**Analysis of L.A. Housing Market**

Yu-lin Shih [yulinshih@calstatela.edu](mailto:yulinshih@calstatela.edu)

Jiyang Liy jliu116@calstatela.edu

Tinayi Deng tdeng3@calstatela.edu

**Introduction**

This tutorial is help for explain how the table is created and method and function we used to query data for the project Analysis of L.A. Housing Market. We provide the code we used in hive and used in pig.

**Prerequisites**

* Set up Hadoop cluster.
* Download putty and pscp.
* Install Tableau and excel.

**Building Linux environment**

mkdir /home/username/data

mkdir /home/username/output

hdfs dfs –mkdir /dataset

**Using pscp to upload data**

pscp localfile/dataname username@ip:/home/username/data

There are five data which need to be upload.

* Business.csv
* Crime.csv
* Income.csv
* schoolDisc.csv
* price.csv

**Using Hive to create tables**

1. create database

create database if **not** **exists** project;

1. use project

use project;

1. Create crime table

create external table crime (

dateRptd Timestamp,

drNo String,

dateOCC Timestamp,

timeOCC String,

area String,

areaName String,

rD String,

crmCd String,

crmCdDesc String,

status String,

statusDesc String,

location String,

crossStreet STRING,

locationOne ARRAY<STRING>

) ROW FORMAT DELIMITED

FIELDS TERMINATED BY '#'

COLLECTION ITEMS TERMINATED BY "|"

STORED AS TEXTFILE

LOCATION '/dataset/crime';

1. create price table

create external table Price (

RegionName STRING,

state STRING,

RegionType STRING,

DataType STRING,

NOV2006 INT,

DEC2006 INT,

JAN2007 INT,

FEB2007 INT,

MAR2007 INT,

APR2007 INT,

MAY2007 INT,

JUN2007 INT,

JUL2007 INT,

AUG2007 INT,

SEP2007 INT,

OCT2007 INT,

NOV2007 INT,

DEC2007 INT,

JAN2008 INT,

FEB2008 INT,

MAR2008 INT,

APR2008 INT,

MAY2008 INT,

JUN2008 INT,

JUL2008 INT,

AUG2008 INT,

SEP2008 INT,

OCT2008 INT,

NOV2008 INT,

DEC2008 INT,

JAN2009 INT,

FEB2009 INT,

MAR2009 INT,

APR2009 INT,

MAY2009 INT,

JUN2009 INT,

JUL2009 INT,

AUG2009 INT,

SEP2009 INT,

OCT2009 INT,

NOV2009 INT,

DEC2009 INT,

JAN2010 INT,

FEB2010 INT,

MAR2010 INT,

APR2010 INT,

MAY2010 INT,

JUN2010 INT,

JUL2010 INT,

AUG2010 INT,

SEP2010 INT,

OCT2010 INT,

NOV2010 INT,

DEC2010 INT,

JAN2011 INT,

FEB2011 INT,

MAR2011 INT,

APR2011 INT,

MAY2011 INT,

JUN2011 INT,

JUL2011 INT,

AUG2011 INT,

SEP2011 INT,

OCT2011 INT,

NOV2011 INT,

DEC2011 INT,

JAN2012 INT,

FEB2012 INT,

MAR2012 INT,

APR2012 INT,

MAY2012 INT,

JUN2012 INT,

JUL2012 INT,

AUG2012 INT,

SEP2012 INT,

OCT2012 INT,

NOV2012 INT,

DEC2012 INT,

JAN2013 INT,

FEB2013 INT,

MAR2013 INT,

APR2013 INT,

MAY2013 INT,

JUN2013 INT,

JUL2013 INT,

AUG2013 INT,

SEP2013 INT,

OCT2013 INT,

NOV2013 INT,

DEC2013 INT,

JAN2014 INT,

FEB2014 INT,

MAR2014 INT,

APR2014 INT,

MAY2014 INT,

JUN2014 INT,

JUL2014 INT,

AUG2014 INT,

SEP2014 INT,

OCT2014 INT,

NOV2014 INT,

DEC2014 INT,

JAN2015 INT,

FEB2015 INT,

MAR2015 INT,

APR2015 INT,

MAY2015 INT,

JUN2015 INT,

JUL2015 INT,

AUG2015 INT,

SEP2015 INT,

OCT2015 INT,

NOV2015 INT,

DEC2015 INT,

JAN2016 INT,

FEB2016 INT,

MAR2016 INT,

APR2016 INT,

MAY2016 INT,

JUN2016 INT,

JUL2016 INT,

AUG2016 INT,

SEP2016 INT,

OCT2016 INT

) ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

COLLECTION ITEMS TERMINATED BY "|"

STORED AS TEXTFILE

LOCATION '/dataset/Price';

1. create schoolDisc table

create external table schoolDisc (

Ranking INT,

city STRING,

regionname STRING

) ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

COLLECTION ITEMS TERMINATED BY "|"

STORED AS TEXTFILE

LOCATION '/dataset/schoolDisc';

1. create income table

create external table income (

RegionName STRING,

state STRING,

income STRING,

level STRING

) ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

COLLECTION ITEMS TERMINATED BY "|"

STORED AS TEXTFILE

LOCATION '/dataset/income';

1. Create business table

create external table business (

LOCATION\_ACCOUNT STRING,

BUSINESS\_NAME STRING,

DBA\_NAME STRING,

STREET\_ADDRESS STRING,

CITY STRING,

ZIP\_CODE STRING,

LOCATION\_DESCRIPTION STRING,

MAILING\_ADDRESS STRING,

MAILING\_CITY STRING,

MAILING\_ZIP\_CODE STRING,

NAICS STRING,

PRIMARY\_NAICS\_DESCRIPTION STRING,

COUNCIL\_DISTRICT STRING,

LOCATION\_START\_DATE STRING,

LOCATION\_**END**\_DATE STRING,

LOCATION STRING

) ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

COLLECTION ITEMS TERMINATED BY "|"

STORED AS TEXTFILE

LOCATION '/dataset/business';

**Upload data to HDFS**

hdfs dfs –put /home/username/data/dataname /dataset/tablename

**Using Hive to query data**

1. Create 2016price table

create table if **not** **exists** 2016price

ROW FORMAT DELIMITED

FIELDS TERMINATED BY '#'

STORED AS TEXTFILE

LOCATION '/dataset/2016price'

as **select** RegionName, state, RegionType, JAN2016+FEB2016+MAR2016+APR2016+MAY2016+JUN2016+JUL2016+AUG2016+SEP2016+OCT2016 AS price2016

**from** price;

1. create 2016pricea table

create table if **not** **exists** 2016pricea

ROW FORMAT DELIMITED

FIELDS TERMINATED BY '#'

STORED AS TEXTFILE

LOCATION '/dataset/2016pricea'

as **select** RegionName, state, RegionType, price2016/10 as price2016a

**from** 2016price;

1. create 20123price table

create table if **not** **exists** 20123price

ROW FORMAT DELIMITED

FIELDS TERMINATED BY '#'

STORED AS TEXTFILE

LOCATION '/dataset/20123price'

as **select** RegionName, state, RegionType, JAN2012+FEB2012+MAR2012+APR2012+MAY2012+JUN2012+JUL2012+AUG2012+SEP2012+OCT2012+NOV2012+DEC2012+JAN2013+FEB2013+MAR2013+APR2013+MAY2013+JUN2013+JUL2013+AUG2013+SEP2013+OCT2013+NOV2013+DEC2013 AS price20123

**from** price;

1. create 20123pricea table

create table if **not** **exists** 20123pricea

ROW FORMAT DELIMITED

FIELDS TERMINATED BY '#'

STORED AS TEXTFILE

LOCATION '/dataset/20123pricea'

as **select** RegionName, state, RegionType, price20123/24 as price20123a

**from** 20123price;

1. create 20067price table

create table if **not** **exists** 20067price

ROW FORMAT DELIMITED

FIELDS TERMINATED BY '#'

STORED AS TEXTFILE

LOCATION '/dataset/2067price'

as **select** RegionName, state, RegionType, NOV2006+DEC2006+JAN2007+FEB2007+MAR2007+APR2007+MAY2007+JUN2007+JUL2007+AUG2007+SEP2007+OCT2007+NOV2007+DEC2007 AS price20067

**from** price;

1. create 20067pricea table

create table if **not** **exists** 20067pricea

ROW FORMAT DELIMITED

FIELDS TERMINATED BY '#'

STORED AS TEXTFILE

LOCATION '/dataset/20067pricea'

as **select** RegionName, state, RegionType, price20067/14 as price20123a

**from** 20067price;

1. create laavg table

create table if **not** **exists** laavg

ROW FORMAT DELIMITED

FIELDS TERMINATED BY '#'

STORED AS TEXTFILE

LOCATION '/dataset/laavg'

as **select** RegionName, state, RegionType,

AVG(NOV2006) AS 2006NOV, AVG(DEC2006) AS 2006DEC,

AVG(JAN2007) AS 2007JAN, AVG(FEB2007) AS 2007FEB, AVG(MAR2007) AS 2007MAR, AVG(APR2007) AS 2007APR, AVG(MAY2007) AS 2007MAY , AVG(JUN2007) AS 2007JUN, AVG(JUL2007) AS 2007JUL, AVG(AUG2007) AS 2007AUG, AVG(SEP2007) AS 2007SEP , AVG(OCT2007) AS 2007OCT, AVG(NOV2007) AS 2007NOV, AVG(DEC2007) AS 2007DEC,

AVG(JAN2008) AS 2008JAN, AVG(FEB2008) AS 2008FEB, AVG(MAR2008) AS 2008MAR, AVG(APR2008) AS 2008APR, AVG(MAY2008) AS 2008MAY , AVG(JUN2008) AS 2008JUN, AVG(JUL2008) AS 2008JUL, AVG(AUG2008) AS 2008AUG, AVG(SEP2008) AS 2008SEP , AVG(OCT2008) AS 2008OCT, AVG(NOV2008) AS 2008NOV, AVG(DEC2008) AS 2008DEC,

AVG(JAN2009) AS 2009JAN, AVG(FEB2009) AS 2009FEB, AVG(MAR2009) AS 2009MAR, AVG(APR2009) AS 2009APR, AVG(MAY2009) AS 2009MAY , AVG(JUN2009) AS 2009JUN, AVG(JUL2009) AS 2009JUL, AVG(AUG2009) AS 2009AUG, AVG(SEP2009) AS 2009SEP , AVG(OCT2009) AS 2009OCT, AVG(NOV2009) AS 2009NOV, AVG(DEC2009) AS 2009DEC,

AVG(JAN2010) AS 2010JAN, AVG(FEB2010) AS 2010FEB, AVG(MAR2010) AS 2010MAR, AVG(APR2010) AS 2010APR, AVG(MAY2010) AS 2010MAY , AVG(JUN2010) AS 2010JUN, AVG(JUL2010) AS 2010JUL, AVG(AUG2010) AS 2010AUG, AVG(SEP2010) AS 2010SEP , AVG(OCT2010) AS 2010OCT, AVG(NOV2010) AS 2010NOV, AVG(DEC2010) AS 2010DEC,

AVG(JAN2011) AS 2011JAN, AVG(FEB2011) AS 2011FEB, AVG(MAR2011) AS 2011MAR, AVG(APR2011) AS 2011APR, AVG(MAY2011) AS 2011MAY , AVG(JUN2011) AS 2011JUN, AVG(JUL2011) AS 2011JUL, AVG(AUG2011) AS 2011AUG, AVG(SEP2011) AS 2011SEP , AVG(OCT2011) AS 2011OCT, AVG(NOV2011) AS 2011NOV, AVG(DEC2011) AS 2011DEC,

AVG(JAN2012) AS 2012JAN, AVG(FEB2012) AS 2012FEB, AVG(MAR2012) AS 2012MAR, AVG(APR2012) AS 2012APR, AVG(MAY2012) AS 2012MAY , AVG(JUN2012) AS 2012JUN, AVG(JUL2012) AS 2012JUL, AVG(AUG2012) AS 2012AUG, AVG(SEP2012) AS 2012SEP , AVG(OCT2012) AS 2012OCT, AVG(NOV2012) AS 2012NOV, AVG(DEC2012) AS 2012DEC,

AVG(JAN2013) AS 2013JAN, AVG(FEB2013) AS 2013FEB, AVG(MAR2013) AS 2013MAR, AVG(APR2013) AS 2013APR, AVG(MAY2013) AS 2013MAY , AVG(JUN2013) AS 2013JUN, AVG(JUL2013) AS 2013JUL, AVG(AUG2013) AS 2013AUG, AVG(SEP2013) AS 2013SEP , AVG(OCT2013) AS 2013OCT, AVG(NOV2013) AS 2013NOV, AVG(DEC2013) AS 2013DEC,

AVG(JAN2014) AS 2014JAN, AVG(FEB2014) AS 2014FEB, AVG(MAR2014) AS 2014MAR, AVG(APR2014) AS 2014APR, AVG(MAY2014) AS 2014MAY , AVG(JUN2014) AS 2014JUN, AVG(JUL2014) AS 2014JUL, AVG(AUG2014) AS 2014AUG, AVG(SEP2014) AS 2014SEP , AVG(OCT2014) AS 2014OCT, AVG(NOV2014) AS 2014NOV, AVG(DEC2014) AS 2014DEC,

AVG(JAN2015) AS 2015JAN, AVG(FEB2015) AS 2015FEB, AVG(MAR2015) AS 2015MAR, AVG(APR2015) AS 2015APR, AVG(MAY2015) AS 2015MAY , AVG(JUN2015) AS 2015JUN, AVG(JUL2015) AS 2015JUL, AVG(AUG2015) AS 2015AUG, AVG(SEP2015) AS 2015SEP , AVG(OCT2015) AS 2015OCT, AVG(NOV2015) AS 2015NOV, AVG(DEC2015) AS 2015DEC,

AVG(JAN2016) AS 2016JAN, AVG(FEB2016) AS 2016FEB, AVG(MAR2016) AS 2016MAR, AVG(APR2016) AS 2016APR, AVG(MAY2016) AS 2016MAY , AVG(JUN2016) AS 2016JUN, AVG(JUL2016) AS 2016JUL, AVG(AUG2016) AS 2016AUG, AVG(SEP2016) AS 2016SEP , AVG(OCT2016) AS 2016OCT

**from** price **GROUP** BY RegionName, state, RegionType;

1. create latureavg table

create table if **not** **exists** latrueavg

ROW FORMAT DELIMITED

FIELDS TERMINATED BY '#'

STORED AS TEXTFILE

LOCATION '/dataset/latrueavg'

as **select**

AVG(NOV2006) AS 2006NOV, AVG(DEC2006) AS 2006DEC,

AVG(JAN2007) AS 2007JAN, AVG(FEB2007) AS 2007FEB, AVG(MAR2007) AS 2007MAR, AVG(APR2007) AS 2007APR, AVG(MAY2007) AS 2007MAY , AVG(JUN2007) AS 2007JUN, AVG(JUL2007) AS 2007JUL, AVG(AUG2007) AS 2007AUG, AVG(SEP2007) AS 2007SEP , AVG(OCT2007) AS 2007OCT, AVG(NOV2007) AS 2007NOV, AVG(DEC2007) AS 2007DEC,

AVG(JAN2008) AS 2008JAN, AVG(FEB2008) AS 2008FEB, AVG(MAR2008) AS 2008MAR, AVG(APR2008) AS 2008APR, AVG(MAY2008) AS 2008MAY , AVG(JUN2008) AS 2008JUN, AVG(JUL2008) AS 2008JUL, AVG(AUG2008) AS 2008AUG, AVG(SEP2008) AS 2008SEP , AVG(OCT2008) AS 2008OCT, AVG(NOV2008) AS 2008NOV, AVG(DEC2008) AS 2008DEC,

AVG(JAN2009) AS 2009JAN, AVG(FEB2009) AS 2009FEB, AVG(MAR2009) AS 2009MAR, AVG(APR2009) AS 2009APR, AVG(MAY2009) AS 2009MAY , AVG(JUN2009) AS 2009JUN, AVG(JUL2009) AS 2009JUL, AVG(AUG2009) AS 2009AUG, AVG(SEP2009) AS 2009SEP , AVG(OCT2009) AS 2009OCT, AVG(NOV2009) AS 2009NOV, AVG(DEC2009) AS 2009DEC,

AVG(JAN2010) AS 2010JAN, AVG(FEB2010) AS 2010FEB, AVG(MAR2010) AS 2010MAR, AVG(APR2010) AS 2010APR, AVG(MAY2010) AS 2010MAY , AVG(JUN2010) AS 2010JUN, AVG(JUL2010) AS 2010JUL, AVG(AUG2010) AS 2010AUG, AVG(SEP2010) AS 2010SEP , AVG(OCT2010) AS 2010OCT, AVG(NOV2010) AS 2010NOV, AVG(DEC2010) AS 2010DEC,

AVG(JAN2011) AS 2011JAN, AVG(FEB2011) AS 2011FEB, AVG(MAR2011) AS 2011MAR, AVG(APR2011) AS 2011APR, AVG(MAY2011) AS 2011MAY , AVG(JUN2011) AS 2011JUN, AVG(JUL2011) AS 2011JUL, AVG(AUG2011) AS 2011AUG, AVG(SEP2011) AS 2011SEP , AVG(OCT2011) AS 2011OCT, AVG(NOV2011) AS 2011NOV, AVG(DEC2011) AS 2011DEC,

AVG(JAN2012) AS 2012JAN, AVG(FEB2012) AS 2012FEB, AVG(MAR2012) AS 2012MAR, AVG(APR2012) AS 2012APR, AVG(MAY2012) AS 2012MAY , AVG(JUN2012) AS 2012JUN, AVG(JUL2012) AS 2012JUL, AVG(AUG2012) AS 2012AUG, AVG(SEP2012) AS 2012SEP , AVG(OCT2012) AS 2012OCT, AVG(NOV2012) AS 2012NOV, AVG(DEC2012) AS 2012DEC,

AVG(JAN2013) AS 2013JAN, AVG(FEB2013) AS 2013FEB, AVG(MAR2013) AS 2013MAR, AVG(APR2013) AS 2013APR, AVG(MAY2013) AS 2013MAY , AVG(JUN2013) AS 2013JUN, AVG(JUL2013) AS 2013JUL, AVG(AUG2013) AS 2013AUG, AVG(SEP2013) AS 2013SEP , AVG(OCT2013) AS 2013OCT, AVG(NOV2013) AS 2013NOV, AVG(DEC2013) AS 2013DEC,

AVG(JAN2014) AS 2014JAN, AVG(FEB2014) AS 2014FEB, AVG(MAR2014) AS 2014MAR, AVG(APR2014) AS 2014APR, AVG(MAY2014) AS 2014MAY , AVG(JUN2014) AS 2014JUN, AVG(JUL2014) AS 2014JUL, AVG(AUG2014) AS 2014AUG, AVG(SEP2014) AS 2014SEP , AVG(OCT2014) AS 2014OCT, AVG(NOV2014) AS 2014NOV, AVG(DEC2014) AS 2014DEC,

AVG(JAN2015) AS 2015JAN, AVG(FEB2015) AS 2015FEB, AVG(MAR2015) AS 2015MAR, AVG(APR2015) AS 2015APR, AVG(MAY2015) AS 2015MAY , AVG(JUN2015) AS 2015JUN, AVG(JUL2015) AS 2015JUL, AVG(AUG2015) AS 2015AUG, AVG(SEP2015) AS 2015SEP , AVG(OCT2015) AS 2015OCT, AVG(NOV2015) AS 2015NOV, AVG(DEC2015) AS 2015DEC,

AVG(JAN2016) AS 2016JAN, AVG(FEB2016) AS 2016FEB, AVG(MAR2016) AS 2016MAR, AVG(APR2016) AS 2016APR, AVG(MAY2016) AS 2016MAY , AVG(JUN2016) AS 2016JUN, AVG(JUL2016) AS 2016JUL, AVG(AUG2016) AS 2016AUG, AVG(SEP2016) AS 2016SEP , AVG(OCT2016) AS 2016OCT

**from** price;

1. create incomenprice table

CREATE TABLE IF **NOT** **EXISTS** incomenprice

ROW FORMAT DELIMITED

FIELDS TERMINATED BY '#'

STORED AS TEXTFILE

LOCATION '/dataset/incomenprice'

as **select** p.regionname, p.state, p.price2016a, i.income, i.level

**FROM** 2016pricea p

CROSS **JOIN** income i

**ON** (p.regionname = i.regionname);

1. create schoolnprice table

CREATE TABLE IF **NOT** **EXISTS** schoolnprice

ROW FORMAT DELIMITED

FIELDS TERMINATED BY '#'

STORED AS TEXTFILE

LOCATION '/dataset/schoolnprice'

as **select** p.regionname, p.state, p.price2016a, s.ranking

**FROM** 2016pricea p

CROSS **JOIN** schoolDisc s

**ON** (p.regionname = s.regionname);

1. Create table businesspercity table

CREATE TABLE IF **NOT** **EXISTS** businesspercity

ROW FORMAT DELIMITED

FIELDS TERMINATED BY '#'

STORED AS TEXTFILE

LOCATION '/dataset/businesspercity'

AS **SELECT** city, count(city) AS number

**FROM** business

**GROUP** BY city;

**Using Pig to create tables**

1. Create crime table

crime = LOAD '/dataset/crime.csv' AS (

dateRptd:Timestamp,

drNo:String,

dateOCC:Timestamp,

timeOCC:String,

area:String,

areaName:String,

rD:String,

crmCd:String,

crmCdDesc:String,

status:String,

statusDesc:String,

location:String,

crossStreet:STRING,

locationOne:STRING

);

1. create price table

price = LOAD '/dataset/price.csv' AS (

RegionName STRING,

state STRING,

RegionType:STRING,

DataType:STRING,

NOV2006:INT,

DEC2006:INT,

JAN2007:INT,

FEB2007:INT,

MAR2007:INT,

APR2007:INT,

MAY2007:INT,

JUN2007:INT,

JUL2007:INT,

AUG2007:INT,

SEP2007:INT,

OCT2007:INT,

NOV2007:INT,

DEC2007:INT,

JAN2008:INT,

FEB2008:INT,

MAR2008:INT,

APR2008:INT,

MAY2008:INT,

JUN2008:INT,

JUL2008:INT,

AUG2008:INT,

SEP2008:INT,

OCT2008:INT,

NOV2008:INT,

DEC2008:INT,

JAN2009:INT,

FEB2009:INT,

MAR2009:INT,

APR2009:INT,

MAY2009:INT,

JUN2009:INT,

JUL2009:INT,

AUG2009:INT,

SEP2009:INT,

OCT2009:INT,

NOV2009:INT,

DEC2009:INT,

JAN2010:INT,

FEB2010:INT,

MAR2010:INT,

APR2010:INT,

MAY2010:INT,

JUN2010:INT,

JUL2010:INT,

AUG2010:INT,

SEP2010:INT,

OCT2010:INT,

NOV2010:INT,

DEC2010:INT,

JAN2011:INT,

FEB2011:INT,

MAR2011:INT,

APR2011:INT,

MAY2011:INT,

JUN2011:INT,

JUL2011:INT,

AUG2011:INT,

SEP2011:INT,

OCT2011:INT,

NOV2011:INT,

DEC2011:INT,

JAN2012:INT,

FEB2012:INT,

MAR2012:INT,

APR2012:INT,

MAY2012:INT,

JUN2012:INT,

JUL2012:INT,

AUG2012:INT,

SEP2012:INT,

OCT2012:INT,

NOV2012:INT,

DEC2012:INT,

JAN2013:INT,

FEB2013:INT,

MAR2013:INT,

APR2013:INT,

MAY2013:INT,

JUN2013:INT,

JUL2013:INT,

AUG2013:INT,

SEP2013:INT,

OCT2013:INT,

NOV2013:INT,

DEC2013:INT,

JAN2014:INT,

FEB2014:INT,

MAR2014:INT,

APR2014:INT,

MAY2014:INT,

JUN2014:INT,

JUL2014:INT,

AUG2014:INT,

SEP2014:INT,

OCT2014:INT,

NOV2014:INT,

DEC2014:INT,

JAN2015:INT,

FEB2015:INT,

MAR2015:INT,

APR2015:INT,

MAY2015:INT,

JUN2015:INT,

JUL2015:INT,

AUG2015:INT,

SEP2015:INT,

OCT2015:INT,

NOV2015:INT,

DEC2015:INT,

JAN2016:INT,

FEB2016:INT,

MAR2016:INT,

APR2016:INT,

MAY2016:INT,

JUN2016:INT,

JUL2016:INT,

AUG2016:INT,

SEP2016:INT,

OCT2016:INT

);

1. create schoolDisc table

schoolDisc = LOAD '/dataset/schoolDisc.csv' AS (

Ranking:INT,

city:STRING,

regionname STRING

);

1. create income table

income = LOAD '/dataset/income.csv' AS (

RegionName:STRING,

state:STRING,

income:STRING,

level:STRING

);

1. Create business table

business = LOAD '/dataset/business.csv' AS (

LOCATION\_ACCOUNT:STRING,

BUSINESS\_NAME:STRING,

DBA\_NAME:STRING,

STREET\_ADDRESS:STRING,

CITY:STRING,

ZIP\_CODE:STRING,

LOCATION\_DESCRIPTION:STRING,

MAILING\_ADDRESS:STRING,

MAILING\_CITY:STRING,

MAILING\_ZIP\_CODE:STRING,

NAICS:STRING,

PRIMARY\_NAICS\_DESCRIPTION:STRING,

COUNCIL\_DISTRICT:STRING,

LOCATION\_START\_DATE:STRING,

LOCATION\_END\_DATE:STRING,

LOCATION:STRING

);

**Using Pig to query data**

1. Create 2016price table

2016price = FOREACH price GENERATE RegionName, state, RegionType, JAN2016+FEB2016+MAR2016+APR2016+MAY2016+JUN2016+JUL2016+AUG2016+SEP2016+OCT2016 AS price2016;

1. create 2016pricea table

2016pricea = FOREACH 2016price GENERATE RegionName, state, RegionType, price2016/10 AS price2016a;

1. create 20123price table

20123price = FOREACH price GENERATE RegionName, state, RegionType, JAN2012+FEB2012+MAR2012+APR2012+MAY2012+JUN2012+JUL2012+AUG2012+SEP2012+OCT2012+NOV2012+DEC2012+JAN2013+FEB2013+MAR2013+APR2013+MAY2013+JUN2013+JUL2013+AUG2013+SEP2013+OCT2013+NOV2013+DEC2013 AS price20123;

1. create 20123pricea table

20123pricea = FOREACH 20123price GENERATE RegionName, state, RegionType, price20123/24 AS price20123a;

1. create 20067price table

20067price = FOREACH price GENERATE RegionName, state, RegionType, NOV2006+DEC2006+JAN2007+FEB2007+MAR2007+APR2007+MAY2007+JUN2007+JUL2007+AUG2007+SEP2007+OCT2007+NOV2007+DEC2007 AS price20067;

1. create 20067pricea table

20067pricea = FOREACH 20067price RegionName, state, RegionType, price20067/14 AS price20123a;

1. create laavg table

laavg = FOREACH price GENERATE RegionName, state, RegionType,

AVG(NOV2006) AS 2006NOV, AVG(DEC2006) AS 2006DEC,

AVG(JAN2007) AS 2007JAN, AVG(FEB2007) AS 2007FEB, AVG(MAR2007) AS 2007MAR, AVG(APR2007) AS 2007APR, AVG(MAY2007) AS 2007MAY , AVG(JUN2007) AS 2007JUN, AVG(JUL2007) AS 2007JUL, AVG(AUG2007) AS 2007AUG, AVG(SEP2007) AS 2007SEP , AVG(OCT2007) AS 2007OCT, AVG(NOV2007) AS 2007NOV, AVG(DEC2007) AS 2007DEC,

AVG(JAN2008) AS 2008JAN, AVG(FEB2008) AS 2008FEB, AVG(MAR2008) AS 2008MAR, AVG(APR2008) AS 2008APR, AVG(MAY2008) AS 2008MAY , AVG(JUN2008) AS 2008JUN, AVG(JUL2008) AS 2008JUL, AVG(AUG2008) AS 2008AUG, AVG(SEP2008) AS 2008SEP , AVG(OCT2008) AS 2008OCT, AVG(NOV2008) AS 2008NOV, AVG(DEC2008) AS 2008DEC,

AVG(JAN2009) AS 2009JAN, AVG(FEB2009) AS 2009FEB, AVG(MAR2009) AS 2009MAR, AVG(APR2009) AS 2009APR, AVG(MAY2009) AS 2009MAY , AVG(JUN2009) AS 2009JUN, AVG(JUL2009) AS 2009JUL, AVG(AUG2009) AS 2009AUG, AVG(SEP2009) AS 2009SEP , AVG(OCT2009) AS 2009OCT, AVG(NOV2009) AS 2009NOV, AVG(DEC2009) AS 2009DEC,

AVG(JAN2010) AS 2010JAN, AVG(FEB2010) AS 2010FEB, AVG(MAR2010) AS 2010MAR, AVG(APR2010) AS 2010APR, AVG(MAY2010) AS 2010MAY , AVG(JUN2010) AS 2010JUN, AVG(JUL2010) AS 2010JUL, AVG(AUG2010) AS 2010AUG, AVG(SEP2010) AS 2010SEP , AVG(OCT2010) AS 2010OCT, AVG(NOV2010) AS 2010NOV, AVG(DEC2010) AS 2010DEC,

AVG(JAN2011) AS 2011JAN, AVG(FEB2011) AS 2011FEB, AVG(MAR2011) AS 2011MAR, AVG(APR2011) AS 2011APR, AVG(MAY2011) AS 2011MAY , AVG(JUN2011) AS 2011JUN, AVG(JUL2011) AS 2011JUL, AVG(AUG2011) AS 2011AUG, AVG(SEP2011) AS 2011SEP , AVG(OCT2011) AS 2011OCT, AVG(NOV2011) AS 2011NOV, AVG(DEC2011) AS 2011DEC,

AVG(JAN2012) AS 2012JAN, AVG(FEB2012) AS 2012FEB, AVG(MAR2012) AS 2012MAR, AVG(APR2012) AS 2012APR, AVG(MAY2012) AS 2012MAY , AVG(JUN2012) AS 2012JUN, AVG(JUL2012) AS 2012JUL, AVG(AUG2012) AS 2012AUG, AVG(SEP2012) AS 2012SEP , AVG(OCT2012) AS 2012OCT, AVG(NOV2012) AS 2012NOV, AVG(DEC2012) AS 2012DEC,

AVG(JAN2013) AS 2013JAN, AVG(FEB2013) AS 2013FEB, AVG(MAR2013) AS 2013MAR, AVG(APR2013) AS 2013APR, AVG(MAY2013) AS 2013MAY , AVG(JUN2013) AS 2013JUN, AVG(JUL2013) AS 2013JUL, AVG(AUG2013) AS 2013AUG, AVG(SEP2013) AS 2013SEP , AVG(OCT2013) AS 2013OCT, AVG(NOV2013) AS 2013NOV, AVG(DEC2013) AS 2013DEC,

AVG(JAN2014) AS 2014JAN, AVG(FEB2014) AS 2014FEB, AVG(MAR2014) AS 2014MAR, AVG(APR2014) AS 2014APR, AVG(MAY2014) AS 2014MAY , AVG(JUN2014) AS 2014JUN, AVG(JUL2014) AS 2014JUL, AVG(AUG2014) AS 2014AUG, AVG(SEP2014) AS 2014SEP , AVG(OCT2014) AS 2014OCT, AVG(NOV2014) AS 2014NOV, AVG(DEC2014) AS 2014DEC,

AVG(JAN2015) AS 2015JAN, AVG(FEB2015) AS 2015FEB, AVG(MAR2015) AS 2015MAR, AVG(APR2015) AS 2015APR, AVG(MAY2015) AS 2015MAY , AVG(JUN2015) AS 2015JUN, AVG(JUL2015) AS 2015JUL, AVG(AUG2015) AS 2015AUG, AVG(SEP2015) AS 2015SEP , AVG(OCT2015) AS 2015OCT, AVG(NOV2015) AS 2015NOV, AVG(DEC2015) AS 2015DEC,

AVG(JAN2016) AS 2016JAN, AVG(FEB2016) AS 2016FEB, AVG(MAR2016) AS 2016MAR, AVG(APR2016) AS 2016APR, AVG(MAY2016) AS 2016MAY , AVG(JUN2016) AS 2016JUN, AVG(JUL2016) AS 2016JUL, AVG(AUG2016) AS 2016AUG, AVG(SEP2016) AS 2016SEP , AVG(OCT2016) AS 2016OCT;

1. create latureavg table

latrueavg = FOREACH price GENERATE AVG(NOV2006) AS 2006NOV, AVG(DEC2006) AS 2006DEC,

AVG(JAN2007) AS 2007JAN, AVG(FEB2007) AS 2007FEB, AVG(MAR2007) AS 2007MAR, AVG(APR2007) AS 2007APR, AVG(MAY2007) AS 2007MAY , AVG(JUN2007) AS 2007JUN, AVG(JUL2007) AS 2007JUL, AVG(AUG2007) AS 2007AUG, AVG(SEP2007) AS 2007SEP , AVG(OCT2007) AS 2007OCT, AVG(NOV2007) AS 2007NOV, AVG(DEC2007) AS 2007DEC,

AVG(JAN2008) AS 2008JAN, AVG(FEB2008) AS 2008FEB, AVG(MAR2008) AS 2008MAR, AVG(APR2008) AS 2008APR, AVG(MAY2008) AS 2008MAY , AVG(JUN2008) AS 2008JUN, AVG(JUL2008) AS 2008JUL, AVG(AUG2008) AS 2008AUG, AVG(SEP2008) AS 2008SEP , AVG(OCT2008) AS 2008OCT, AVG(NOV2008) AS 2008NOV, AVG(DEC2008) AS 2008DEC,

AVG(JAN2009) AS 2009JAN, AVG(FEB2009) AS 2009FEB, AVG(MAR2009) AS 2009MAR, AVG(APR2009) AS 2009APR, AVG(MAY2009) AS 2009MAY , AVG(JUN2009) AS 2009JUN, AVG(JUL2009) AS 2009JUL, AVG(AUG2009) AS 2009AUG, AVG(SEP2009) AS 2009SEP , AVG(OCT2009) AS 2009OCT, AVG(NOV2009) AS 2009NOV, AVG(DEC2009) AS 2009DEC,

AVG(JAN2010) AS 2010JAN, AVG(FEB2010) AS 2010FEB, AVG(MAR2010) AS 2010MAR, AVG(APR2010) AS 2010APR, AVG(MAY2010) AS 2010MAY , AVG(JUN2010) AS 2010JUN, AVG(JUL2010) AS 2010JUL, AVG(AUG2010) AS 2010AUG, AVG(SEP2010) AS 2010SEP , AVG(OCT2010) AS 2010OCT, AVG(NOV2010) AS 2010NOV, AVG(DEC2010) AS 2010DEC,

AVG(JAN2011) AS 2011JAN, AVG(FEB2011) AS 2011FEB, AVG(MAR2011) AS 2011MAR, AVG(APR2011) AS 2011APR, AVG(MAY2011) AS 2011MAY , AVG(JUN2011) AS 2011JUN, AVG(JUL2011) AS 2011JUL, AVG(AUG2011) AS 2011AUG, AVG(SEP2011) AS 2011SEP , AVG(OCT2011) AS 2011OCT, AVG(NOV2011) AS 2011NOV, AVG(DEC2011) AS 2011DEC,

AVG(JAN2012) AS 2012JAN, AVG(FEB2012) AS 2012FEB, AVG(MAR2012) AS 2012MAR, AVG(APR2012) AS 2012APR, AVG(MAY2012) AS 2012MAY , AVG(JUN2012) AS 2012JUN, AVG(JUL2012) AS 2012JUL, AVG(AUG2012) AS 2012AUG, AVG(SEP2012) AS 2012SEP , AVG(OCT2012) AS 2012OCT, AVG(NOV2012) AS 2012NOV, AVG(DEC2012) AS 2012DEC,

AVG(JAN2013) AS 2013JAN, AVG(FEB2013) AS 2013FEB, AVG(MAR2013) AS 2013MAR, AVG(APR2013) AS 2013APR, AVG(MAY2013) AS 2013MAY , AVG(JUN2013) AS 2013JUN, AVG(JUL2013) AS 2013JUL, AVG(AUG2013) AS 2013AUG, AVG(SEP2013) AS 2013SEP , AVG(OCT2013) AS 2013OCT, AVG(NOV2013) AS 2013NOV, AVG(DEC2013) AS 2013DEC,

AVG(JAN2014) AS 2014JAN, AVG(FEB2014) AS 2014FEB, AVG(MAR2014) AS 2014MAR, AVG(APR2014) AS 2014APR, AVG(MAY2014) AS 2014MAY , AVG(JUN2014) AS 2014JUN, AVG(JUL2014) AS 2014JUL, AVG(AUG2014) AS 2014AUG, AVG(SEP2014) AS 2014SEP , AVG(OCT2014) AS 2014OCT, AVG(NOV2014) AS 2014NOV, AVG(DEC2014) AS 2014DEC,

AVG(JAN2015) AS 2015JAN, AVG(FEB2015) AS 2015FEB, AVG(MAR2015) AS 2015MAR, AVG(APR2015) AS 2015APR, AVG(MAY2015) AS 2015MAY , AVG(JUN2015) AS 2015JUN, AVG(JUL2015) AS 2015JUL, AVG(AUG2015) AS 2015AUG, AVG(SEP2015) AS 2015SEP , AVG(OCT2015) AS 2015OCT, AVG(NOV2015) AS 2015NOV, AVG(DEC2015) AS 2015DEC,

AVG(JAN2016) AS 2016JAN, AVG(FEB2016) AS 2016FEB, AVG(MAR2016) AS 2016MAR, AVG(APR2016) AS 2016APR, AVG(MAY2016) AS 2016MAY , AVG(JUN2016) AS 2016JUN, AVG(JUL2016) AS 2016JUL, AVG(AUG2016) AS 2016AUG, AVG(SEP2016) AS 2016SEP , AVG(OCT2016) AS 2016OCT;

1. create incomenprice table

incomenprice\_tmp = JOIN 2016pricea BY regionname FULL OUTER, income BY regionname;

incomenprice = FOREACH incomenprice\_tmp GENERATE regionname, state, price2016a, income, level;

1. create schoolnprice table

schoolnprice\_tmp = JOIN 2016pricea BY regionname FULL OUTER, schoolDisc BY regionname;

schoolnprice = FOREACH schoolnprice\_tmp GENERATE regionname, state, price2016a, ranking;

1. Create table businesspercity table

businesspercity = FOREACH business GENERATE city, count(city) AS nubmer;

**Download data from HDFS**

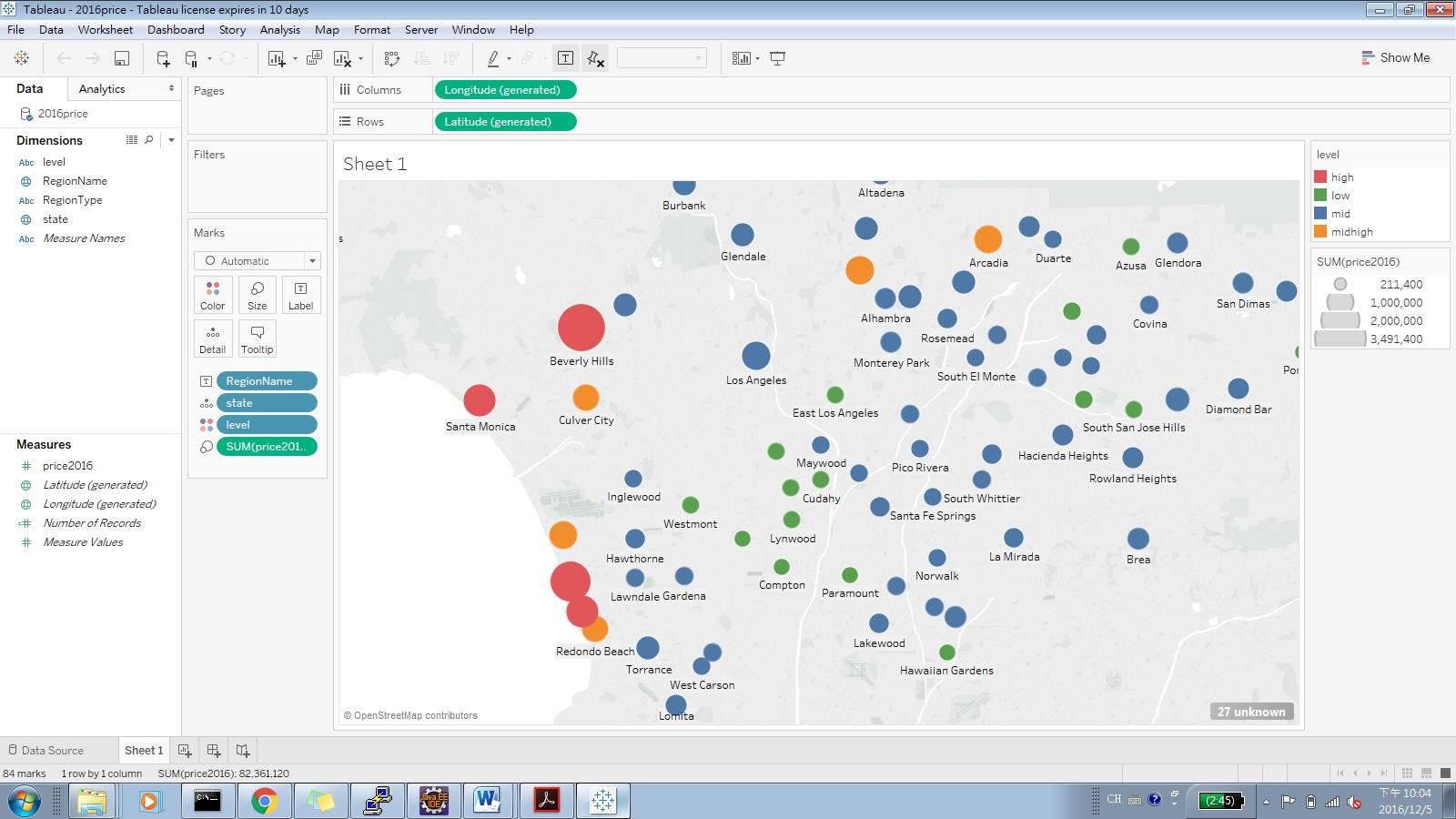
Hdfs dfs –get /dataset/tablename /home/username/output

**Using pscp to download data**

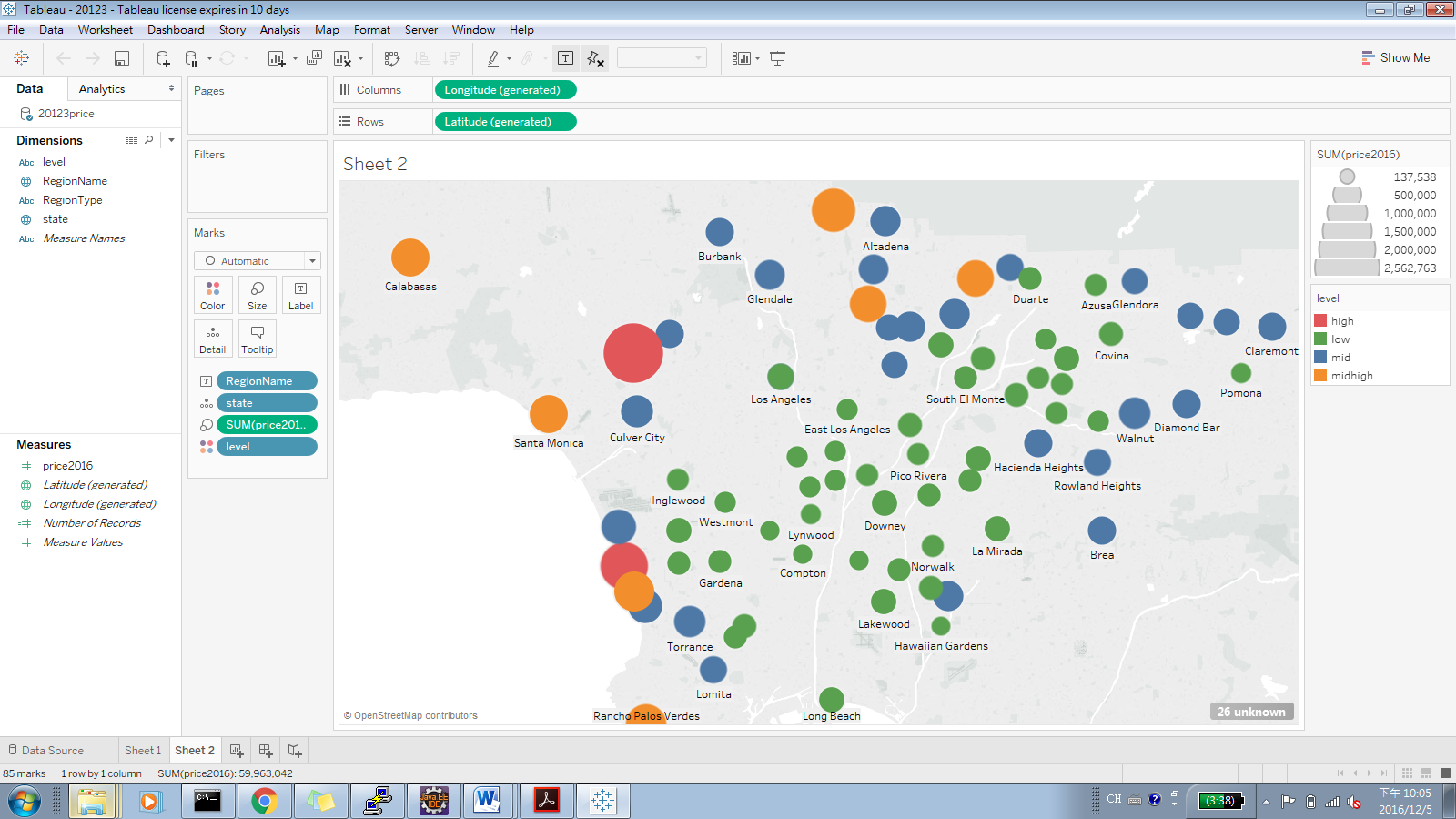
Pscp username@ip:/home/username/data/tablename/000000\_0 localfile

**Data Use Tableau or Excel to Manufacture the graph**

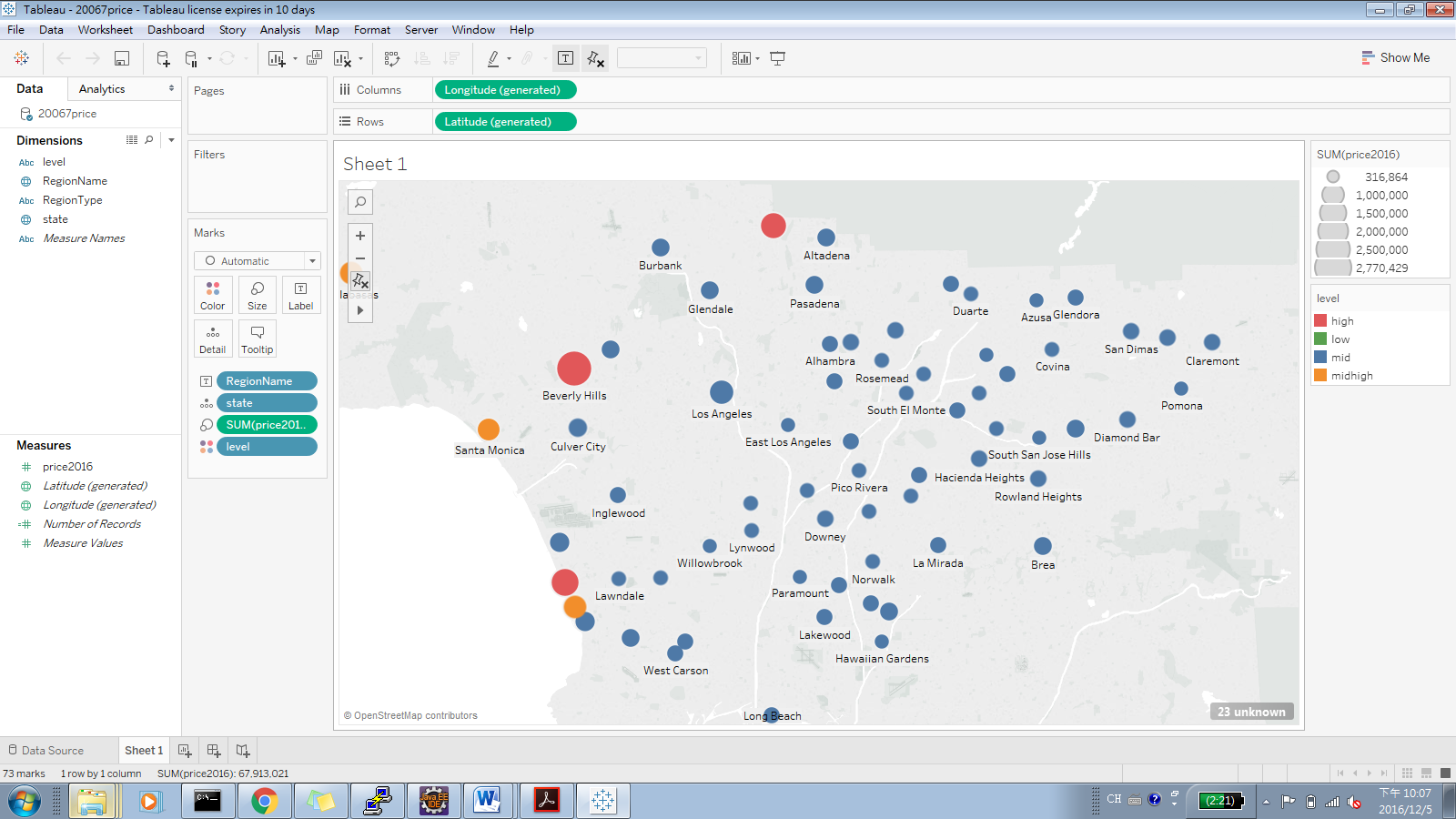
* 2016Price



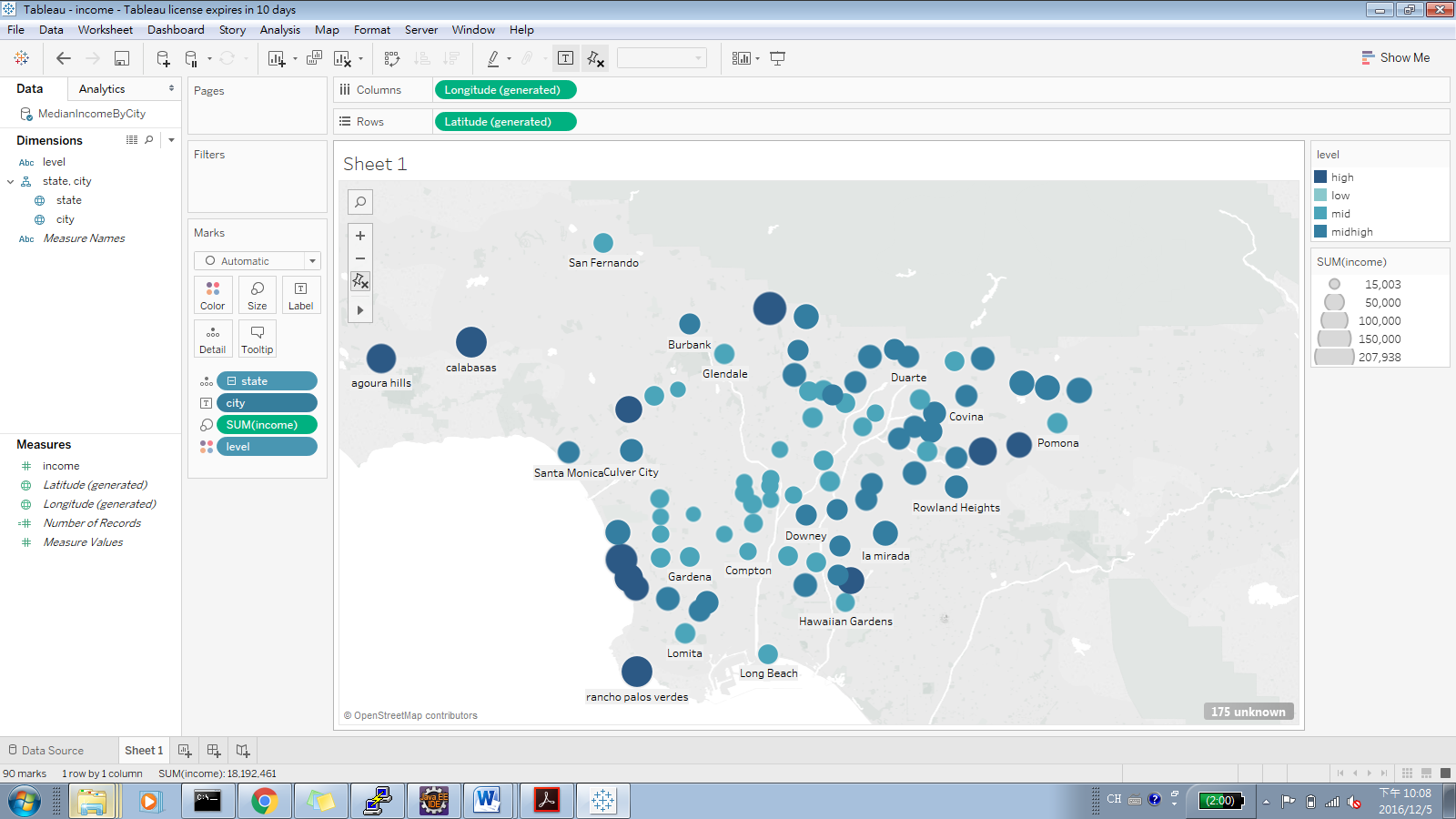
* 20023Price



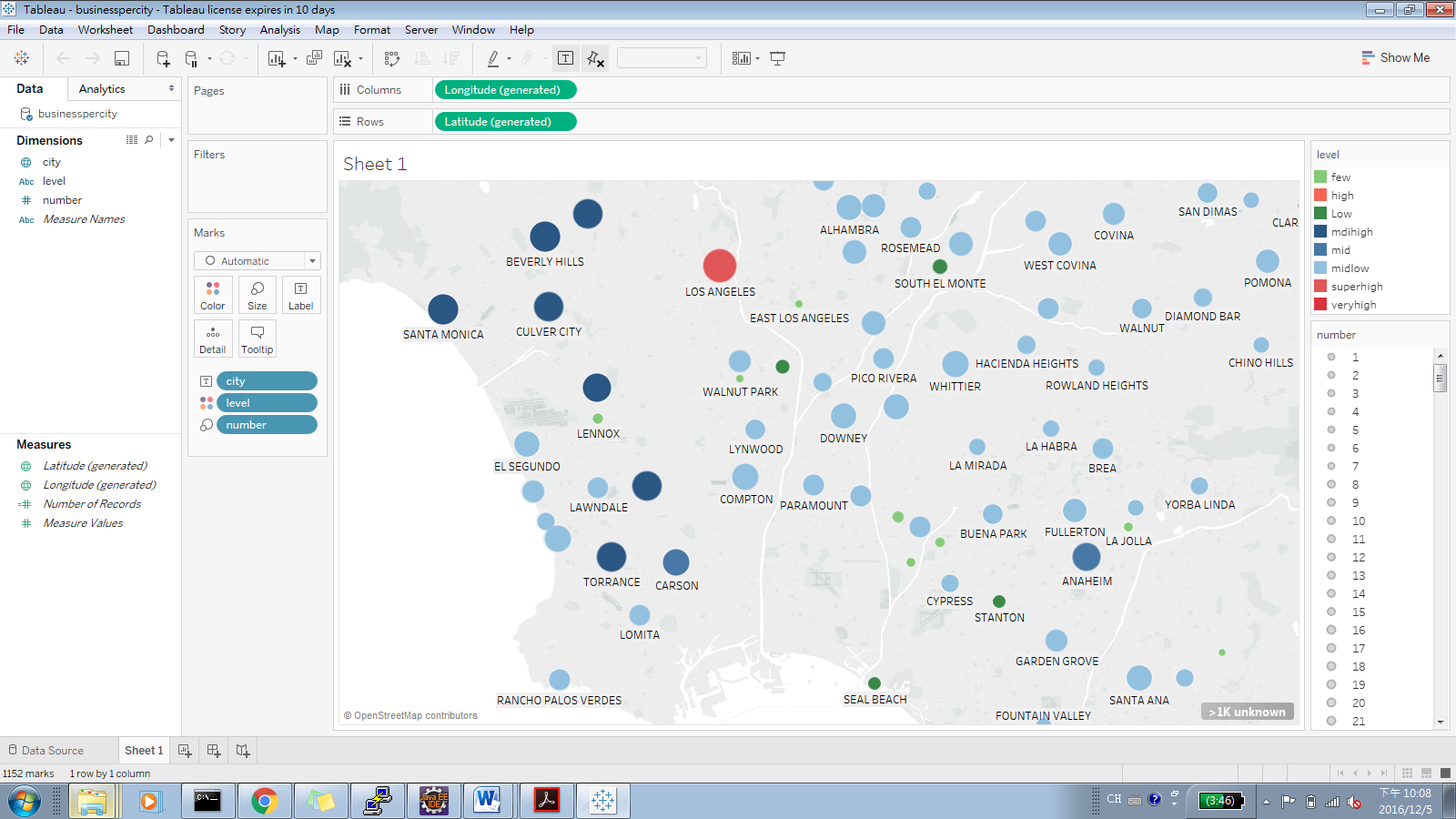
* 20067Price



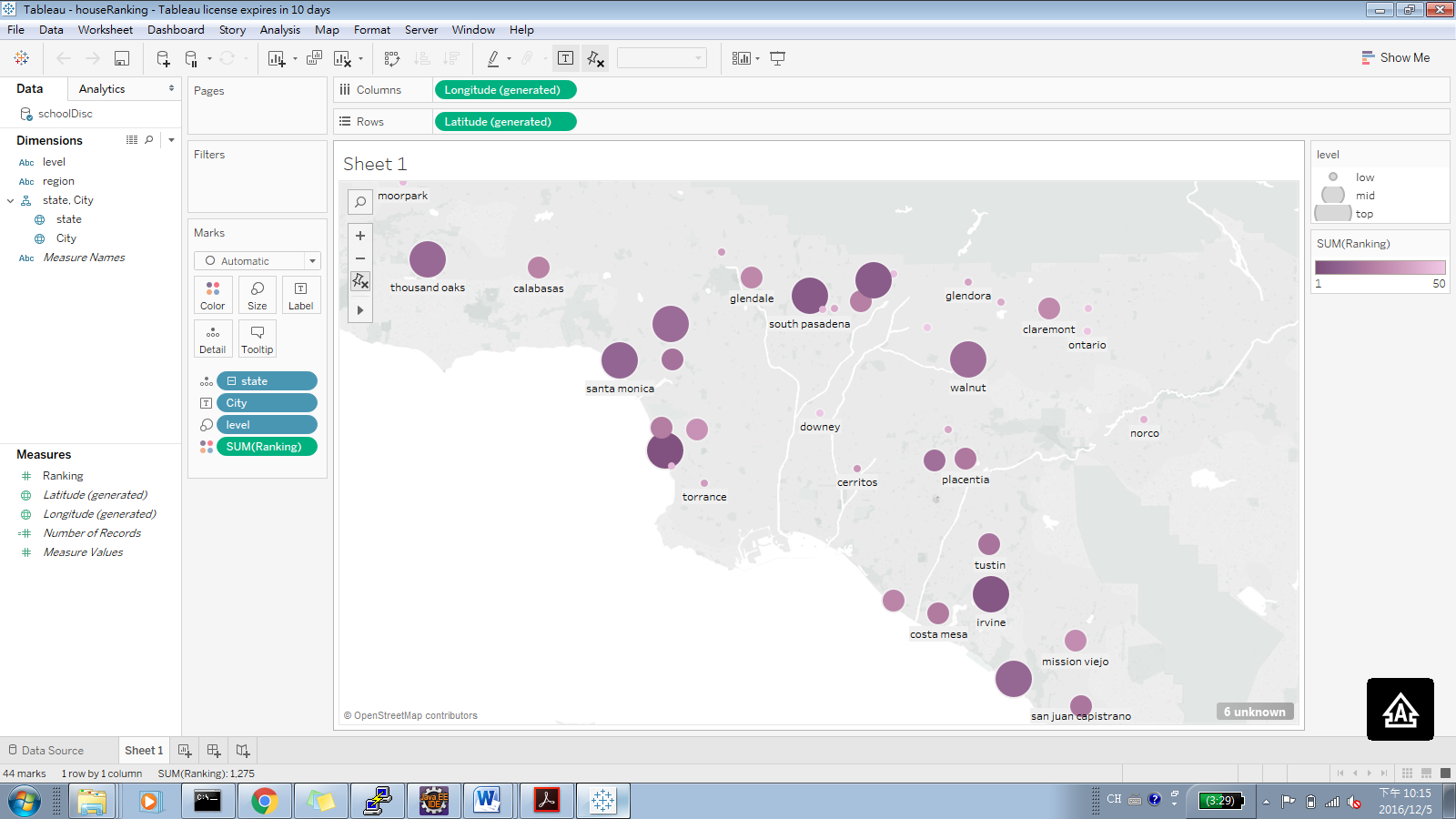
* Income



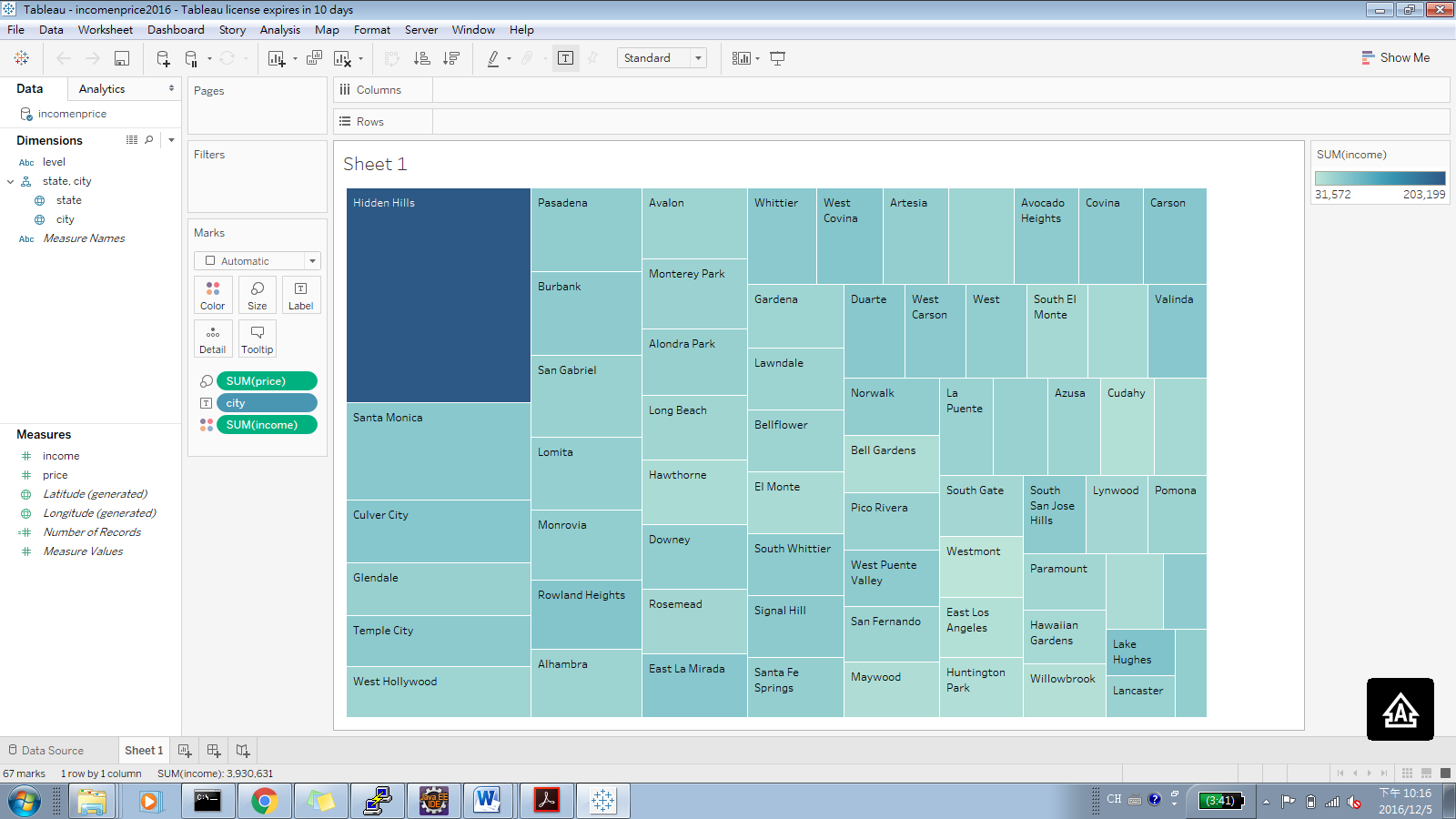
* businesspercity



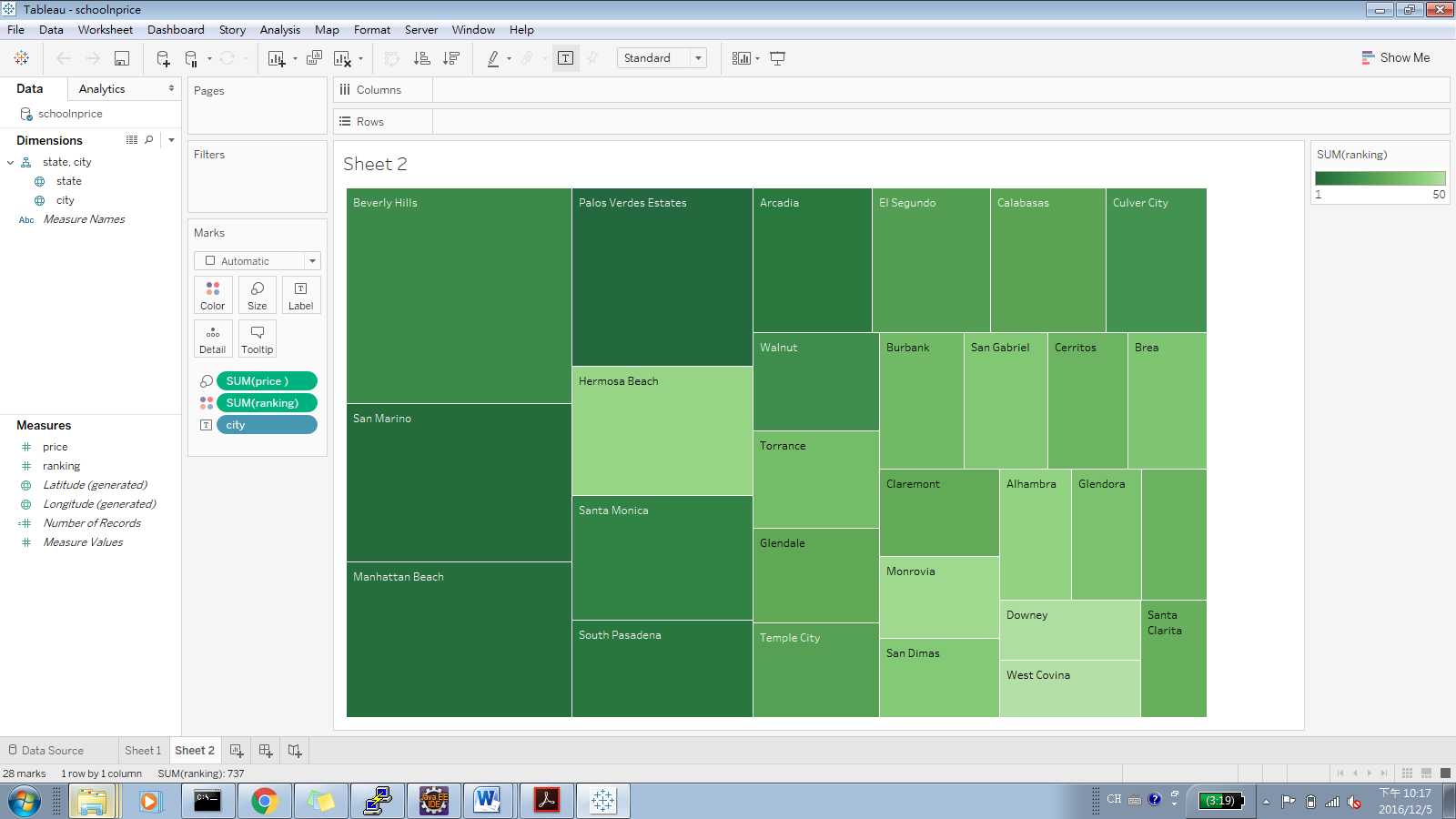
* housingRanking



* incomenprice



* schoolnprice



**Reference**

1."LAPD Crime and Collision Raw Data for 2015 | Los Angeles - Open Data Portal." *LAPD Crime and Collision Raw Data for 2015 | Los Angeles - Open Data Portal*. N.p., n.d. Web. 01 Dec. 2016.

2."Housing Affordability Data System (HADS)." *Housing Affordability Data System (HADS) - Data.gov*. Publisher US Department of Housing and Urban Development, 15 Mar. 2015. Web. 01 Dec. 2016.

3.Zillow, Inc. "Los Angeles County CA Home Prices & Home Values | Zillow." *Zillow*. N.p., n.d. Web. 01 Dec. 2016.

4."2017 Best School Districts in Los Angeles Area." *Niche*. N.p., n.d. Web. 01 Dec. 2016.

5."Housing Affordability Data System (HADS)." *Housing Affordability Data System (HADS) - Data.gov*. Publisher US Department of Housing and Urban Development, 15 Mar. 2015. Web. 01 Dec. 2016.

6.Research on 1/24/2013, By Zillow. "Zillow Home Value Forecast: Methodology - Zillow Research." *Zillow Research*. N.p., 09 May 2016. Web. 01 Dec. 2016.

**This is the end of the tutorial**