# [https://avatars2.githubusercontent.com/u/4156894?v=3&s=100](http://www.calstatela.edu/centers/hipic) CIS5200 Term Project Tutorial

#### Authors: Toney Kim

#### Instructor: [Jongwook Woo](https://www.linkedin.com/in/jongwook-woo-7081a85)

#### Date: 12/16/2018

**Lab Tutorial**

Toney Kim ([tkim69@calstatela.edu](mailto:tkim69@calstatela.edu))

12/16/2018

**Twitter Sentiment Analysis for 2016 Election**

**Objectives**

**List what your objectives are.** In this hands-on lab, you will learn how to:

* Download dataset from Amazon S3
* Put data from Linux to Hadoop
* Use Hive to do Sentiment Analysis
* Get results from Hadoop to Linux
* Visualization

**Platform Spec**

* Amazon Elastic Map Reduce (EMR)
* CPU Speed: 2.8GHZ
* # of CPU cores: 3
* # of nodes: 3
* Total Memory Size: 15GB

Step 1: Download dataset from Amazon S3

This step is to download the Twitter data to the Linux file system.

1. SSH and sign into your Amazon Elastic MapReduce (EMR) master node.
2. Enter commands in linux terminal

wget https://s3.amazonaws.com/vm-election10282016-2/tweets10302016.tgz

wget -O dictionary.tsv https://s3.amazonaws.com/hipicdatasets/dictionary.tsv

tar -xzvf tweets10302016.tgz

Step 2: Put data to Hadoop

This step is to make file structure and folders in Hadoop and put data into Hadoop

* 1. Make folders in Hadoop

hdfs dfs -mkdir p3

hdfs dfs -mkdir p3/a100

hdfs dfs -mkdir p3/dictionary

hdfs dfs -mkdir p3/zones

hdfs dfs -mkdir p3/zones/p

hdfs dfs -mkdir p3/zones/p/t

hdfs dfs -mkdir p3/zones/p/t/pr

hdfs dfs -mkdir p3/zones/p/t/nr

hdfs dfs -mkdir p3/zones/p/h

hdfs dfs -mkdir p3/zones/p/h/pr

hdfs dfs -mkdir p3/zones/p/h/nr

hdfs dfs -mkdir p3/zones/m

hdfs dfs -mkdir p3/zones/m/t

hdfs dfs -mkdir p3/zones/m/t/pr

hdfs dfs -mkdir p3/zones/m/t/nr

hdfs dfs -mkdir p3/zones/m/h

hdfs dfs -mkdir p3/zones/m/h/pr

hdfs dfs -mkdir p3/zones/m/h/nr

hdfs dfs -mkdir p3/zones/c

hdfs dfs -mkdir p3/zones/c/t

hdfs dfs -mkdir p3/zones/c/t/pr

hdfs dfs -mkdir p3/zones/c/t/nr

hdfs dfs -mkdir p3/zones/c/h

hdfs dfs -mkdir p3/zones/c/h/pr

hdfs dfs -mkdir p3/zones/c/h/nr

hdfs dfs -mkdir p3/zones/e

hdfs dfs -mkdir p3/zones/e/t

hdfs dfs -mkdir p3/zones/e/t/pr

hdfs dfs -mkdir p3/zones/e/t/nr

hdfs dfs -mkdir p3/zones/e/h

hdfs dfs -mkdir p3/zones/e/h/pr

hdfs dfs -mkdir p3/zones/e/h/nr

* 1. Copy dataset to Hadoop

hdfs dfs –put dictionary.tsv p3/dictionary

cd tweets10302016

hdfs dfs –put \* p3/a100

Step 3: Sentiment Analysis in Hive

This step is to do sentiment analysis in Hive

1. Start and enter Hive

hive

1. Create Database

CREATE DATABASE p3;

use p3;

CREATE EXTERNAL TABLE if not exists runone(

idone BIGINT,

idtwo BIGINT,

created STRING,

language STRING,

avatar STRING,

country STRING,

title STRING,

first\_struct STRING

)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY '|'

STORED AS TEXTFILE

LOCATION '/user/hadoop/p3/a100';

1. Create dictionary table

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/hadoop/p3/dictionary';

1. Store separate data by timezones

INSERT OVERWRITE DIRECTORY '/user/hadoop/p3/zones/p' ROW FORMAT DELIMITED FIELDS TERMINATED BY ':' SELECT idone, LOWER(title) FROM runone WHERE country LIKE "%Pacific%";

INSERT OVERWRITE DIRECTORY '/user/hadoop/p3/zones/m' ROW FORMAT DELIMITED FIELDS TERMINATED BY ':' SELECT idone, LOWER(title) FROM runone WHERE country LIKE "%Mountain%";

INSERT OVERWRITE DIRECTORY '/user/hadoop/p3/zones/c' ROW FORMAT DELIMITED FIELDS TERMINATED BY ':' SELECT idone, LOWER(title) FROM runone WHERE country LIKE "%Central%";

INSERT OVERWRITE DIRECTORY '/user/hadoop/p3/zones/e' ROW FORMAT DELIMITED FIELDS TERMINATED BY ':' SELECT idone, LOWER(title) FROM runone WHERE country LIKE "%Eastern%";

1. Create tables for timezones

CREATE EXTERNAL TABLE if not exists pzone(

idone BIGINT,

avatar STRING,

tweet STRING

)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ':'

STORED AS TEXTFILE

LOCATION '/user/hadoop/p3/zones/p';

CREATE EXTERNAL TABLE if not exists mzone(

idone BIGINT,

avatar STRING,

tweet STRING

)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ':'

STORED AS TEXTFILE

LOCATION '/user/hadoop/p3/zones/m';

CREATE EXTERNAL TABLE if not exists czone(

idone BIGINT,

avatar STRING,

tweet STRING

)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ':'

STORED AS TEXTFILE

LOCATION '/user/hadoop/p3/zones/c';

CREATE EXTERNAL TABLE if not exists ezone(

idone BIGINT,

avatar STRING,

tweet STRING

)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ':'

STORED AS TEXTFILE

LOCATION '/user/hadoop/p3/zones/e';

1. Zone Analysis Segmentation

INSERT OVERWRITE DIRECTORY '/user/hadoop/p3/zones/p/t' ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' SELECT idone, tweet FROM pzone WHERE tweet LIKE "%trump%" and tweet NOT LIKE "%hillary%";

INSERT OVERWRITE DIRECTORY '/user/hadoop/p3/zones/p/h' ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' SELECT idone, tweet FROM pzone WHERE tweet LIKE "%hillary%" and tweet NOT LIKE "%trump%";

INSERT OVERWRITE DIRECTORY '/user/hadoop/p3/zones/m/t' ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' SELECT idone, tweet FROM mzone WHERE tweet LIKE "%trump%" and tweet NOT LIKE "%hillary%";

INSERT OVERWRITE DIRECTORY '/user/hadoop/p3/zones/m/h' ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' SELECT idone, tweet FROM mzone WHERE tweet LIKE "%hillary%" and tweet NOT LIKE "%trump%";

INSERT OVERWRITE DIRECTORY '/user/hadoop/p3/zones/c/t' ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' SELECT idone, tweet FROM czone WHERE tweet LIKE "%trump%" and tweet NOT LIKE "%hillary%";

INSERT OVERWRITE DIRECTORY '/user/hadoop/p3/zones/c/h' ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' SELECT idone, tweet FROM czone WHERE tweet LIKE "%hillary%" and tweet NOT LIKE "%trump%";

INSERT OVERWRITE DIRECTORY '/user/hadoop/p3/zones/e/t' ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' SELECT idone, tweet FROM ezone WHERE tweet LIKE "%trump%" and tweet NOT LIKE "%hillary%";

INSERT OVERWRITE DIRECTORY '/user/hadoop/p3/zones/e/h' ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' SELECT idone, tweet FROM ezone WHERE tweet LIKE "%hillary%" and tweet NOT LIKE "%trump%";

1. Create Trump and Hillary Zone Tables

CREATE EXTERNAL TABLE if not exists pzonet(

idone BIGINT,

tweet STRING

)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

STORED AS TEXTFILE

LOCATION '/user/hadoop/p3/zones/p/t';

CREATE EXTERNAL TABLE if not exists pzoneh(

idone BIGINT,

tweet STRING

)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

STORED AS TEXTFILE

LOCATION '/user/hadoop/p3/zones/p/h';

CREATE EXTERNAL TABLE if not exists mzonet(

idone BIGINT,

tweet STRING

)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

STORED AS TEXTFILE

LOCATION '/user/hadoop/p3/zones/m/t';

CREATE EXTERNAL TABLE if not exists mzoneh(

idone BIGINT,

tweet STRING

)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

STORED AS TEXTFILE

LOCATION '/user/hadoop/p3/zones/m/h';

CREATE EXTERNAL TABLE if not exists czonet(

idone BIGINT,

tweet STRING

)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

STORED AS TEXTFILE

LOCATION '/user/hadoop/p3/zones/c/t';

CREATE EXTERNAL TABLE if not exists czoneh(

idone BIGINT,

tweet STRING

)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

STORED AS TEXTFILE

LOCATION '/user/hadoop/p3/zones/c/h';

CREATE EXTERNAL TABLE if not exists ezonet(

idone BIGINT,

tweet STRING

)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

STORED AS TEXTFILE

LOCATION '/user/hadoop/p3/zones/e/t';

CREATE EXTERNAL TABLE if not exists ezoneh(

idone BIGINT,

tweet STRING

)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

STORED AS TEXTFILE

LOCATION '/user/hadoop/p3/zones/e/h';

1. Positive/Negative Sentiment Analysis for Pacific Trump

create view IF NOT EXISTS p11 as

select idone, message

from pzonet

LATERAL VIEW explode(sentences(lower(tweet))) dummy as message;

create view IF NOT EXISTS p12 as

SELECT idone, msg

FROM p11

lateral view explode(message) dummy as msg;

create view IF NOT EXISTS p13 as SELECT

idone tweet\_id, p12.msg,

case d.polarity

when 'negative' then -1

when 'positive' then 1

else 0 end as polarity

FROM p12 left outer join dictionary d on p12.msg = d.word;

create table IF NOT EXISTS ptsentiment

stored as orc as select

tweet\_id,

case

when sum( polarity ) > 0 then 'positive'

when sum( polarity ) < 0 then 'negative'

else 'neutral' end as sentiment

from p13 group by tweet\_id;

1. Count total positive and negative

INSERT OVERWRITE DIRECTORY '/user/hadoop/p3/zones/p/t/pr' ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' SELECT COUNT(sentiment) FROM ptsentiment WHERE sentiment LIKE "%positive%";

INSERT OVERWRITE DIRECTORY '/user/hadoop/p3/zones/p/t/nr' ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' SELECT COUNT(sentiment) FROM ptsentiment WHERE sentiment LIKE "%negative%";

1. Positive/Negative Sentiment Analysis for Pacific Hillary

create view IF NOT EXISTS p21 as

select idone, message

from pzoneh

LATERAL VIEW explode(sentences(lower(tweet))) dummy as message;

create view IF NOT EXISTS p22 as

SELECT idone, msg

FROM p21

lateral view explode(message) dummy as msg;

create view IF NOT EXISTS p23 as SELECT

idone tweet\_id, p22.msg,

case d.polarity

when 'negative' then -1

when 'positive' then 1

else 0 end as polarity

FROM p22 left outer join dictionary d on p22.msg = d.word;

create table IF NOT EXISTS phsentiment

stored as orc as select

tweet\_id,

case

when sum( polarity ) > 0 then 'positive'

when sum( polarity ) < 0 then 'negative'

else 'neutral' end as sentiment

from p23 group by tweet\_id;

1. Count total positive and negative

INSERT OVERWRITE DIRECTORY '/user/hadoop/p3/zones/p/h/pr' ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' SELECT COUNT(sentiment) FROM phsentiment WHERE sentiment LIKE "%positive%";

INSERT OVERWRITE DIRECTORY '/user/hadoop/p3/zones/p/h/nr' ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' SELECT COUNT(sentiment) FROM phsentiment WHERE sentiment LIKE "%negative%";

1. Positive/Negative Sentiment Analysis for Mountain Trump

create view IF NOT EXISTS m11 as

select idone, message

from mzonet

LATERAL VIEW explode(sentences(lower(tweet))) dummy as message;

create view IF NOT EXISTS m12 as

SELECT idone, msg

FROM m11

lateral view explode(message) dummy as msg;

create view IF NOT EXISTS m13 as SELECT

idone tweet\_id, m12.msg,

case d.polarity

when 'negative' then -1

when 'positive' then 1

else 0 end as polarity

FROM m12 left outer join dictionary d on m12.msg = d.word;

create table IF NOT EXISTS mtsentiment

stored as orc as select

tweet\_id,

case

when sum( polarity ) > 0 then 'positive'

when sum( polarity ) < 0 then 'negative'

else 'neutral' end as sentiment

from m13 group by tweet\_id;

1. Count total positive and negative

INSERT OVERWRITE DIRECTORY '/user/hadoop/p3/zones/m/t/pr' ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' SELECT COUNT(sentiment) FROM mtsentiment WHERE sentiment LIKE "%positive%";

INSERT OVERWRITE DIRECTORY '/user/hadoop/p3/zones/m/t/nr' ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' SELECT COUNT(sentiment) FROM mtsentiment WHERE sentiment LIKE "%negative%";

1. Positive/Negative Sentiment Analysis for Mountain Hillary

create view IF NOT EXISTS m21 as

select idone, message

from mzoneh

LATERAL VIEW explode(sentences(lower(tweet))) dummy as message;

create view IF NOT EXISTS m22 as

SELECT idone, msg

FROM m21

lateral view explode(message) dummy as msg;

create view IF NOT EXISTS m23 as SELECT

idone tweet\_id, m22.msg,

case d.polarity

when 'negative' then -1

when 'positive' then 1

else 0 end as polarity

FROM m22 left outer join dictionary d on m22.msg = d.word;

create table IF NOT EXISTS mhsentiment

stored as orc as select

tweet\_id,

case

when sum( polarity ) > 0 then 'positive'

when sum( polarity ) < 0 then 'negative'

else 'neutral' end as sentiment

from m23 group by tweet\_id;

1. Count total positive and negative

INSERT OVERWRITE DIRECTORY '/user/hadoop/p3/zones/m/h/pr' ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' SELECT COUNT(sentiment) FROM mhsentiment WHERE sentiment LIKE "%positive%";

INSERT OVERWRITE DIRECTORY '/user/hadoop/p3/zones/m/h/nr' ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' SELECT COUNT(sentiment) FROM mhsentiment WHERE sentiment LIKE "%negative%";

1. Positive/Negative Sentiment Analysis for Central Trump

create view IF NOT EXISTS c11 as

select idone, message

from czonet

LATERAL VIEW explode(sentences(lower(tweet))) dummy as message;

create view IF NOT EXISTS c12 as

SELECT idone, msg

FROM c11

lateral view explode(message) dummy as msg;

create view IF NOT EXISTS c13 as SELECT

idone tweet\_id, c12.msg,

case d.polarity

when 'negative' then -1

when 'positive' then 1

else 0 end as polarity

FROM c12 left outer join dictionary d on c12.msg = d.word;

create table IF NOT EXISTS ctsentiment

stored as orc as select

tweet\_id,

case

when sum( polarity ) > 0 then 'positive'

when sum( polarity ) < 0 then 'negative'

else 'neutral' end as sentiment

from c13 group by tweet\_id;

1. Count total positive and negative

INSERT OVERWRITE DIRECTORY '/user/hadoop/p3/zones/c/t/pr' ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' SELECT COUNT(sentiment) FROM ctsentiment WHERE sentiment LIKE "%positive%";

INSERT OVERWRITE DIRECTORY '/user/hadoop/p3/zones/c/t/nr' ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' SELECT COUNT(sentiment) FROM ctsentiment WHERE sentiment LIKE "%negative%";

1. Positive/Negative Sentiment Analysis for Central Hillary

create view IF NOT EXISTS c21 as

select idone, message

from czoneh

LATERAL VIEW explode(sentences(lower(tweet))) dummy as message;

create view IF NOT EXISTS c22 as

SELECT idone, msg

FROM c21

lateral view explode(message) dummy as msg;

create view IF NOT EXISTS c23 as SELECT

idone tweet\_id, c22.msg,

case d.polarity

when 'negative' then -1

when 'positive' then 1

else 0 end as polarity

FROM c22 left outer join dictionary d on c22.msg = d.word;

create table IF NOT EXISTS chsentiment

stored as orc as select

tweet\_id,

case

when sum( polarity ) > 0 then 'positive'

when sum( polarity ) < 0 then 'negative'

else 'neutral' end as sentiment

from c23 group by tweet\_id;

1. Count total positive and negative

INSERT OVERWRITE DIRECTORY '/user/hadoop/p3/zones/c/h/pr' ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' SELECT COUNT(sentiment) FROM chsentiment WHERE sentiment LIKE "%positive%";

INSERT OVERWRITE DIRECTORY '/user/hadoop/p3/zones/c/h/nr' ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' SELECT COUNT(sentiment) FROM chsentiment WHERE sentiment LIKE "%negative%";

1. Positive/Negative Sentiment Analysis for Eastern Trump

create view IF NOT EXISTS e11 as

select idone, message

from ezonet

LATERAL VIEW explode(sentences(lower(tweet))) dummy as message;

create view IF NOT EXISTS e12 as

SELECT idone, msg

FROM e11

lateral view explode(message) dummy as msg;

create view IF NOT EXISTS e13 as SELECT

idone tweet\_id, e12.msg,

case d.polarity

when 'negative' then -1

when 'positive' then 1

else 0 end as polarity

FROM e12 left outer join dictionary d on e12.msg = d.word;

create table IF NOT EXISTS etsentiment

stored as orc as select

tweet\_id,

case

when sum( polarity ) > 0 then 'positive'

when sum( polarity ) < 0 then 'negative'

else 'neutral' end as sentiment

from e13 group by tweet\_id;

1. Count total positive and negative

INSERT OVERWRITE DIRECTORY '/user/hadoop/p3/zones/e/t/pr' ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' SELECT COUNT(sentiment) FROM etsentiment WHERE sentiment LIKE "%positive%";

INSERT OVERWRITE DIRECTORY '/user/hadoop/p3/zones/e/t/nr' ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' SELECT COUNT(sentiment) FROM etsentiment WHERE sentiment LIKE "%negative%";

1. Positive/Negative Sentiment Analysis for Eastern Hillary

create view IF NOT EXISTS e21 as

select idone, message

from ezoneh

LATERAL VIEW explode(sentences(lower(tweet))) dummy as message;

create view IF NOT EXISTS e22 as

SELECT idone, msg

FROM e21

lateral view explode(message) dummy as msg;

create view IF NOT EXISTS e23 as SELECT

idone tweet\_id, e22.msg,

case d.polarity

when 'negative' then -1

when 'positive' then 1

else 0 end as polarity

FROM e22 left outer join dictionary d on e22.msg = d.word;

create table IF NOT EXISTS ehsentiment

stored as orc as select

tweet\_id,

case

when sum( polarity ) > 0 then 'positive'

when sum( polarity ) < 0 then 'negative'

else 'neutral' end as sentiment

from e23 group by tweet\_id;

1. Count total positive and negative

INSERT OVERWRITE DIRECTORY '/user/hadoop/p3/zones/e/h/pr' ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' SELECT COUNT(sentiment) FROM ehsentiment WHERE sentiment LIKE "%positive%";

INSERT OVERWRITE DIRECTORY '/user/hadoop/p3/zones/e/h/nr' ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' SELECT COUNT(sentiment) FROM ehsentiment WHERE sentiment LIKE "%negative%";

1. Exit Hive

quit;

Step 4: Get Results from Hadoop to Linux

This step is to get results from Hadoop to Linux

1. Create folders in main directory in Linux

cd ..

mkdir p3

1. Get files from Hadoop to Linux

hdfs dfs -get p3/zones/p/t/pr/000000\_0 p3/ptpr.txt

hdfs dfs -get p3/zones/p/t/nr/000000\_0 p3/ptnr.txt

hdfs dfs -get p3/zones/p/h/pr/000000\_0 p3/phpr.txt

hdfs dfs -get p3/zones/p/h/nr/000000\_0 p3/phnr.txt

hdfs dfs -get p3/zones/m/t/pr/000000\_0 p3/mtpr.txt

hdfs dfs -get p3/zones/m/t/nr/000000\_0 p3/mtnr.txt

hdfs dfs -get p3/zones/m/h/pr/000000\_0 p3/mhpr.txt

hdfs dfs -get p3/zones/m/h/nr/000000\_0 p3/mhnr.txt

hdfs dfs -get p3/zones/c/t/pr/000000\_0 p3/ctpr.txt

hdfs dfs -get p3/zones/c/t/nr/000000\_0 p3/ctnr.txt

hdfs dfs -get p3/zones/c/h/pr/000000\_0 p3/chpr.txt

hdfs dfs -get p3/zones/c/h/nr/000000\_0 p3/chnr.txt

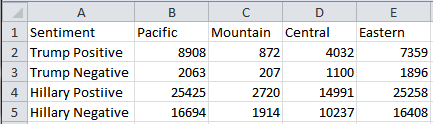
hdfs dfs -get p3/zones/e/t/pr/000000\_0 p3/etpr.txt

hdfs dfs -get p3/zones/e/t/nr/000000\_0 p3/etnr.txt

hdfs dfs -get p3/zones/e/h/pr/000000\_0 p3/ehpr.txt

hdfs dfs -get p3/zones/e/h/nr/000000\_0 p3/ehnr.txt

1. Read files and record results in Excel as shown



cd p3

1. Record Pacific Trump Positive Results

cat ptpr.txt

1. Record Pacific Trump Negative Results

cat ptnr.txt

1. Record Pacific Hillary Positive Results

cat phpr.txt

1. Record Pacific Hillary Negative Results

cat phnr.txt

1. Record Mountain Trump Positive Results

cat mtpr.txt

1. Record Mountain Trump Negative Results

cat mtnr.txt

1. Record Mountain Hillary Positive Results

cat mhpr.txt

1. Record Mountain Hillary Negative Results

cat mhnr.txt

1. Record Central Trump Positive Results

cat ctpr.txt

1. Record Central Trump Negative Results

cat ctnr.txt

1. Record Central Hillary Positive Results

cat chpr.txt

1. Record Central Hillary Negative Results

cat chnr.txt

1. Record Eastern Trump Positive Results

cat etpr.txt

1. Record Eastern Trump Negative Results

cat etnr.txt

1. Record Eastern Hillary Positive Results

cat ehpr.txt

1. Record Eastern Hillary Negative Results

cat ehnr.txt

Step 5: Visualization

This step is to do visualizations in Tableau

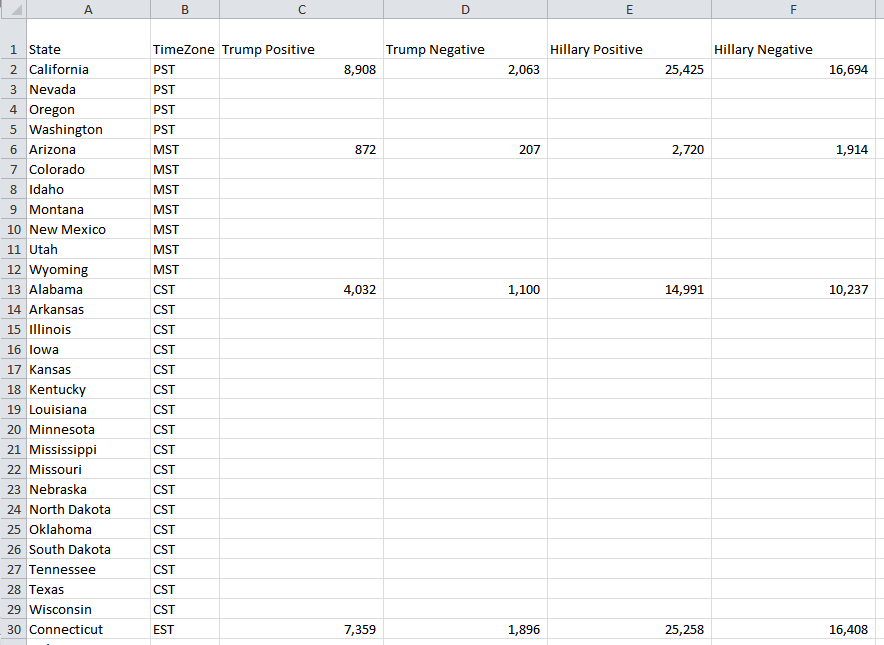
1. Download tableau.csv from github.com/tkim69/CIS5200project1
2. Enter Data from Previous Excel File to tableau.csv file

Enter Pacific data in California row

Enter Mountain data in Arizona row

Enter Central data in Alabama row

Enter Eastern data in Connecticut row



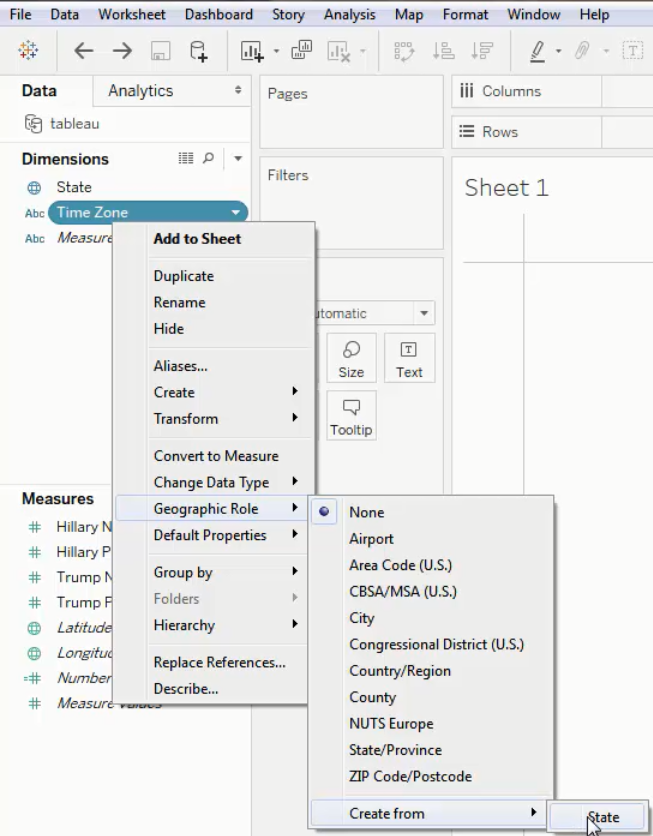
1. Save tableau.csv
2. Start Tableau application
3. Open tableau.csv in Tableau
4. Open Worksheet Sheet 1

Right click on Time Zone

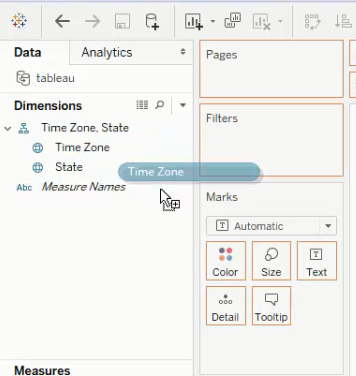
Select Geographic Role

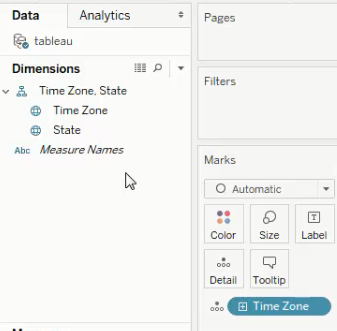
Select Create from

Click on State

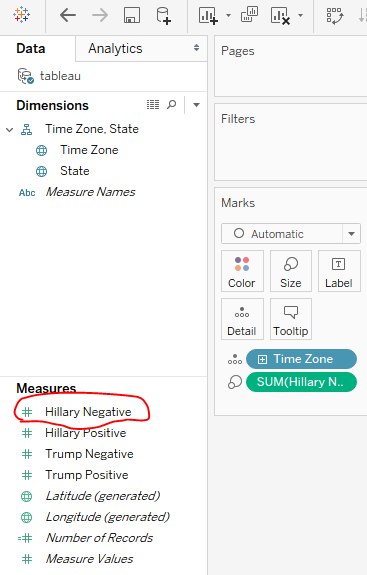


1. Drag and drop Time Zone from Dimension to Marks Box

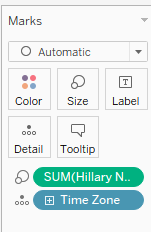




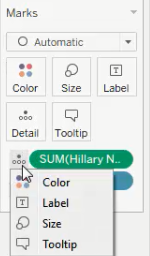
1. Double click on Hillary Negative in Measures box to add to Marks box



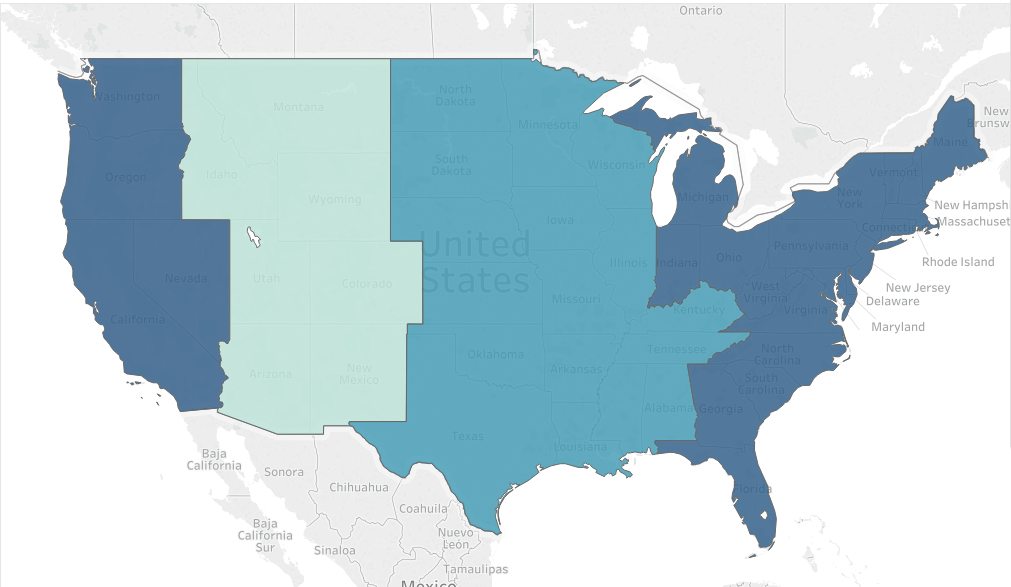
1. Drag and drop Timezone below SUM(Hillary Negative) in Marks box



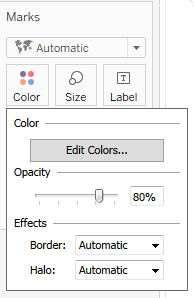
1. Click on icon next to the left of SUM(Hillary Negative) and change to Color



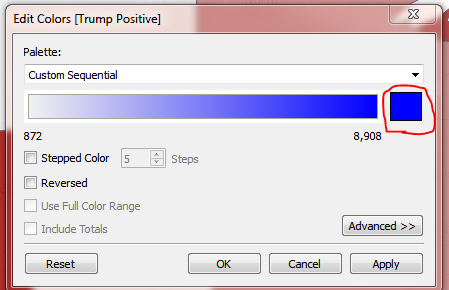
1. Verify Map has changed



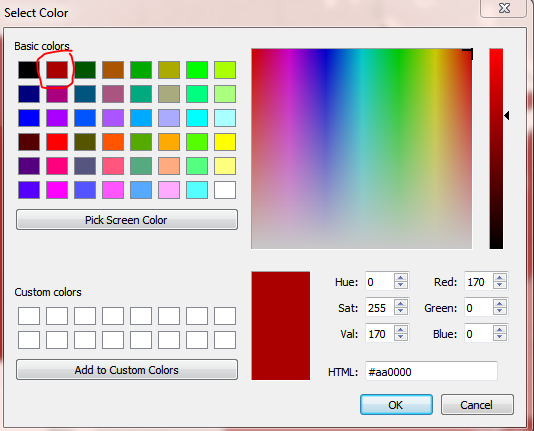
1. Remove SUM(Hillary Negative) from Marks box by dragging and dropping out of the Marks box
2. Double click on Trump Positive to add to Marks box
3. Change icon on left of SUM(Trump Positive) to color
4. Click on Color icon in Marks box and click on Edit Colors



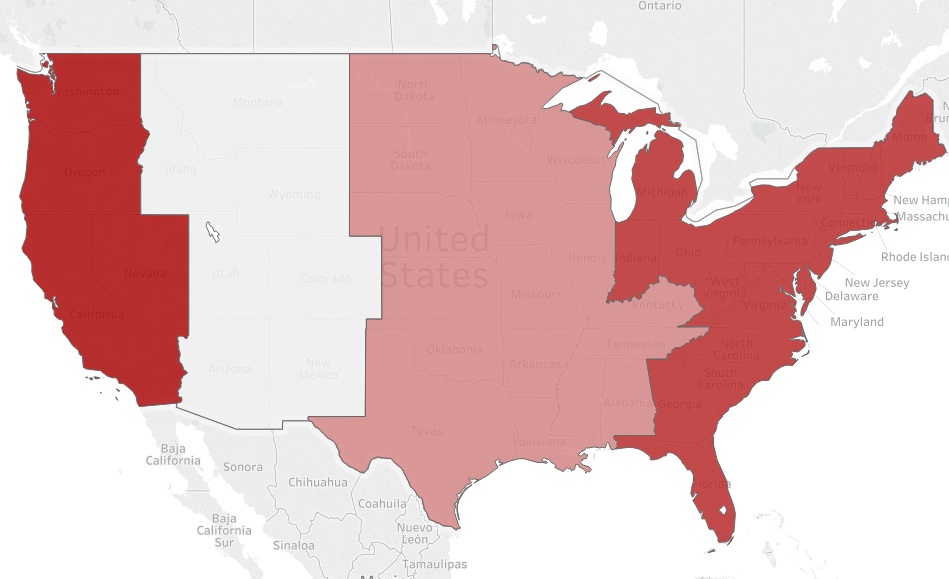
1. Click on blue box on right



1. Click on red color



1. Click on OK
2. Click on Apply
3. Click on OK
4. Verify that map color has changed



References

* 1. URL of Data Source, https://s3.amazonaws.com/vm-election10282016-2/tweets10302016.tgz
  2. URL of Dictionary Source, https://s3.amazonaws.com/hipicdatasets/dictionary.tsv
  3. URL of tableau.csv Source, https://github.com/tkim69/CIS5200project1
  4. URL of your Github, https://github.com/tkim69/CIS5200project1