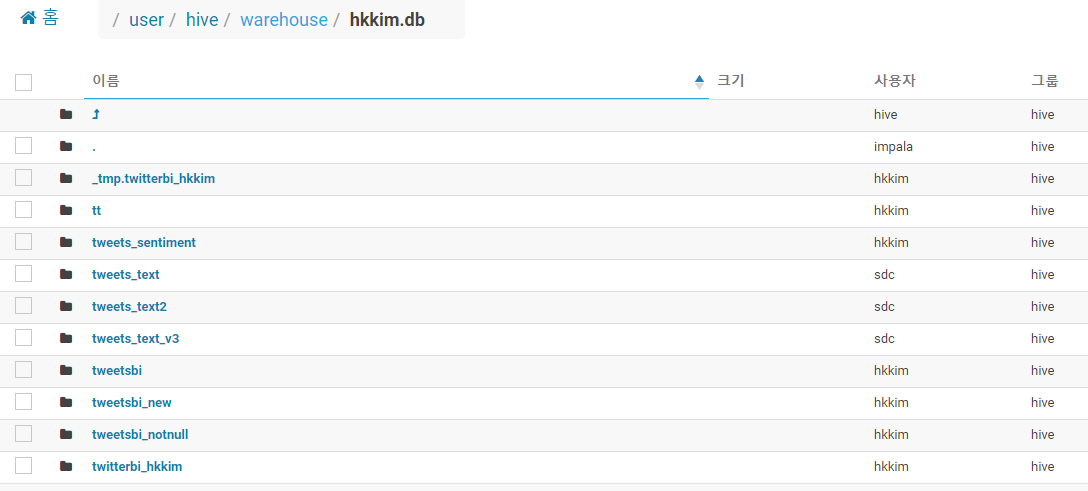
*Data Format : JSON  
File당 사이즈 : 500MB  
Streaming한 데이터를 저장하는 위치 : /tmp/out/hkkim/twitter*



1. 받아온 Twitter Data들을 JSON파일로 정리하고, 다시 JSON의 Field들 중에서 몇몇 요소들만 골라낸다.

  
  
  
  
  
  
2) “Field Remover”에서 필요한 Field들만 남겨놓는다.  
붉은색 field는 대부분이 NULL이지만, 가끔씩 값이 들어있는 경우가 있다.  
( *id, text, created\_at, timestamp\_ms, lang, user.location, user.name, user.id, extended\_tweet.full\_text, retweeted\_status.text, retweet\_status.extended\_tweet.full\_text, place.location, place.country* )  


3)정리한 데이터들을 JSON파일로도 저장해두고, Hive Table도 생성해둔다.  
(Table명 : tweets\_text2)

1. **Hue에서 Hive Table 만들기  
   2.(1) Hue에서 Hive Table 만들기  
     
   dictionary 테이블 만들기**CREATE EXTERNAL TABLE if not exists dictionary   
   ( type string,   
   length int,   
   word string,   
   pos string,   
   stemmed string,   
   polarity string )   
   ROW FORMAT DELIMITED FIELDS TERMINATED BY 't'   
   STORED AS TEXTFILE LOCATION '/user/hkkim/twitter/data/tables/dictionary';  
   **tweets\_clean 뷰 만들기**  
   CREATE VIEW IF NOT EXISTS tweets\_clean AS   
   SELECT tweet\_id, ts, msg, country FROM tweets\_simple;  
     
   **l1, l2, l3 뷰 만들기**create view IF NOT EXISTS l1 as select tweet\_id, words from tweets\_text2 lateral view explode(sentences(lower(msg))) dummy as words;   
     
   create view IF NOT EXISTS l2 as select tweet\_id, word from l1 lateral view explode( words ) dummy as word;   
     
   create view IF NOT EXISTS l3 as select   
   tweet\_id,   
   l2.word,   
   case d.polarity   
   when 'negative' then -1   
   when 'positive' then 1   
   else 0 end as polarity   
   from l2 left outer join dictionary d on l2.word = d.word;

**tweets\_sentiment 테이블 만들기**

create table IF NOT EXISTS tweets\_sentiment as select   
tweet\_id,   
case   
when sum( polarity ) > 0 then 'positive'   
when sum( polarity ) < 0 then 'negative'   
else 'neutral' end as sentiment   
from l3 group by tweet\_id;  
**tweetsbi 테이블 만들기**  
CREATE TABLE IF NOT EXISTS tweetsbi AS SELECT   
t.\*,   
case s.sentiment   
when 'positive' then 2   
when 'neutral' then 1   
when 'negative' then 0   
end as sentiment   
FROM tweets\_clean t LEFT OUTER JOIN tweets\_sentiment s on t.tweet\_id = s.tweet\_id;  
  
**tweetsbi\_notnull 테이블 만들기   
(sentiment 및 country에 NULL이 다수 포함되어 있어 NULL을 없앰)**create table tweetsbi\_notnull as   
select \* from tweetsbi where country != 'NULL' and sentiment is not null

**tweetsbi\_new 테이블 만들기  
(tweets\_text2의 lang Column도 포함)**create table tweetsbi\_new as   
select tweetsbi\_notnull.\*, tweets\_text2.lang    
from tweetsbi\_notnull left outer join tweets\_text2   
on tweetsbi\_notnull.tweet\_id = tweets\_text2.tweet\_id

**2.(1) Timestamp type이 포함된 Hive Table 만들기  
dictionary 테이블 만들기**CREATE EXTERNAL TABLE if not exists dictionary   
( type string,   
length int,   
word string,   
pos string,   
stemmed string,   
polarity string )   
ROW FORMAT DELIMITED FIELDS TERMINATED BY 't'   
STORED AS TEXTFILE LOCATION '/user/hkkim/twitter/data/tables/dictionary';  
  
  
**tweets\_clean 뷰 만들기**  
CREATE VIEW IF NOT EXISTS tweets\_clean AS   
SELECT tweet\_id,   
cast( from\_unixtime( cast ( substring ( created\_unixtime,1,10) as bigint ) ) as timestamp) ts,  
msg, country FROM tweets\_text2;  
  
**l1, l2, l3 뷰 만들기**create view IF NOT EXISTS l1 as select tweet\_id, words from tweets\_text2 lateral view explode(sentences(lower(msg))) dummy as words;   
  
create view IF NOT EXISTS l2 as select tweet\_id, word from l1 lateral view explode( words ) dummy as word;   
  
create view IF NOT EXISTS l3 as select   
tweet\_id,   
l2.word,   
case d.polarity   
when 'negative' then -1   
when 'positive' then 1   
else 0 end as polarity   
from l2 left outer join dictionary d on l2.word = d.word;

**tweets\_sentiment 테이블 만들기**

create table IF NOT EXISTS tweets\_sentiment as select   
tweet\_id,   
case   
when sum( polarity ) > 0 then 'positive'   
when sum( polarity ) < 0 then 'negative'   
else 'neutral' end as sentiment   
from l3 group by tweet\_id;  
  
**tweetsbi 테이블 만들기**  
CREATE TABLE IF NOT EXISTS tweetsbi AS SELECT   
t.\*,   
case s.sentiment   
when 'positive' then 2   
when 'neutral' then 1   
when 'negative' then 0   
end as sentiment   
FROM tweets\_clean t LEFT OUTER JOIN tweets\_sentiment s on t.tweet\_id = s.tweet\_id;  
  
**tweetsbi\_notnull 테이블 만들기   
(sentiment 및 country에 NULL이 다수 포함되어 있어 NULL을 없앰)**create table tweetsbi\_notnull as   
select \* from tweetsbi where country != 'NULL' and sentiment is not null

**tweetsbi\_new 테이블 만들기  
(tweets\_text2의 lang Column도 포함)**create table tweetsbi\_new as   
select tweetsbi\_notnull.\*, tweets\_text2.lang    
from tweetsbi\_notnull left outer join tweets\_text2   
on tweetsbi\_notnull.tweet\_id = tweets\_text2.tweet\_id

1. **Hive Table을 Kibana에서 분석  
   1)Kibana에서 사용할 external 테이블을 생성한다.**

CREATE EXTERNAL TABLE twitterbi\_hkkim\_kibana (

tweet\_id string,

ts string,

msg string,

country string,

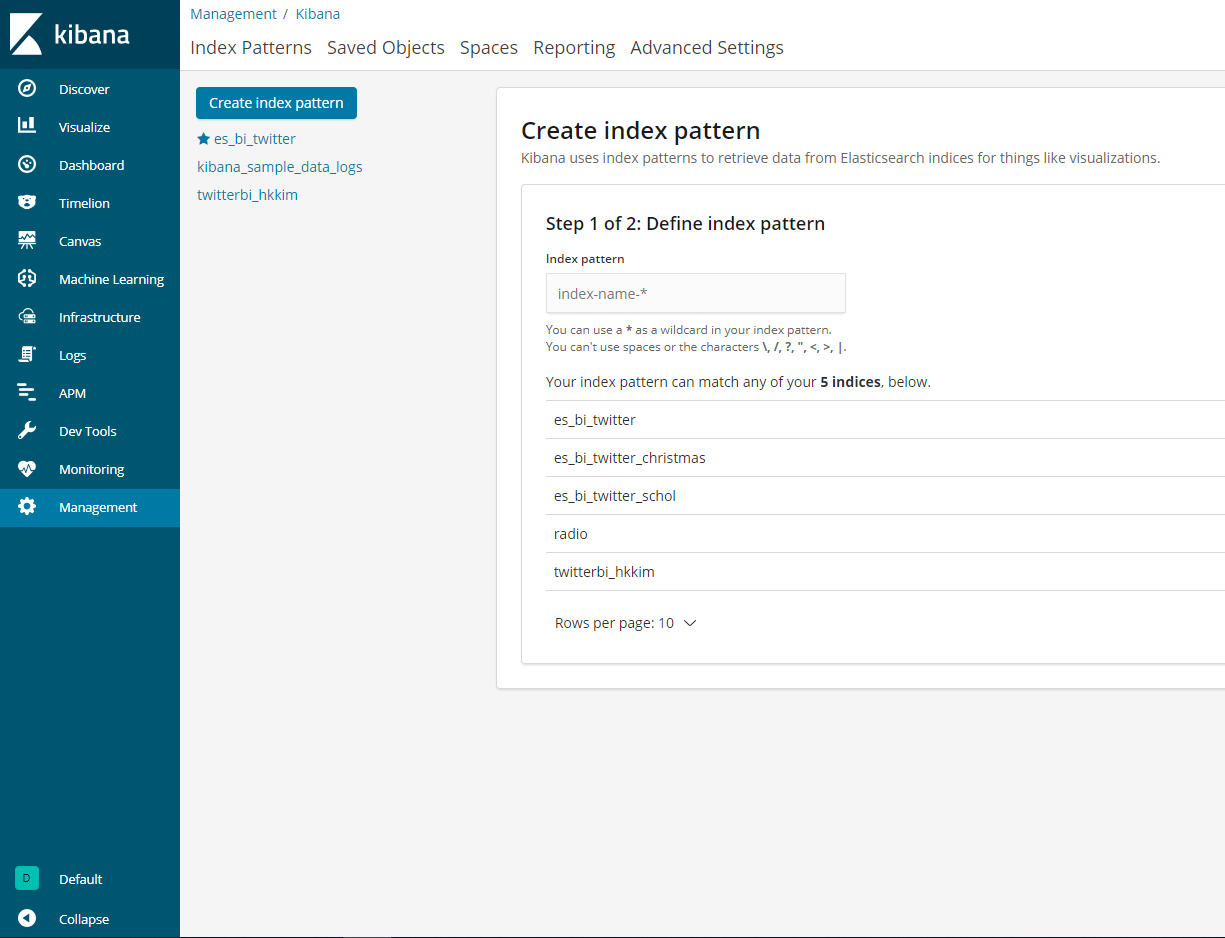
sentiment int,

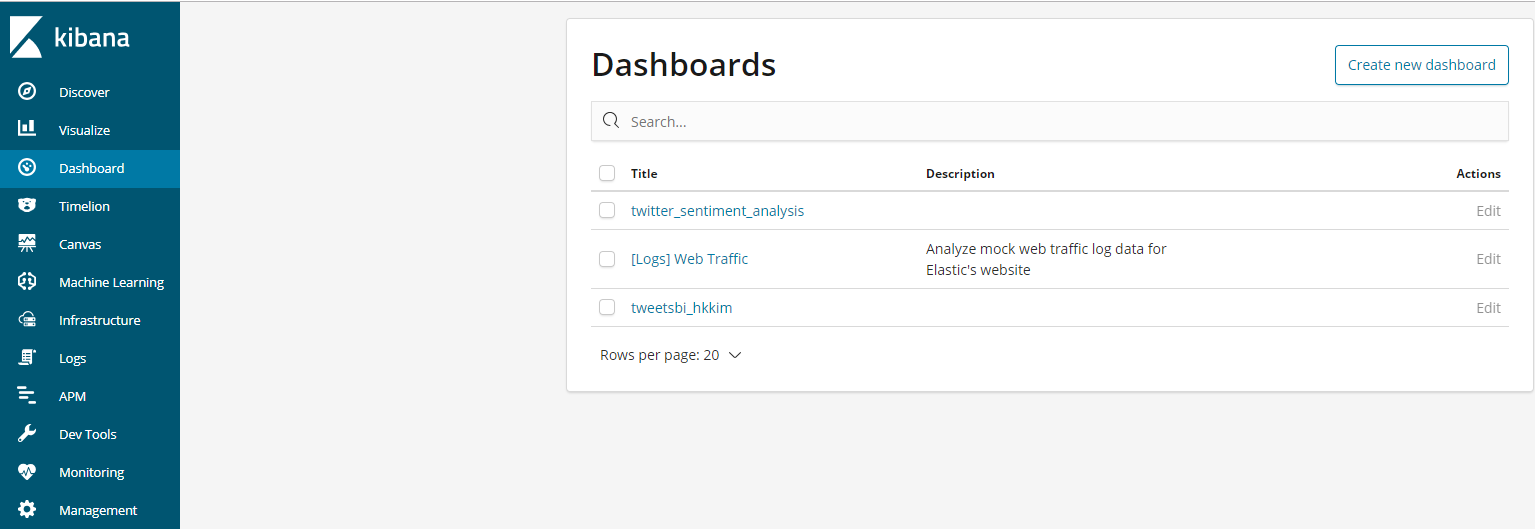
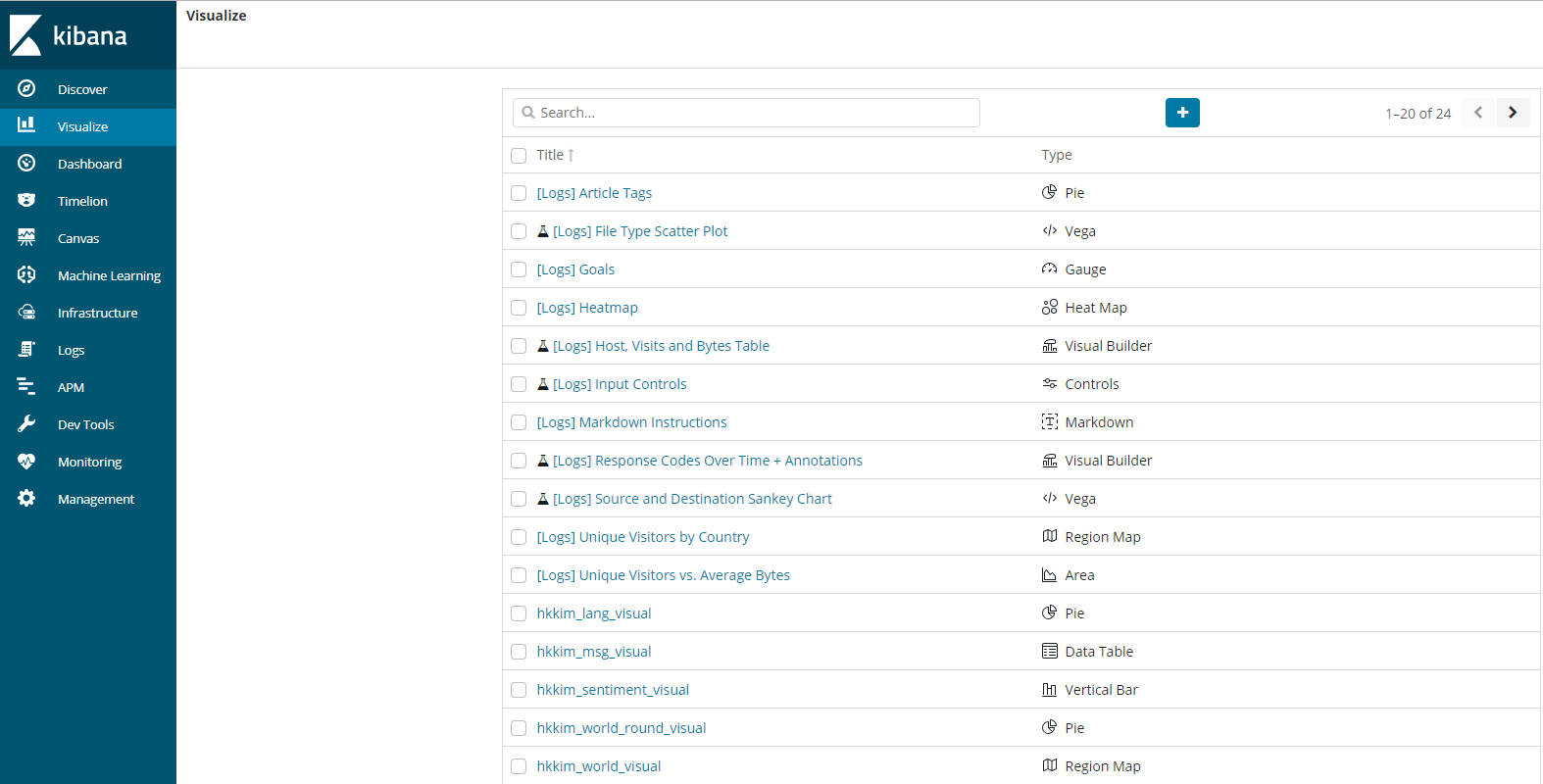
lang string

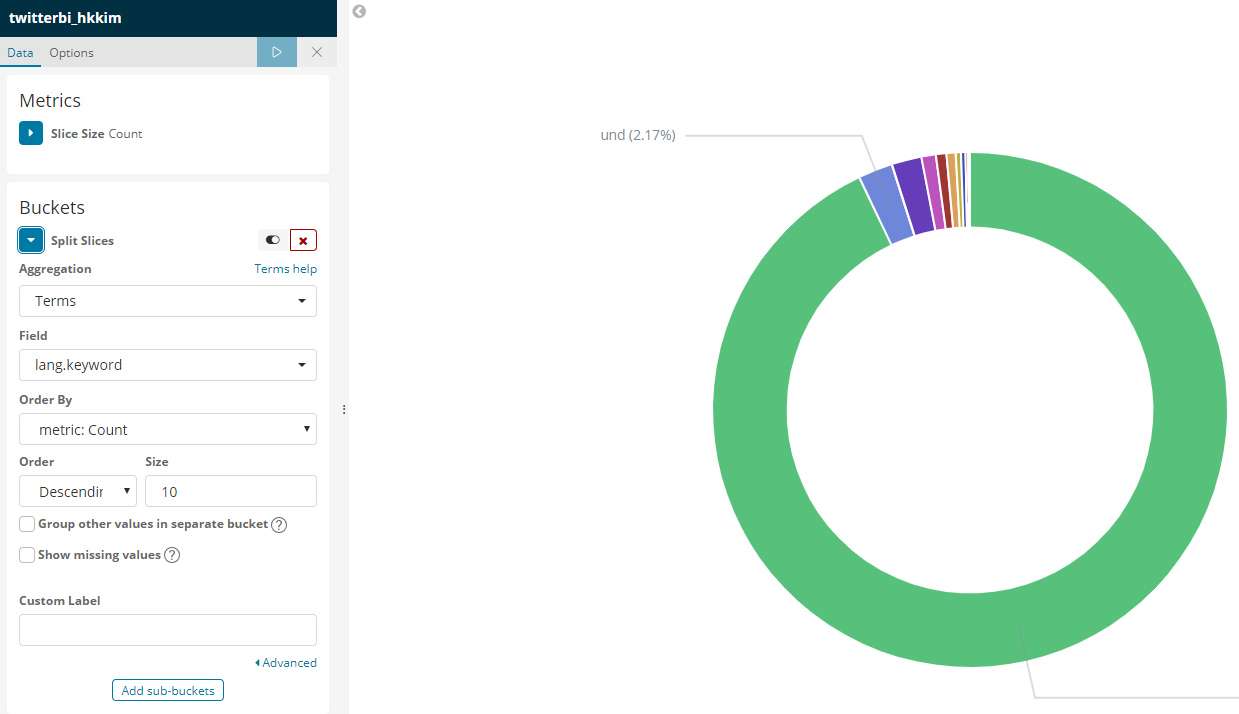
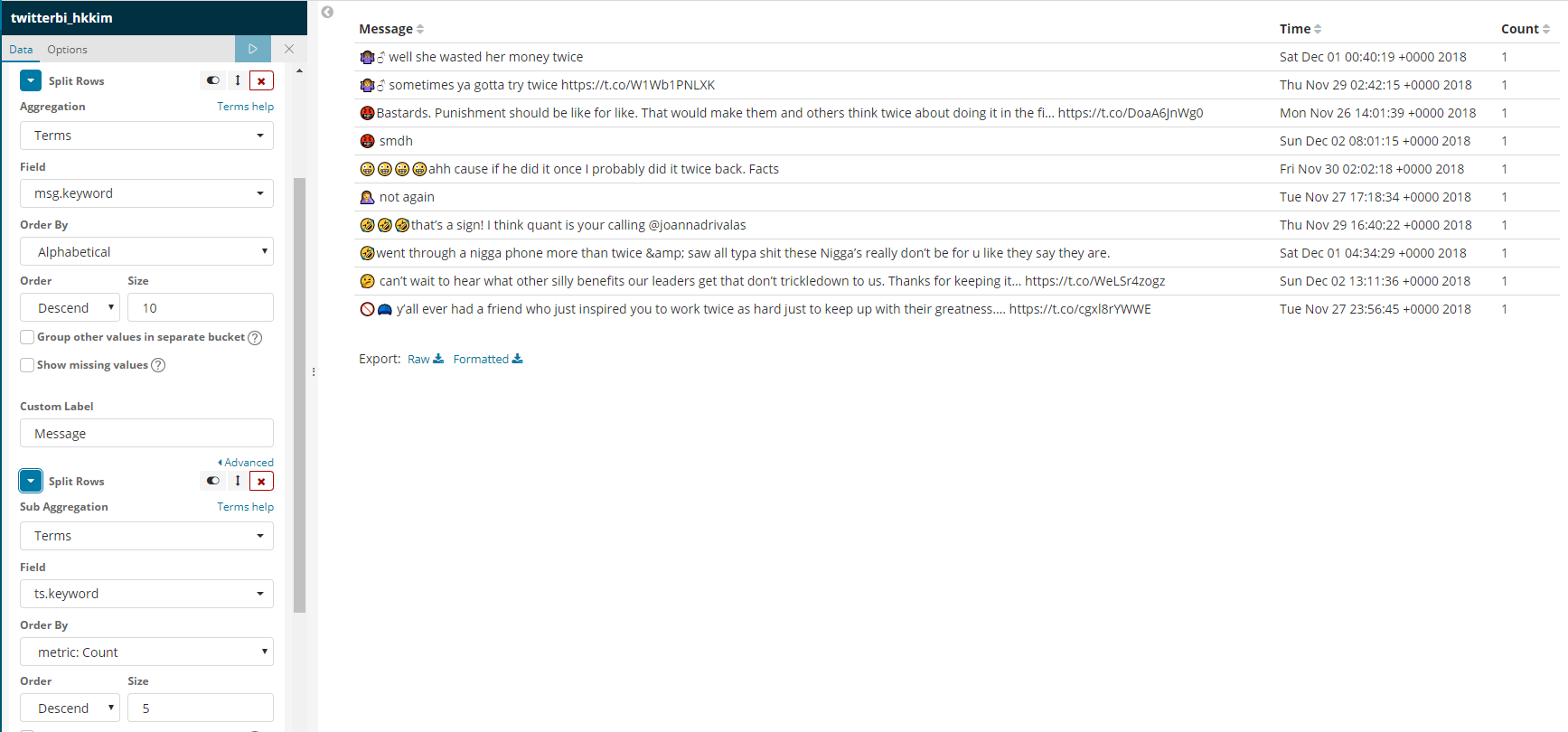
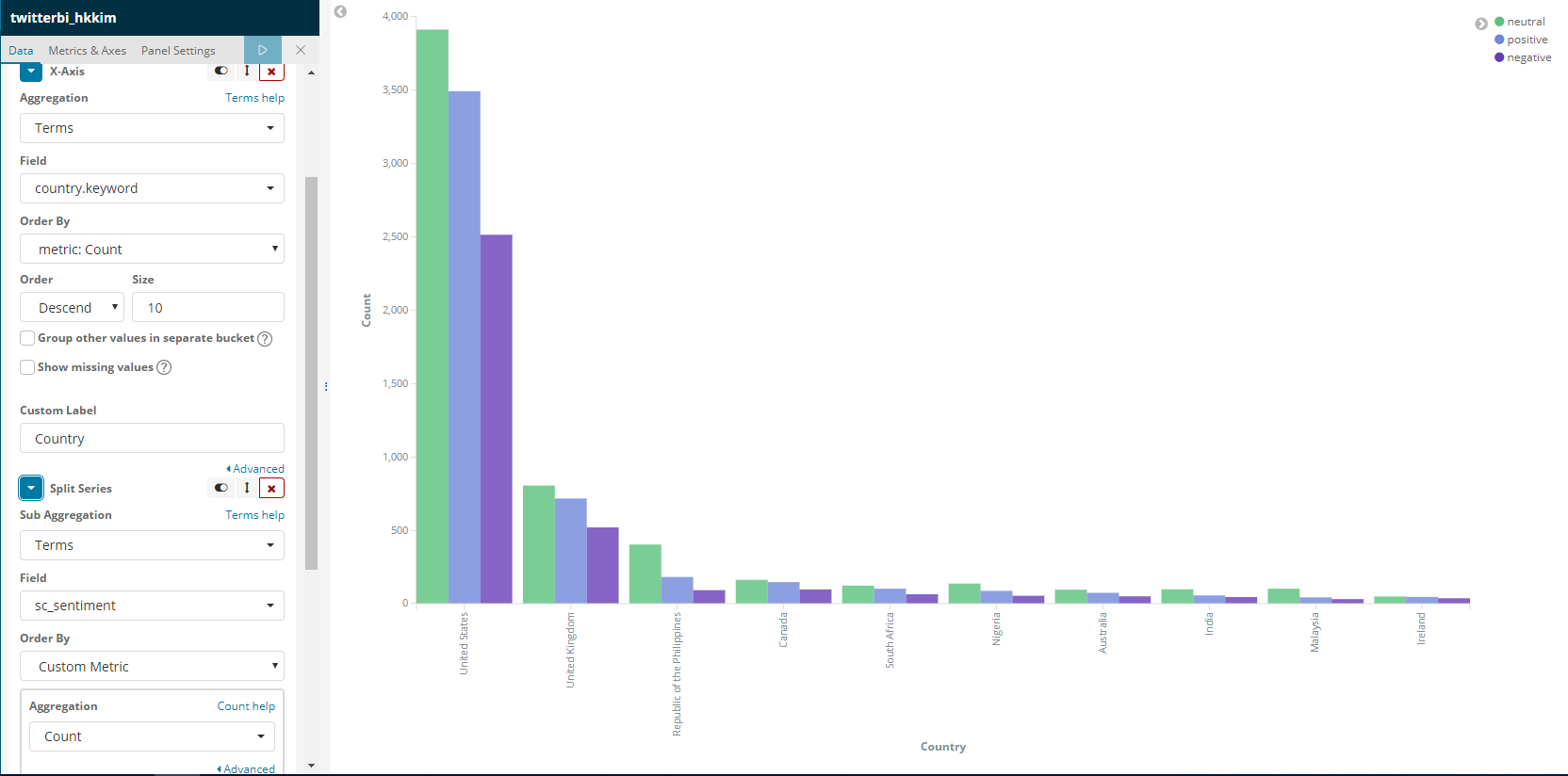
)

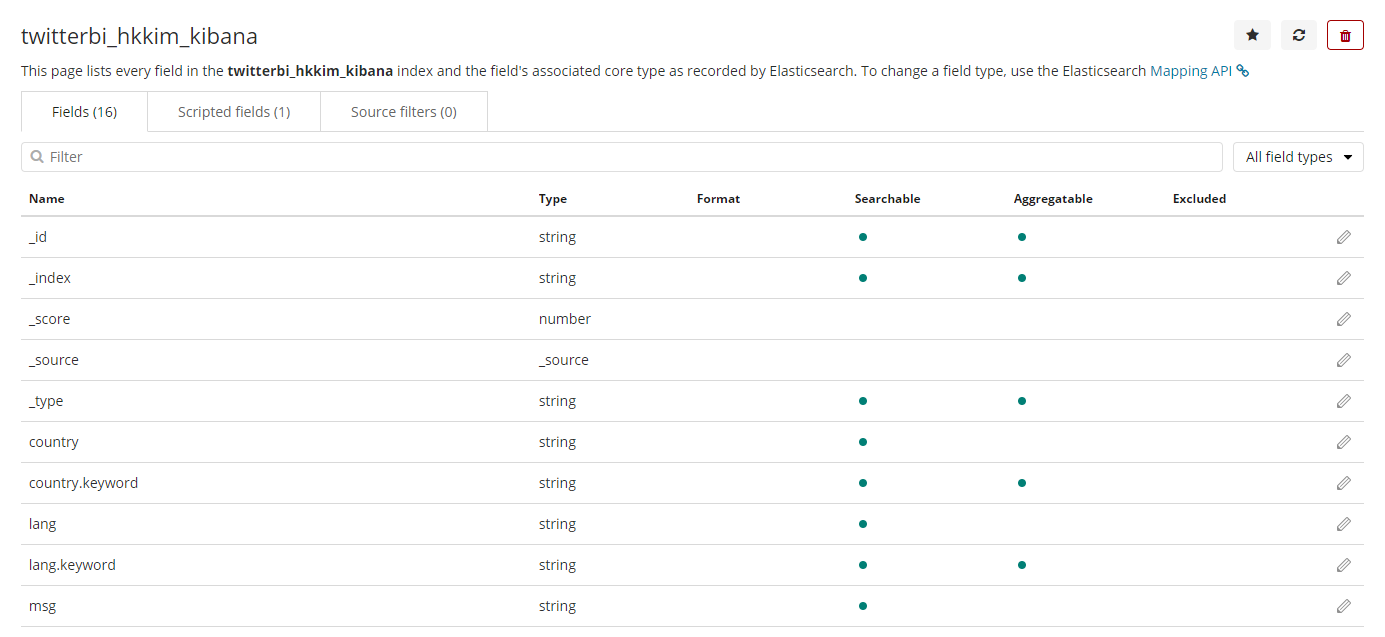
STORED BY 'org.elasticsearch.hadoop.hive.EsStorageHandler'

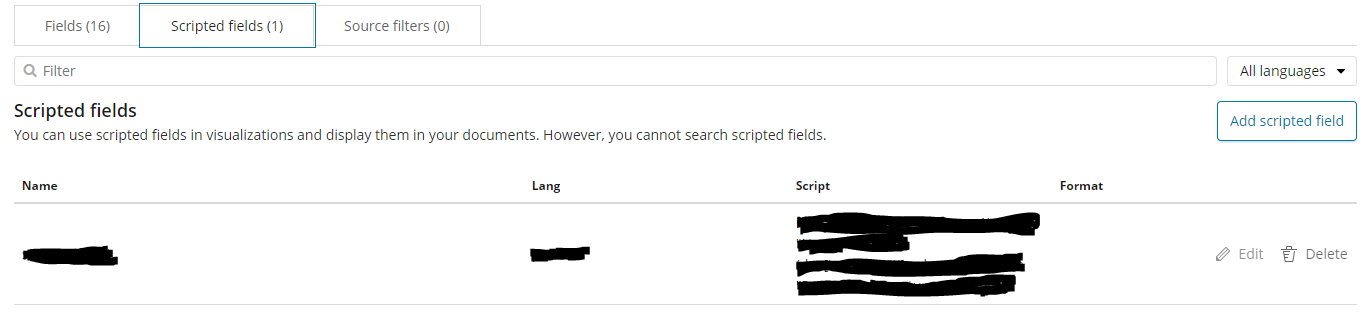
TBLPROPERTIES('es.resource' = 'twitterbi\_hkkim\_kibana/doc','es.nodes'='10.100.3.210');  
INSERT OVERWRITE TABLE twitterbi\_hkkim\_kibana SELECT \* FROM tweetsbi\_new;

**10.100.3.210:5601 Kibana로 접속**

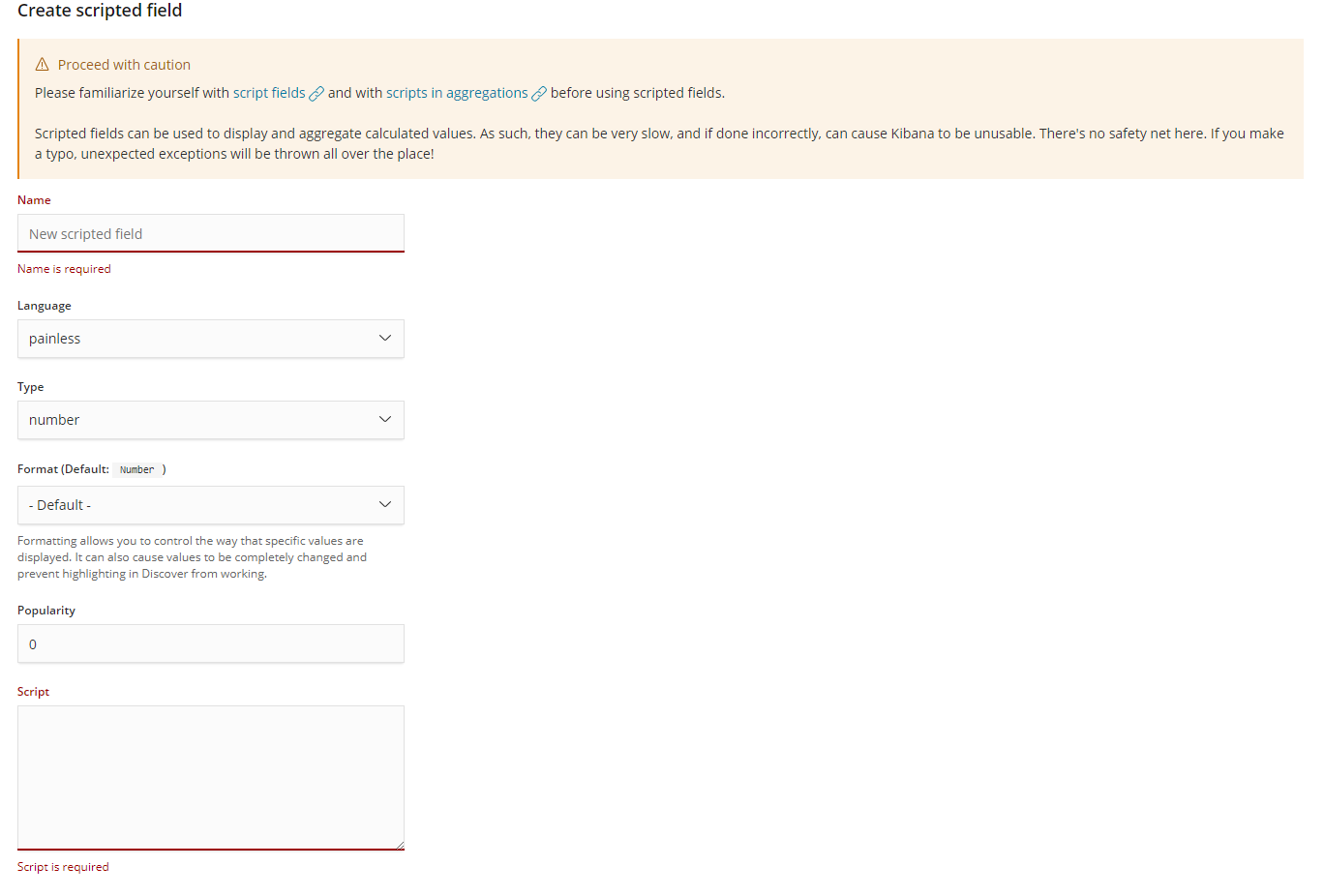
1. **Dashboard 생성  
   “tweetsbi\_hkkim”이란 이름의 Dashboard 생성**
2. **Dashboard에 들어갈 Visualize(그래프) 생성**

**hkkim\_lang\_visual : 언어별 사용자들에 대한 통계를 원그래프로 표현**  
  
  
**hkkim\_msg\_visual : 사용자들이 작성한 트위터 메시지들을 표시**  
  
  
  
  
  
  
  
  
  
**hkkim\_sentiment\_visual : 사용자들의 긍정, 중립, 부정의 정도를 표현**  


*(+ value명을 변경하는 법.)*  
*(1) Management -> twitterbi\_hkkim\_kibana -> Scripted fields -> Add scripted field*



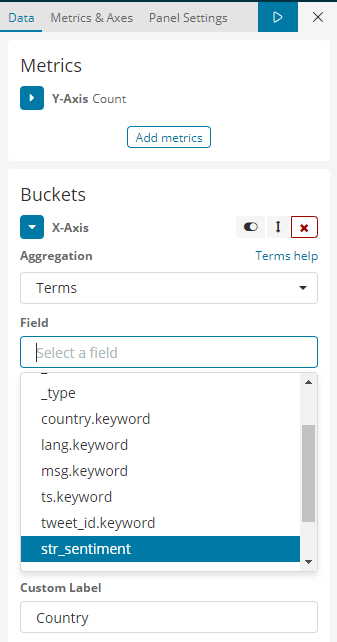
*(2) Script 작성*

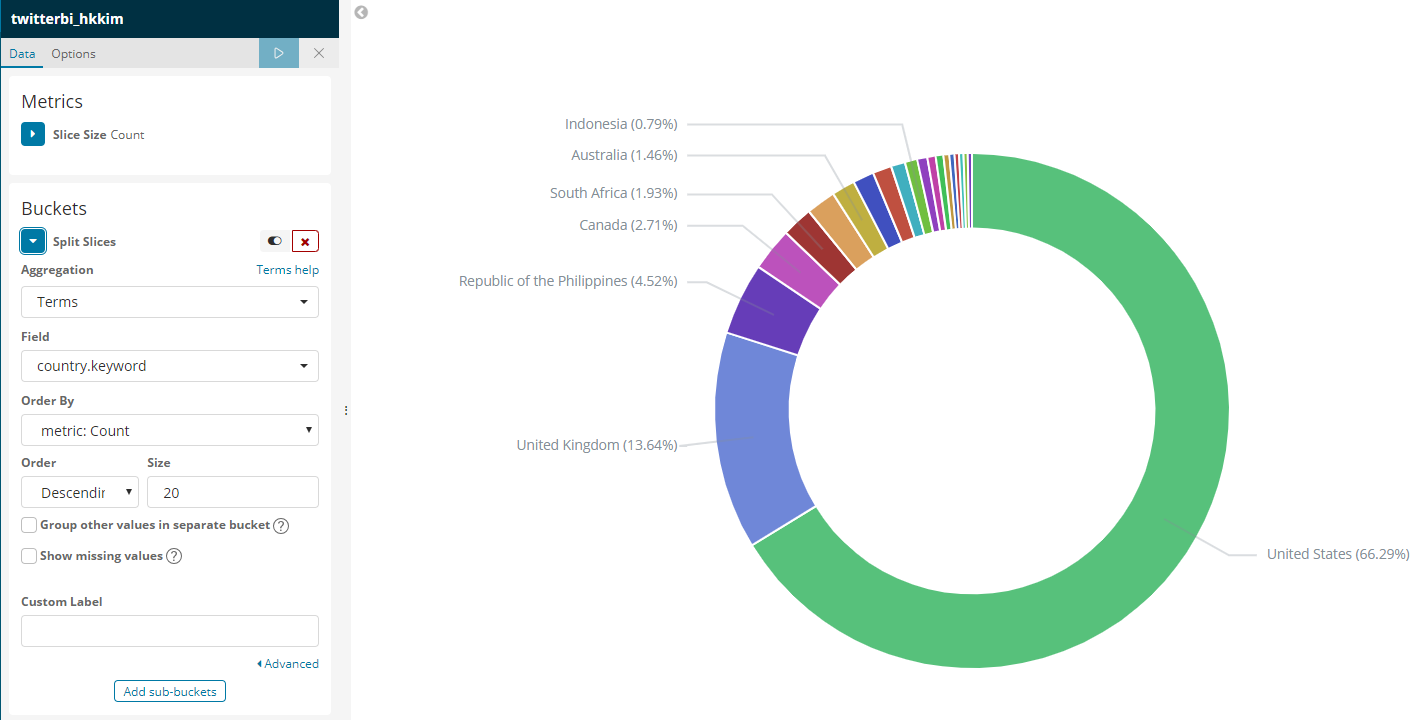
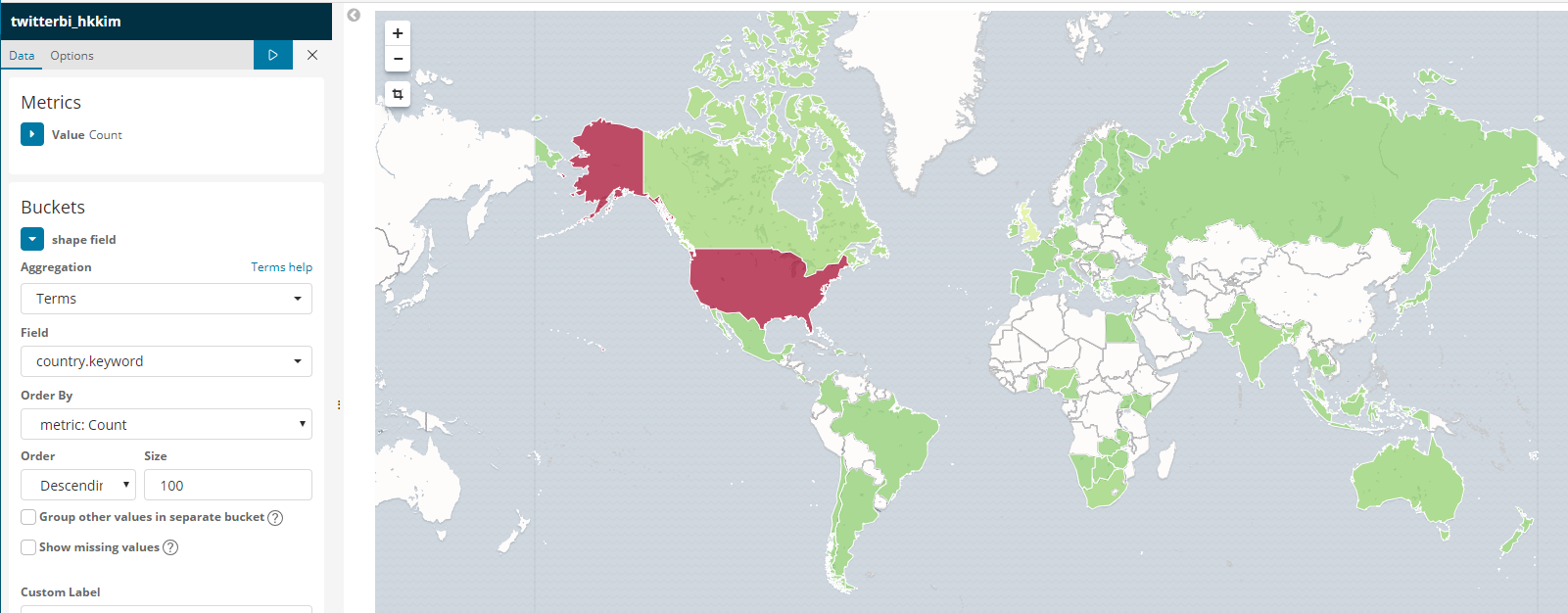


**Name :** scripted field 이름 짓기 *ex) str\_sentiment*  
**Language :** Painless, Expression 2개가 있는데, default는 Painless.   
필자는 Painless로 설정되어 있는 것으로 바로 진행.  
Type : 잘 모르겠음.  
Format : 잘 모르겠음. ( Type이 바뀌면 format의 default값 또한 변경된다. )  
Popularity : 잘 모르겠음.  
**Script :** Script내용 작성  
( ex) sentiment field는 중립, 긍정, 부정을 ‘1’, ‘2’, ’0’으로 표현. 그러나 그 표현방식을 “Neutral”, ”Positive”, “Negative”로 바꾸어보는 script. )

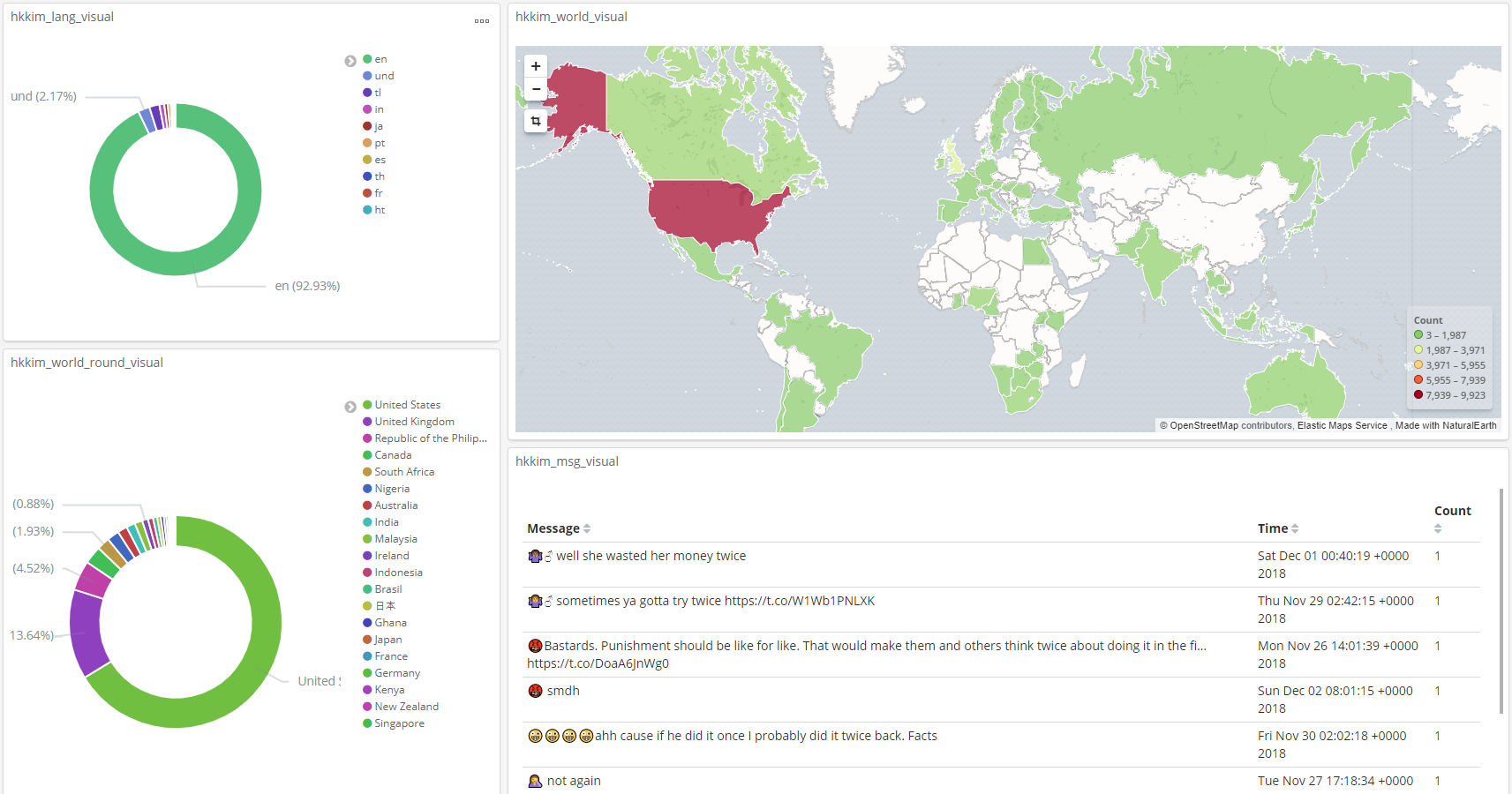
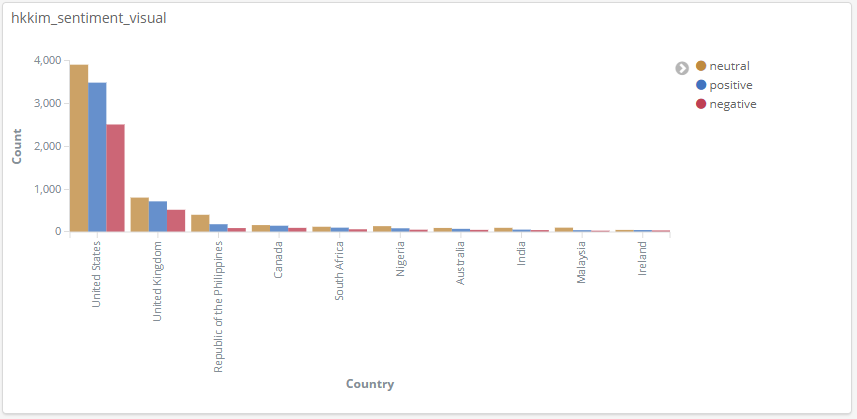
**꼭 알아야할 점** : *doc[‘field\_name’].value*을 통해 해당 field에 있는 value들에게 access할 수 있다.

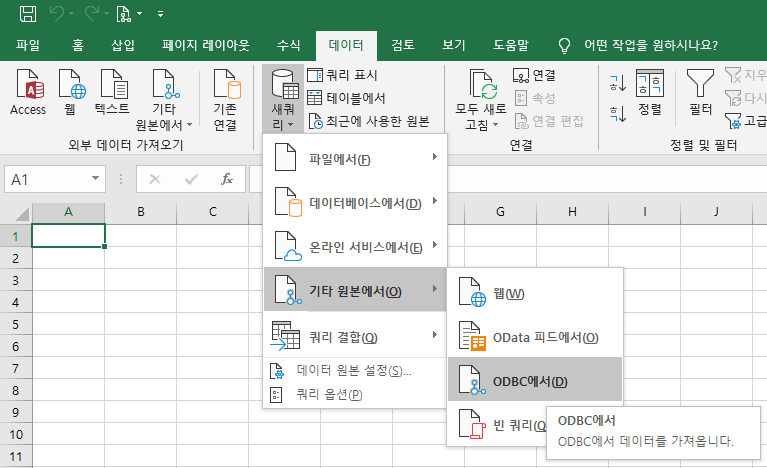
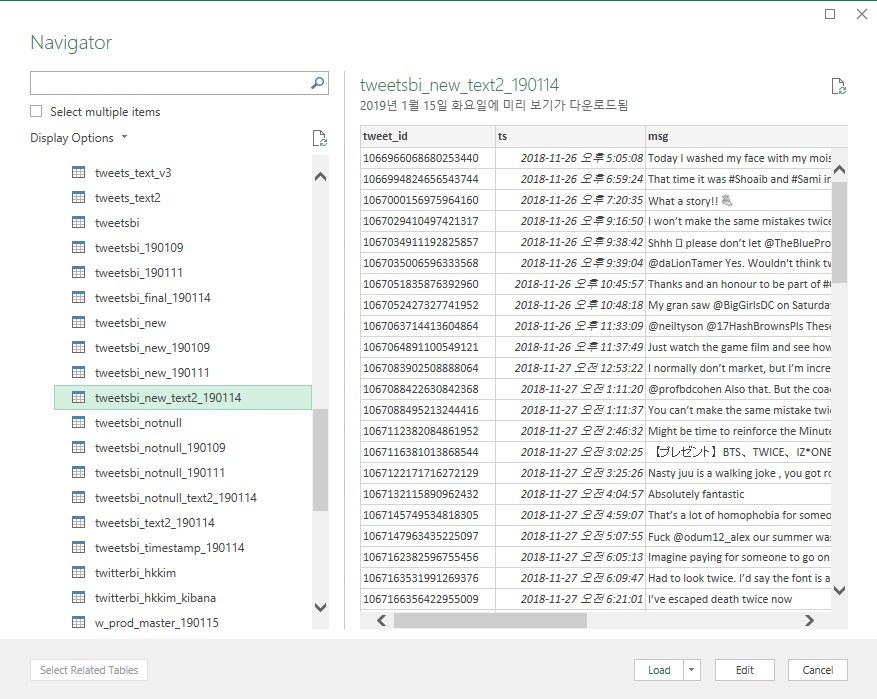
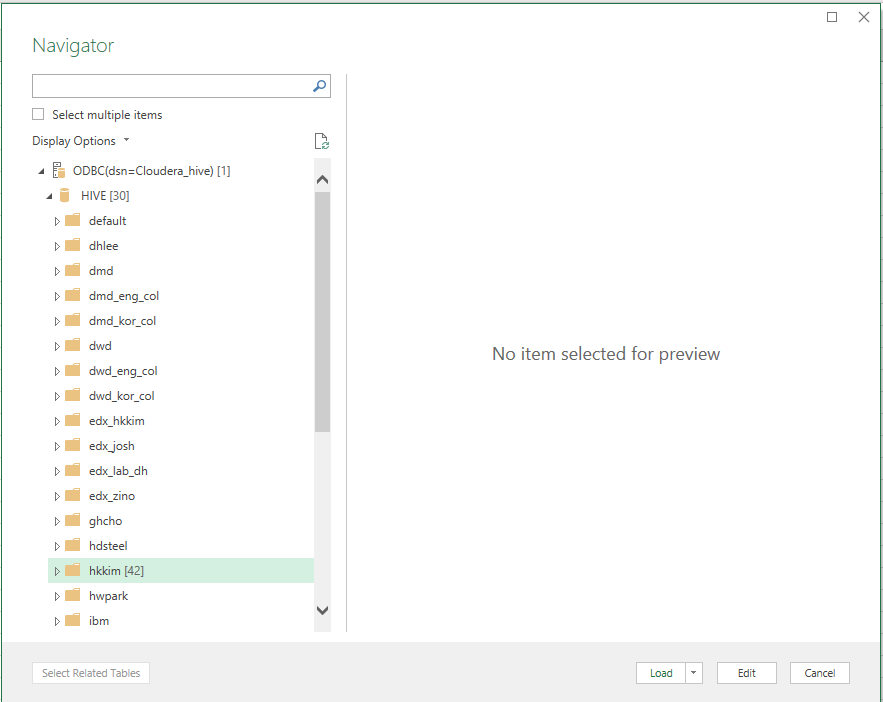
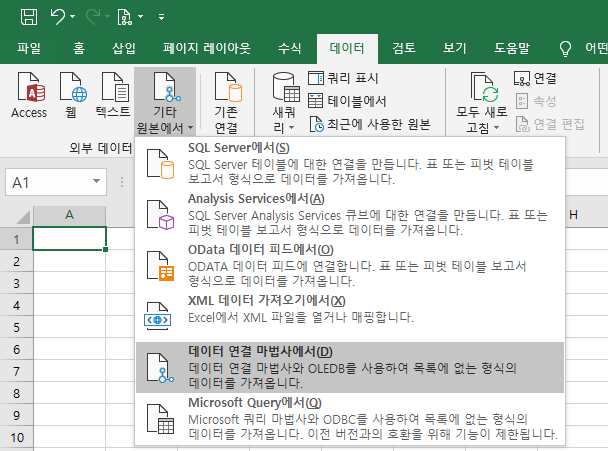
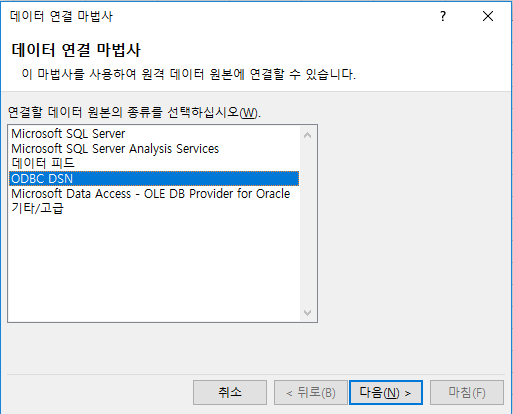
*if (doc['sentiment'].value == 1) return "Neutral";   
else if (doc['sentiment'].value == 2) return "Positive";   
else return "Negative";*

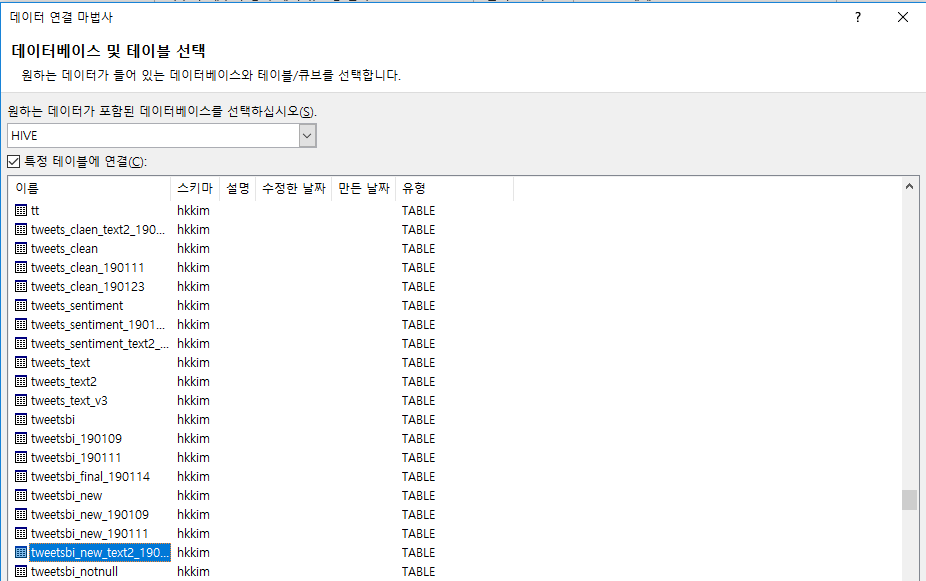
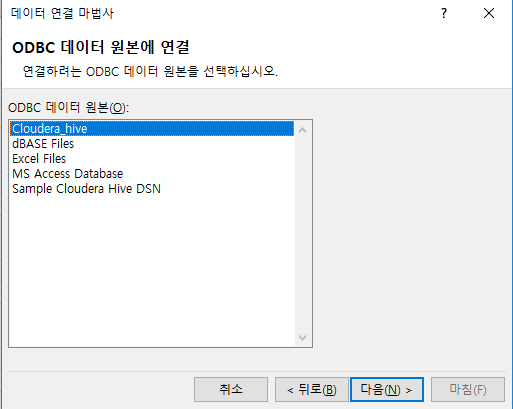


위 과정을 거쳐서 “str\_sentiment”란 이름을 가진 scripted field를 새로 생성하였다.  
  
  
  
  
  
  
  
  
  
**hkkim\_world\_round\_visual : 나라별 사용자들에 대한 통계를 원그래프로 표현**  
  
  
**hkkim\_world\_visual : 나라별 사용자들에 대한 통계를 세계지도에서 표현**  


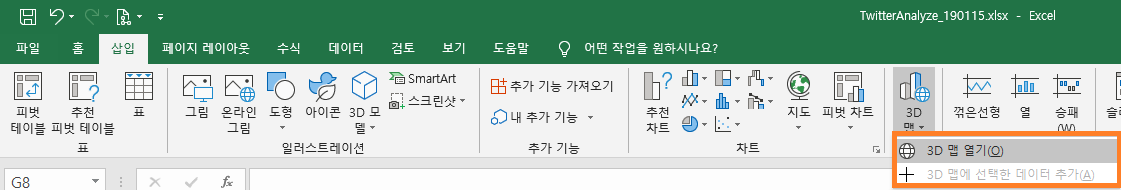
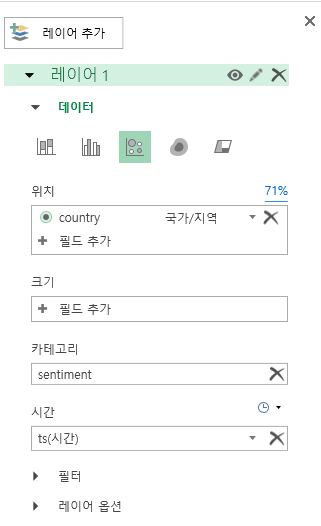
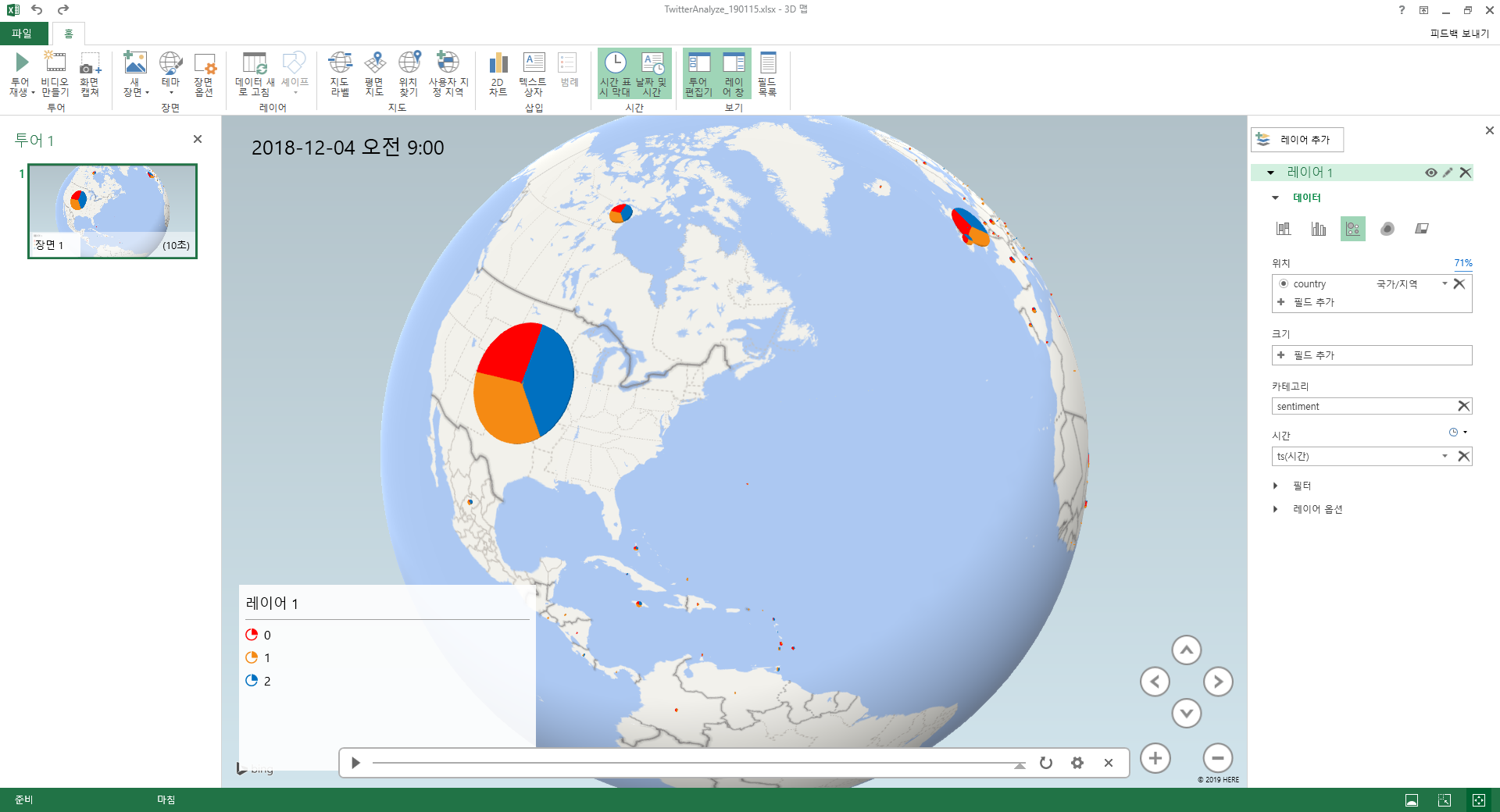
(Dashboard에 적용된 모든 그래프들)

1. **Excel 3D Map 사용  
   (1) Hive Table 가져오기 : “데이터 탭” -> “새 쿼리” -> “기타 원본에서” - > “ODBC에서”  
   ** **(2) Hive Table 가져오기 : “데이터 탭” -> “기타 원본에서” - > “데이터 연결 마법사에서”  
   **

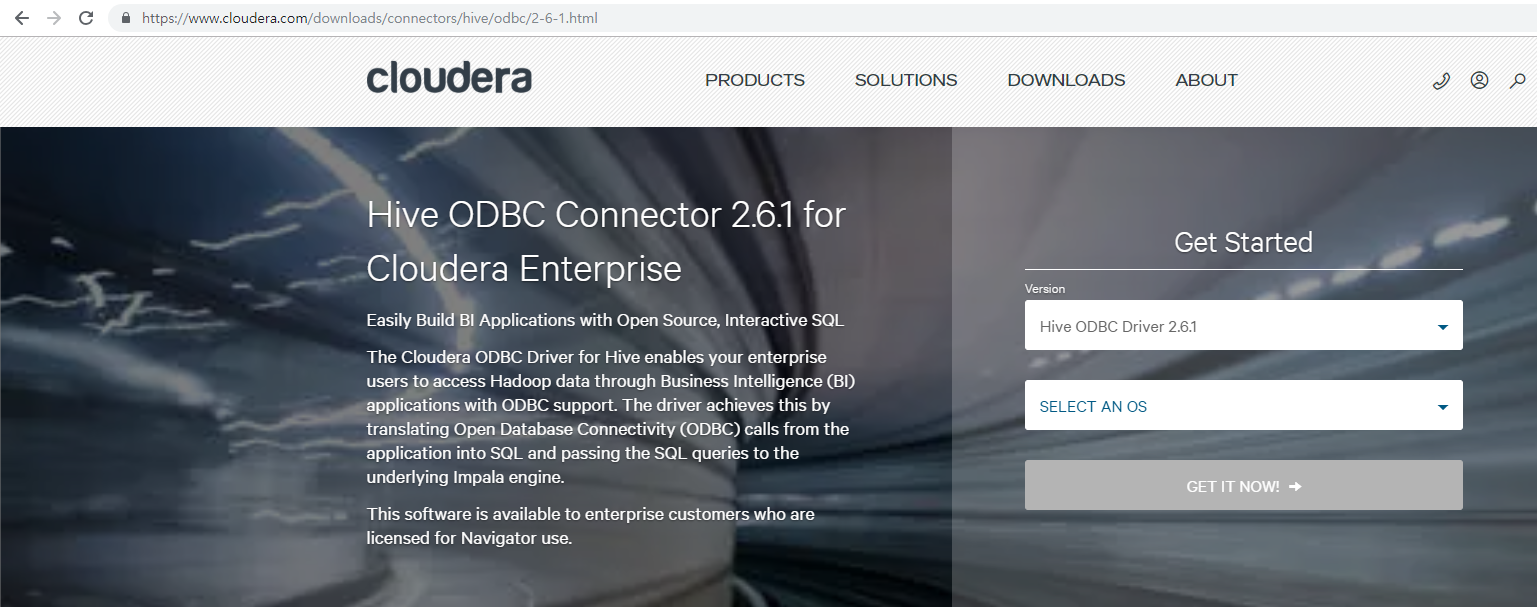


**결과 : 데이터를 Excel에 Import 했을 시의 모습**

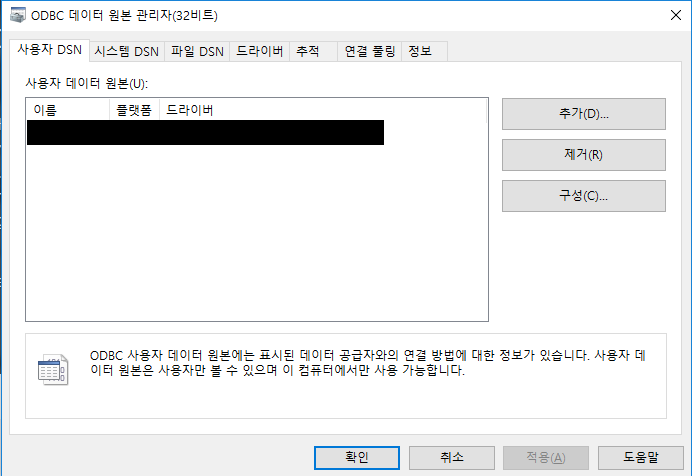
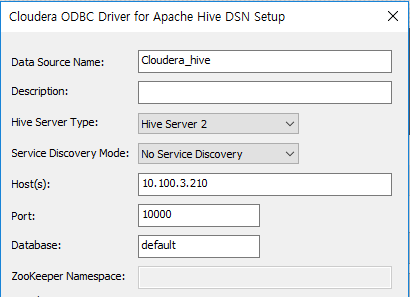
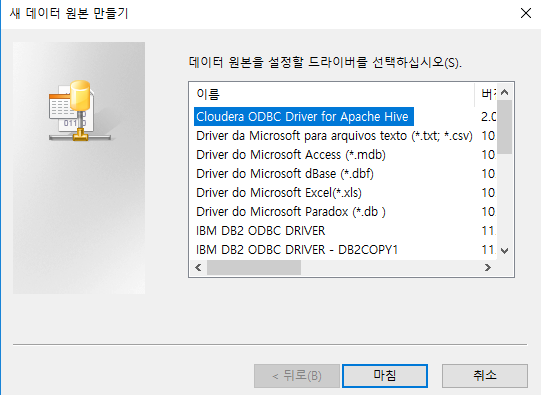
**(3) Excel 3D Map 사용  
** **위치 : 각 나라별 위치를 뜻한다,  
sentiment : 각 나라별로 응답의 긍정,중립,부정의 정도를 나타낸다.  
ts : 시간의 흐름에 따라 각 나라별 sentiment의 정도의 변화를 볼 수 있다.**

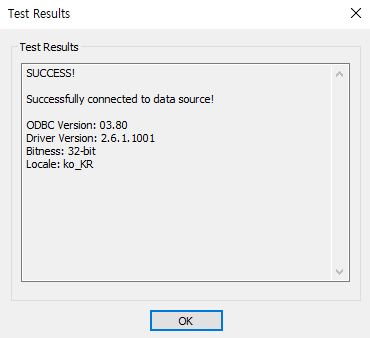
**참고 : https://github.com/nberah/tutorials/blob/hdp/tutorials/hortonworks/analyzing-social-media-sentiment-data/tutorial.md**

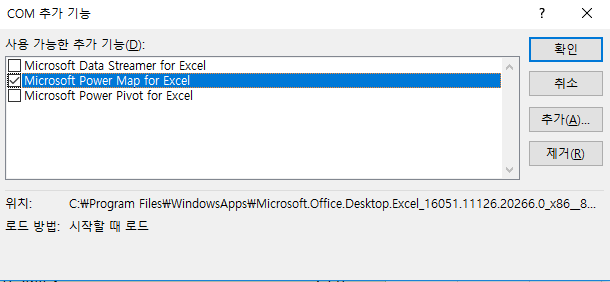
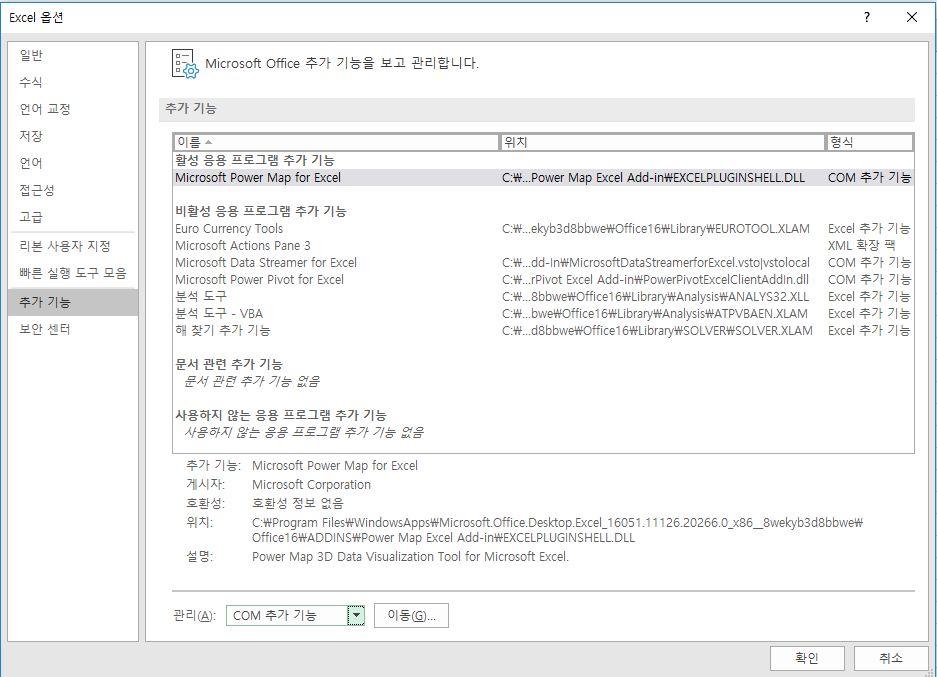
**+ 부록1 : Cloudera ODBC Connector 설치하는 방법**

**다운로드 경로 :** [**https://www.cloudera.com/downloads/connectors/hive/odbc/2-6-1.html**](https://www.cloudera.com/downloads/connectors/hive/odbc/2-6-1.html) **(32bit, 64bit 2가지가 있는데 본인에게 알맞은 bit로 선택한다.   
작성자의 경우에는 32bit 버전을 다운받았다.)  
  
(1) 다운을 받아 ClouderaHiveODBC32(2.6).msi 파일을 열어 설치 진행.**

**(2) “시작” -> “ODBC 데이터 원본 (32비트)”**

**(3) 아래와 같이 진행  
 **

**테스트 결과가 아래와 같이 나오면 연결 성공.  
Excel “데이터” 탭에서 Hive테이블로 연결하고 싶을 시, Cloudera Hive Driver가 목록에 뜬다.**

**+ 부록2 : Excel 3D Map 활성화  
“파일” -> “옵션” 을 클릭하면 아래와 같이 나온다.**

**Reference : https://github.com/nberah/tutorials/blob/hdp/tutorials/hortonworks/analyzing-social-media-sentiment-data/tutorial.md**