CIS5200 Term Project Tutorial

Authors: Zalak Patel, Ayushi Porwal, Clifford Lin, Kajal Bhandare

Instructor: Jongwook Woo

Date: 12/14/2023

Lab Tutorial

Zalak Patel (zpatel6@calstatela.edu)

Ayushi Porwal (aporwal@calstatela.edu)

Clifford Lin (clin22@calstatela.edu)

Kajal Bhandare (kbhanda3@calstatela.edu)

12/08/2023

Health Insurance Marketplace Analysis (Hive)

Objectives

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

- Download files from Kaggle
- Upload Zip Files to Hadoop File System (HDFS)
- Create data tables from uploaded CSV files
- Insert clean data into new tables and create views for export
- Export data and create visualization

Platform Spec

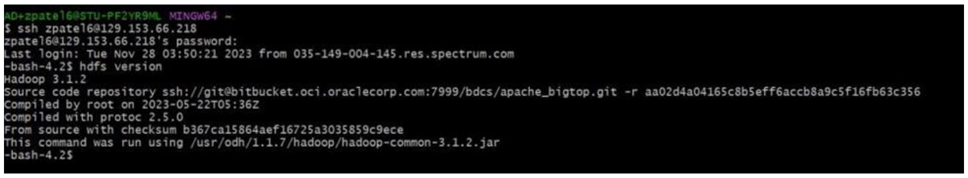
- Health Insurance Marketplace Dataset
- CPU Speed: 1995.312 mhz
- # of CPU cores: 8
- # of nodes: 5 (2 master & 3 data nodes)
- Total Memory Size: 367.68 GB

To Find Cluster Details Execute the below Commands:

CLUSTER VERSION: Hadoop 3.1.2

Command: `hdfs version`

- Give information about Hadoop cluster version

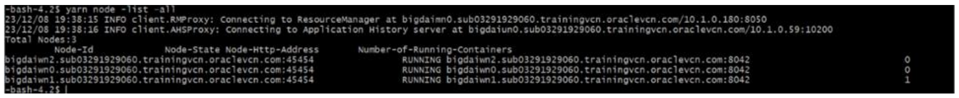


```
AD+zpatel6@STU-PF2YR9ML MINGW64 ~
$ ssh zpatel6@129.153.66.218
zpatel6@129.153.66.218's password:
Last login: Tue Nov 28 03:50:21 2023 from 035-149-004-145.res.spectrum.com
-bash-4.2$ hdfs version
Hadoop 3.1.2
Source code repository ssh://git@bitbucket.oci.oraclecorp.com:7999/bdcs/apache_bigtop.git -r aa02d4a04165c8b5eff6accb8a9c5f16fb63c356
Compiled by root on 2023-05-22T05:36Z
Compiled with protoc 2.5.0
From source with checksum b367ca15864aef16725a3035859c9ece
This command was run using /usr/odh/1.1.7/hadoop/hadoop-common-3.1.2.jar
-bash-4.2$
```

CLUSTER NODES: 5 (2 master nodes & 3 data nodes)

Command: `yarn node -list –all`

- Give information about several working nodes, this command won't show information about data nodes.

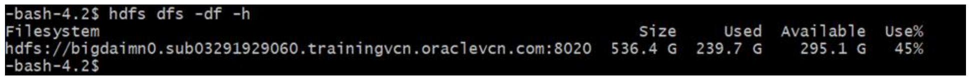


```
-bash-4.2$ yarn node -list -all
23/12/08 19:38:15 INFO client.RMProxy: Connecting to ResourceManager at bigdaimn0.sub03291929060.trainingvcn.oraclevcn.com/10.1.0.180:8050
23/12/08 19:38:16 INFO client.AHSProxy: Connecting to Application History server at bigdaiun0.sub03291929060.trainingvcn.oraclevcn.com/10.1.0.59:10200
Total Nodes:3
         Node-Id             Node-State Node-Http-Address       Number-of-Running-Containers
bigdaiwn2.sub03291929060.trainingvcn.oraclevcn.com:45454                RUNNING bigdaiwn2.sub03291929060.trainingvcn.oraclevcn.com:8042                 0
bigdaiwn0.sub03291929060.trainingvcn.oraclevcn.com:45454                RUNNING bigdaiwn0.sub03291929060.trainingvcn.oraclevcn.com:8042                 0
bigdaiwn1.sub03291929060.trainingvcn.oraclevcn.com:45454                RUNNING bigdaiwn1.sub03291929060.trainingvcn.oraclevcn.com:8042                 1
-bash-4.2$ |
```

MEMORY SIZE: Memory Used – 367.68 GB, Memory Remaining – 20.96 GB

Command: `hdfs dfs -df -h`

- Give information about memory size (Used and remaining).

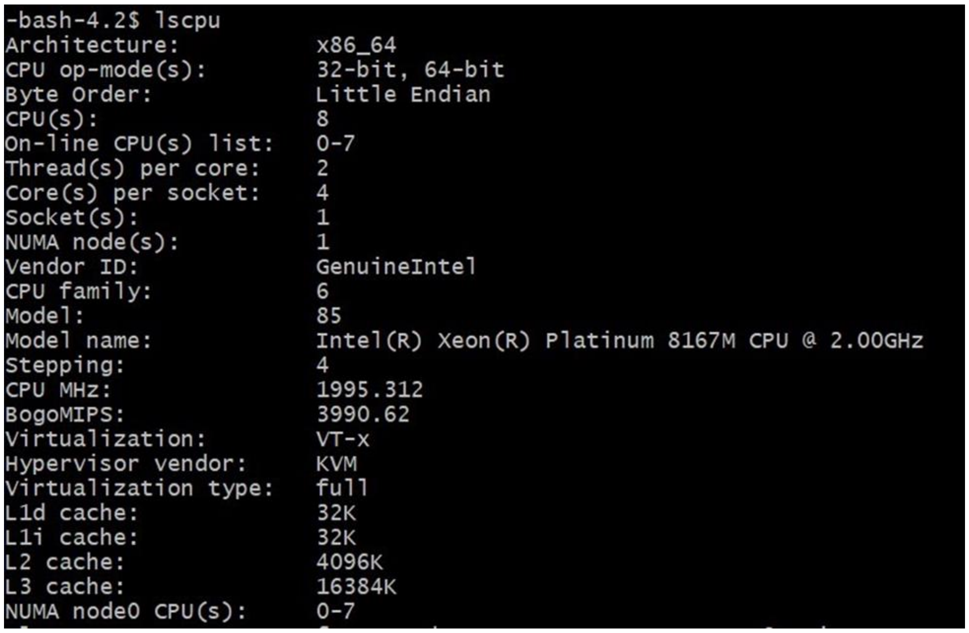


```
-bash-4.2$ hdfs dfs -df -h
Filesystem                                                   Size     Used  Available  Use%
hdfs://bigdaimn0.sub03291929060.trainingvcn.oraclevcn.com:8020  536.4 G  239.7 G    295.1 G   45%
-bash-4.2$
```

CPU Core: 1995.312 MHz

Command: `lscpu`

- Give information about CPU speed.

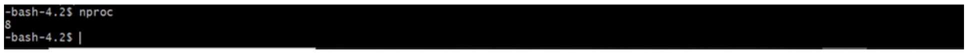


```
-bash-4.2$ lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                8
On-line CPU(s) list:   0-7
Thread(s) per core:    2
Core(s) per socket:    4
Socket(s):             1
NUMA node(s):          1
Vendor ID:             GenuineIntel
CPU family:            6
Model:                 85
Model name:            Intel(R) Xeon(R) Platinum 8167M CPU @ 2.00GHz
Stepping:              4
CPU MHz:               1995.312
BogoMIPS:              3990.62
Virtualization:        VT-x
Hypervisor vendor:     KVM
Virtualization type:   full
L1d cache:             32K
L1i cache:             32K
L2 cache:              4096K
L3 cache:              16384K
NUMA node0 CPU(s):     0-7
```

Number of CPU Cores: 8

Command: `nproc`

- Give information about CPU core.

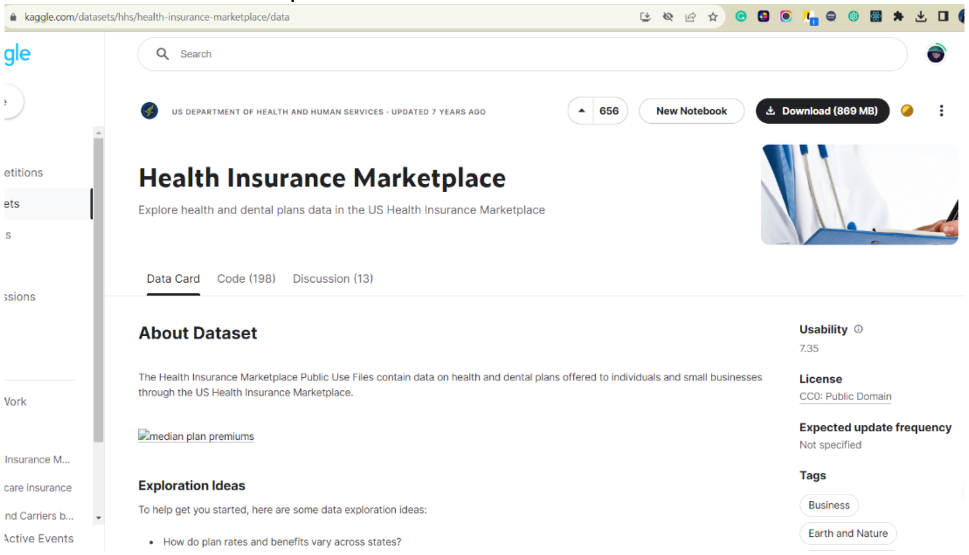


```
-bash-4.2$ nproc
8
-bash-4.2$ |
```

Step 1: Download the data from Kaggle

This step is to get data manually to the local system. Following are the steps to download:

1. Health Insurance Marketplace - Download dataset to local machine, click on the download button to download a zip file.



kaggle.com/datasets/hhs/health-insurance-marketplace/data

gle

Search

US DEPARTMENT OF HEALTH AND HUMAN SERVICES · UPDATED 7 YEARS AGO

656 New Notebook Download (869 MB)

etitions
ets
s
ssions
Vork
Insurance M...
care insurance
nd Carriers b...
Active Events

Health Insurance Marketplace

Explore health and dental plans data in the US Health Insurance Marketplace

Data Card Code (198) Discussion (13)

About Dataset

The Health Insurance Marketplace Public Use Files contain data on health and dental plans offered to individuals and small businesses through the US Health Insurance Marketplace.

median plan premiums

Exploration Ideas

To help get you started, here are some data exploration ideas:

- How do plan rates and benefits vary across states?

Usability
7.35

License
CC0: Public Domain

Expected update frequency
Not specified

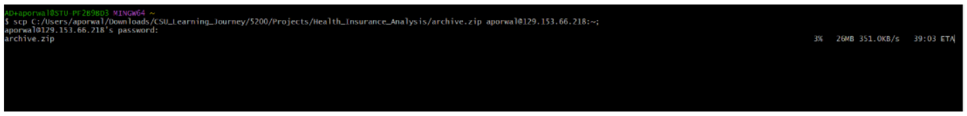
Tags
Business
Earth and Nature

Step 2: Upload Zip Files to the Hadoop File System

Upload and unzip file in the Linux file server. Create new directories in Hadoop, then put the unzipped files into their respective directories.

1. Copy the archive.zip file to the remote server.

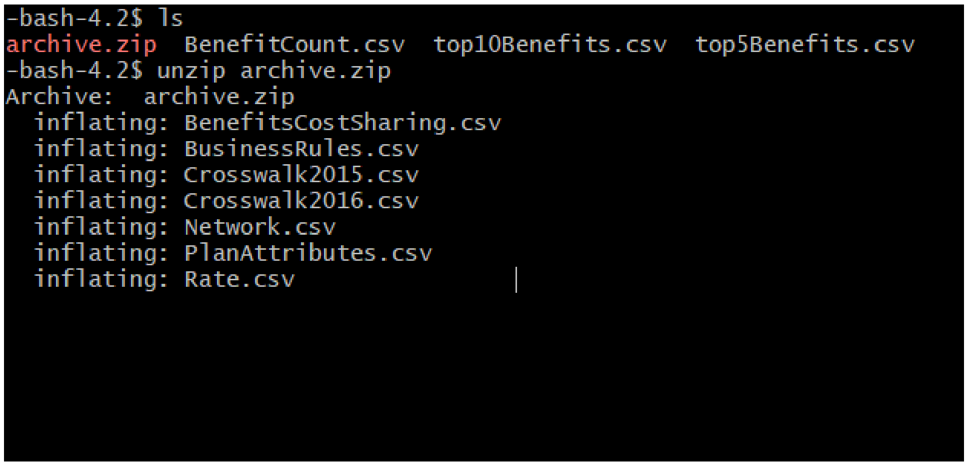
```
scp
C:/Users/aporwal/Downloads/CSU_Learning_Journey/5200/Projects/Hea
lth_Insurance_Analysis/archive.zip aporwal@129.153.66.218:~;
```



```
AD+aporwal@STU-PF2B9BD3 MINGW64 ~
$ scp C:/Users/aporwal/Downloads/CSU_Learning_Journey/5200/Projects/Health_Insurance_Analysis/archive.zip aporwal@129.153.66.218:~;
aporwal@129.153.66.218's password:
archive.zip                                                                              3%   26MB 351.0KB/s   39:03 ETA
```

2. Unzip the file

```
ssh aporwal@129.153.66.218;

unzip archive.zip;
```



```
-bash-4.2$ ls
archive.zip  BenefitCount.csv  top10Benefits.csv  top5Benefits.csv
-bash-4.2$ unzip archive.zip
Archive:  archive.zip
  inflating: BenefitsCostSharing.csv
  inflating: BusinessRules.csv
  inflating: Crosswalk2015.csv
  inflating: Crosswalk2016.csv
  inflating: Network.csv
  inflating: PlanAttributes.csv
  inflating: Rate.csv
```

3. Create directories in HDFS

```
hdfs dfs -mkdir project;
hdfs dfs -mkdir project/RawData;
hdfs dfs -mkdir project/CleanedData;
hdfs dfs -mkdir project/CleanedData/Benefitscostsharing;
hdfs dfs -mkdir project/CleanedData/Network;
hdfs dfs -mkdir project/CleanedData/Rate
hdfs dfs -mkdir project/CleanedData/Plan;
```

4. Put the files into the RawData directory

```
hdfs dfs -put BenefitsCostSharing.csv project/RawData/;
hdfs dfs -put PlanAttributes.csv project/RawData/;
hdfs dfs -put Network.csv project/RawData/;
hdfs dfs -put Rate.csv project/RawData/;
```

Step 3: Create data tables from uploaded CSV files

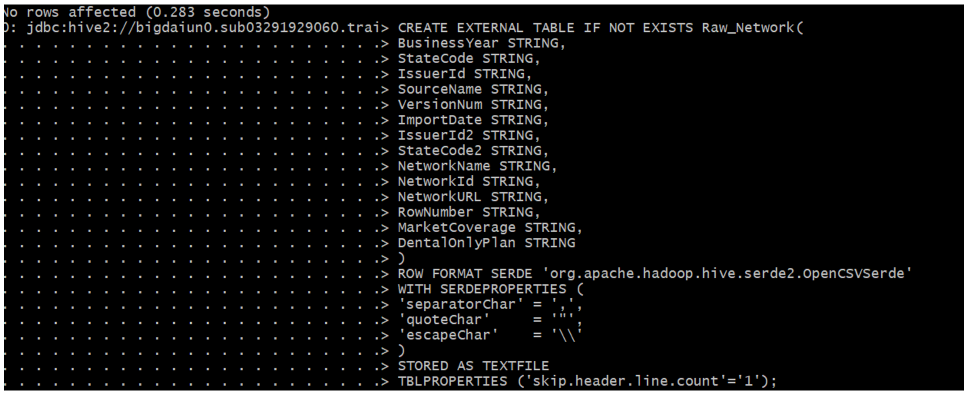
Create data table definitions using the SERDE command to preserve the text format, then populate those tables using data from the CSV files in Hadoop.

1. Login to Hadoop cluster using IP address "129.153.66.218"
   ```
   ssh aprowal@129.153.66.218
   ```

2. Use 'beeline' execute commands in Hive for analysis:
   ```
   Beeline;
   ```

3. Use below command to use your database:
   ```
   USE aporwal;
   ```

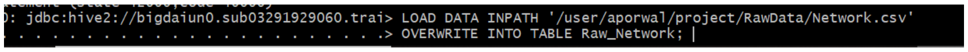
4. Create tables definitions for Raw_Networks
   ```
   CREATE EXTERNAL TABLE IF NOT EXISTS Raw_Network(
   BusinessYear STRING,
   StateCode STRING,
   IssuerId STRING,
   SourceName STRING,
   VersionNum STRING,
   ImportDate STRING,
   IssuerId2 STRING,
   StateCode2 STRING,
   NetworkName STRING,
   NetworkId STRING,
   NetworkURL STRING,
   RowNumber STRING,
   MarketCoverage STRING,
   DentalOnlyPlan STRING)
   ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
   WITH SERDEPROPERTIES (
   'separatorChar' = ',',
   'quoteChar'     = '"',
   'escapeChar'    = '\\')
   STORED AS TEXTFILE
   TBLPROPERTIES ('skip.header.line.count'='1');
   ```



```
No rows affected (0.283 seconds)
0: jdbc:hive2://bigdaiun0.sub03291929060.trai> CREATE EXTERNAL TABLE IF NOT EXISTS Raw_Network(
. . . . . . . . . . . . . . . . . . . . . . .> BusinessYear STRING,
. . . . . . . . . . . . . . . . . . . . . . .> StateCode STRING,
. . . . . . . . . . . . . . . . . . . . . . .> IssuerId STRING,
. . . . . . . . . . . . . . . . . . . . . . .> SourceName STRING,
. . . . . . . . . . . . . . . . . . . . . . .> VersionNum STRING,
. . . . . . . . . . . . . . . . . . . . . . .> ImportDate STRING,
. . . . . . . . . . . . . . . . . . . . . . .> IssuerId2 STRING,
. . . . . . . . . . . . . . . . . . . . . . .> StateCode2 STRING,
. . . . . . . . . . . . . . . . . . . . . . .> NetworkName STRING,
. . . . . . . . . . . . . . . . . . . . . . .> NetworkId STRING,
. . . . . . . . . . . . . . . . . . . . . . .> NetworkURL STRING,
. . . . . . . . . . . . . . . . . . . . . . .> RowNumber STRING,
. . . . . . . . . . . . . . . . . . . . . . .> MarketCoverage STRING,
. . . . . . . . . . . . . . . . . . . . . . .> DentalOnlyPlan STRING
. . . . . . . . . . . . . . . . . . . . . . .> )
. . . . . . . . . . . . . . . . . . . . . . .> ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
. . . . . . . . . . . . . . . . . . . . . . .> WITH SERDEPROPERTIES (
. . . . . . . . . . . . . . . . . . . . . . .> 'separatorChar' = ',',
. . . . . . . . . . . . . . . . . . . . . . .> 'quoteChar'     = '"',
. . . . . . . . . . . . . . . . . . . . . . .> 'escapeChar'    = '\\'
. . . . . . . . . . . . . . . . . . . . . . .> )
. . . . . . . . . . . . . . . . . . . . . . .> STORED AS TEXTFILE
. . . . . . . . . . . . . . . . . . . . . . .> TBLPROPERTIES ('skip.header.line.count'='1');
```

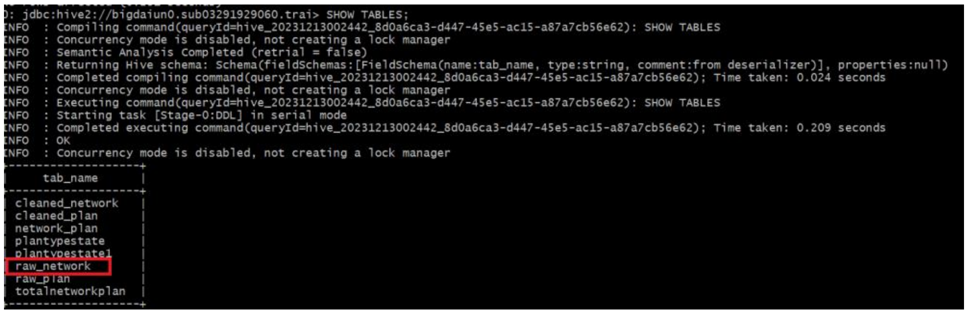
5. Load data into `Raw_Network`

```
LOAD DATA INPATH '/user/aporwal/project/RawData/network.csv';
OVERWRITE INTO TABLE Raw_Network;
```



```
0: jdbc:hive2://bigdaiun0.sub03291929060.trai> LOAD DATA INPATH '/user/aporwal/project/RawData/Network.csv'
. . . . . . . . . . . . . . . . . . . . . . .> OVERWRITE INTO TABLE Raw_Network; |
```

```
SHOW TABLES;
```



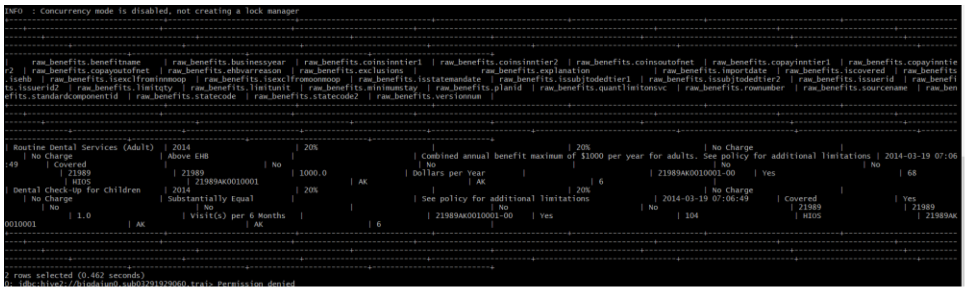
```
0: jdbc:hive2://bigdaiun0.sub03291929060.trai> SHOW TABLES;
INFO  : Compiling command(queryId=hive_20231213002442_8d0a6ca3-d447-45e5-ac15-a87a7cb56e62): SHOW TABLES
INFO  : Concurrency mode is disabled, not creating a lock manager
INFO  : Semantic Analysis Completed (retrial = false)
INFO  : Returning Hive schema: Schema(fieldSchemas:[FieldSchema(name:tab_name, type:string, comment:from deserializer)], properties:null)
INFO  : Completed compiling command(queryId=hive_20231213002442_8d0a6ca3-d447-45e5-ac15-a87a7cb56e62); Time taken: 0.024 seconds
INFO  : Concurrency mode is disabled, not creating a lock manager
INFO  : Executing command(queryId=hive_20231213002442_8d0a6ca3-d447-45e5-ac15-a87a7cb56e62): SHOW TABLES
INFO  : Starting task [Stage-0:DDL] in serial mode
INFO  : Completed executing command(queryId=hive_20231213002442_8d0a6ca3-d447-45e5-ac15-a87a7cb56e62); Time taken: 0.209 seconds
INFO  : OK
INFO  : Concurrency mode is disabled, not creating a lock manager
+-------------------+
|     tab_name      |
+-------------------+
| cleaned_network   |
| cleaned_plan      |
| network_plan      |
| plantypestate     |
| plantypestate1    |
| raw_network       |
| raw_plan          |
| totalnetworkplan  |
+-------------------+
```

6. Create table definitions for Raw_Benefits

```
CREATE EXTERNAL TABLE IF NOT EXISTS Raw_Benefits(
BenefitName STRING,
BusinessYear STRING,
CoinsInnTier1 STRING,
CoinsInnTier2 STRING,
CoinsOutofNet STRING,
CopayInnTier1 STRING,
CopayInnTier2 STRING,
CopayOutofNet STRING,
EHBVarReason STRING,
Exclusions STRING,
Explanation STRING,
ImportDate STRING,
IsCovered STRING,
IsEHB STRING,
IsExclFromInnMOOP STRING,
IsExclFromOonMOOP STRING,
IsStateMandate STRING,
IsSubjToDedTier1 STRING,
IsSubjToDedTier2 STRING,
IssuerId STRING,
IssuerId2 STRING,
LimitQty STRING,
LimitUnit STRING,
MinimumStay STRING,
PlanId STRING,
QuantLimitOnSvc STRING,
RowNumber STRING,
SourceName STRING,
StandardComponentId STRING,
StateCode STRING,
StateCode2 STRING,
VersionNum STRING)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
WITH SERDEPROPERTIES (
    'separatorChar' = ',',
    'quoteChar'     = '"',
    'escapeChar'    = '\\'
)
STORED AS TEXTFILE
TBLPROPERTIES ('skip.header.line.count'='1');
```

7. Load data into `Raw_Benefits`

```
LOAD DATA INPATH
'/user/aporwal/project/RawData/BenefitsCostSharing.csv'
OVERWRITE INTO TABLE Raw_Benefits;

show tables;

SELECT * FROM Raw_Benefits LIMIT 2;
```



```
INFO  : Concurrency mode is disabled, not creating a lock manager
+-----------------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------
|     raw_benefits.benefitname     | raw_benefits.businessyear  | raw_benefits.coinsinntier1  | raw_benefits.coinsinntier2  | raw_benefits.coinsoutofnet  | raw_benefits.copayinntier1  | raw_benefits.copayinntie
r2  | raw_benefits.copayoutofnet  | raw_benefits.ehbvarreason  | raw_benefits.exclusions  |              raw_benefits.explanation              | raw_benefits.importdate  | raw_benefits.iscovered  | raw_benefits
.isehb  | raw_benefits.isexclfrominnmoop  | raw_benefits.isexclfromoonmoop  | raw_benefits.isstatemandate  | raw_benefits.issubjtodedtier1  | raw_benefits.issubjtodedtier2  | raw_benefits.issuerid  | raw_benefi
ts.issuerid2  | raw_benefits.limitqty  | raw_benefits.limitunit  | raw_benefits.minimumstay  | raw_benefits.planid  | raw_benefits.quantlimitonsvc  | raw_benefits.rownumber  | raw_benefits.sourcename  | raw_ben
efits.standardcomponentid  | raw_benefits.statecode  | raw_benefits.statecode2  | raw_benefits.versionnum  |
+-----------------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------
| Routine Dental Services (Adult)  | 2014                        | 20%                         |                             | 20%                         | No Charge                   |
    | No Charge                   | Above EHB                   |                             | Combined annual benefit maximum of $1000 per year for adults. See policy for additional limitations | 2014-03-19 07:06
:49      | Covered                 |                             | No                          | No                          |                             | No                          | No
          | 21989                 | 21989                       | 1000.0                      | Dollars per Year            |                             | 21989AK0010001-00    | Yes                         | 68
           | HIOS                 | 21989AK0010001              | AK                          | AK                          | 6                           |
| Dental Check-Up for Children    | 2014                        | 20%                         |                             | 20%                         | No Charge                   |
    | No Charge                   | Substantially Equal         |                             | See policy for additional limitations           | 2014-03-19 07:06:49      | Covered                     | Yes
        | No                       | No                          |                             | No                          | No                          | 21989                       | 21989
            | 1.0                  | Visit(s) per 6 Months       |                             | 21989AK0010001-00    | Yes                         | 104                         | HIOS                        | 21989AK
0010001                | AK                          | AK                          | 6                           |
+-----------------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------------------------+-----------
2 rows selected (0.462 seconds)
0: jdbc:hive2://bigdaiun0.sub03291929060.trai> Permission denied
```

8. Create table definitions for Rates

```
CREATE EXTERNAL TABLE IF NOT EXISTS Rate(
BusinessYear STRING,
StateCode STRING,
IssuerId STRING,
SourceName STRING,
VersionNum STRING,
ImportDate STRING,
IssuerId2 STRING,
FederalTIN STRING,
RateEffectiveDate STRING,
RateExpirationDate STRING,
PlanId STRING,
RatingAreaId STRING,
Tobacco STRING,
Age STRING,
IndividualRate STRING,
IndividualTobaccoRate STRING,
Couple STRING,
PrimarySubscriberAndOneDependent STRING,
PrimarySubscriberAndTwoDependents STRING,
PrimarySubscriberAndThreeOrMoreDependents STRING,
CoupleAndOneDependent STRING,
CoupleAndTwoDependents STRING,
CoupleAndThreeOrMoreDependents STRING,
RowNumber STRING
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
WITH SERDEPROPERTIES (
   'separatorChar' = ',',
   'quoteChar'     = '"',
   'escapeChar'    = '\\'
)
STORED AS TEXTFILE
TBLPROPERTIES ('skip.header.line.count'='1');
```

9. Load data into Rates

```
LOAD DATA INPATH '/user/clin22/project/RawData/Rate.csv'
OVERWRITE INTO TABLE Rate;
```

10. Create table definition for raw_planAttr

```
CREATE EXTERNAL TABLE raw_planAttr (
AVCalculatorOutputNumber INT,
BeginPrimaryCareCostSharingAfterNumberOfVisits STRING,
BeginPrimaryCareDeductibleCoinsuranceAfterNumberOfCopays STRING,
BenefitPackageId STRING,
BusinessYear INT,
CSRVariationType STRING,
ChildOnlyOffering STRING,
ChildOnlyPlanId STRING,
CompositeRatingOffered STRING,
DEHBCombInnOonFamilyMOOP STRING,
DEHBCombInnOonFamilyPerGroupMOOP STRING,
DEHBCombInnOonFamilyPerPersonMOOP STRING,
DEHBCombInnOonIndividualMOOP STRING,
DEHBDedCombInnOonFamily STRING,
DEHBDedCombInnOonFamilyPerGroup STRING,
DEHBDedCombInnOonFamilyPerPerson STRING,
DEHBDedCombInnOonIndividual STRING,
DEHBDedInnTier1Coinsurance STRING,
DEHBDedInnTier1Family STRING,
DEHBDedInnTier1FamilyPerGroup STRING,
DEHBDedInnTier1FamilyPerPerson STRING,
DEHBDedInnTier1Individual STRING,
DEHBDedInnTier2Coinsurance STRING,
DEHBDedInnTier2Family STRING,
DEHBDedInnTier2FamilyPerGroup STRING,
DEHBDedInnTier2FamilyPerPerson STRING,
DEHBDedInnTier2Individual STRING,
DEHBDedOutOfNetFamily STRING,
DEHBDedOutOfNetFamilyPerGroup STRING,
DEHBDedOutOfNetFamilyPerPerson STRING,
DEHBDedOutOfNetIndividual STRING,
DEHBInnTier1FamilyMOOP STRING,
DEHBInnTier1FamilyPerGroupMOOP STRING,
DEHBInnTier1FamilyPerPersonMOOP STRING,
DEHBInnTier1IndividualMOOP STRING,
DEHBInnTier2FamilyMOOP STRING,
DEHBInnTier2FamilyPerGroupMOOP STRING,
DEHBInnTier2FamilyPerPersonMOOP STRING,
DEHBInnTier2IndividualMOOP STRING,
DEHBOutOfNetFamilyMOOP STRING,
DEHBOutOfNetFamilyPerGroupMOOP STRING,
DEHBOutOfNetFamilyPerPersonMOOP STRING,
DEHBOutOfNetIndividualMOOP STRING,
DentalOnlyPlan STRING,
DiseaseManagementProgramsOffered STRING,
EHBPediatricDentalApportionmentQuantity STRING,
EHBPercentPremiumS4 STRING,
EHBPercentTotalPremium STRING,
FirstTierUtilization STRING,
FormularyId STRING,
FormularyURL STRING,
HIOSProductId STRING,
HPID STRING,
HSAOrHRAEmployerContribution STRING,
HSAOrHRAEmployerContributionAmount STRING,
ImportDate STRING,
IndianPlanVariationEstimatedAdvancedPaymentAmountPerEnrollee
STRING,
InpatientCopaymentMaximumDays STRING,
IsGuaranteedRate STRING,
IsHSAEligible STRING,
IsNewPlan STRING,
IsNoticeRequiredForPregnancy STRING,
IsReferralRequiredForSpecialist STRING,
IssuerActuarialValue STRING,
IssuerId STRING,
IssuerId2 STRING,
MEHBCombInnOonFamilyMOOP STRING,
MEHBCombInnOonFamilyPerGroupMOOP STRING,
MEHBCombInnOonFamilyPerPersonMOOP STRING,
MEHBCombInnOonIndividualMOOP STRING,
MEHBDedCombInnOonFamily STRING,
MEHBDedCombInnOonFamilyPerGroup STRING,
MEHBDedCombInnOonFamilyPerPerson STRING,
MEHBDedCombInnOonIndividual STRING,
MEHBDedInnTier1Coinsurance STRING,
MEHBDedInnTier1Family STRING,
MEHBDedInnTier1FamilyPerGroup STRING,
MEHBDedInnTier1FamilyPerPerson STRING,
MEHBDedInnTier1Individual STRING,
MEHBDedInnTier2Coinsurance STRING,
MEHBDedInnTier2Family STRING,
MEHBDedInnTier2FamilyPerGroup STRING,
MEHBDedInnTier2FamilyPerPerson STRING,
MEHBDedInnTier2Individual STRING,
MEHBDedOutOfNetFamily STRING,
MEHBDedOutOfNetFamilyPerGroup STRING,
MEHBDedOutOfNetFamilyPerPerson STRING,
MEHBDedOutOfNetIndividual STRING,
MEHBInnTier1FamilyMOOP STRING,
MEHBInnTier1FamilyPerGroupMOOP STRING,
MEHBInnTier1FamilyPerPersonMOOP STRING,
MEHBInnTier1IndividualMOOP STRING,
MEHBInnTier2FamilyMOOP STRING,
MEHBInnTier2FamilyPerGroupMOOP STRING,
MEHBInnTier2FamilyPerPersonMOOP STRING,
MEHBInnTier2IndividualMOOP STRING,
MEHBOutOfNetFamilyMOOP STRING,
MEHBOutOfNetFamilyPerGroupMOOP STRING,
MEHBOutOfNetFamilyPerPersonMOOP STRING,
MEHBOutOfNetIndividualMOOP STRING,
MarketCoverage STRING,
MedicalDrugDeductiblesIntegrated STRING,
MedicalDrugMaximumOutofPocketIntegrated STRING,
MetalLevel STRING,
MultipleInNetworkTiers STRING,
NationalNetwork STRING,
NetworkId STRING,
OutOfCountryCoverage STRING,
OutOfCountryCoverageDescription STRING,
OutOfServiceAreaCoverage STRING,
OutOfServiceAreaCoverageDescription STRING,
PlanBrochure STRING,
PlanEffictiveDate STRING,
PlanExpirationDate STRING,
PlanId STRING,
PlanLevelExclusions STRING,
PlanMarketingName STRING,
PlanType STRING,
QHPNonQHPTypeId STRING,
RowNumber STRING,
SBCHavingDiabetesCoinsurance STRING,
SBCHavingDiabetesCopayment STRING,
SBCHavingDiabetesDeductible STRING,
SBCHavingDiabetesLimit STRING,
SBCHavingaBabyCoinsurance STRING,
SBCHavingaBabyCopayment STRING,
SBCHavingaBabyDeductible STRING,
SBCHavingaBabyLimit STRING,
SecondTierUtilization STRING,
ServiceAreaId STRING,
SourceName STRING,
SpecialistRequiringReferral STRING,
SpecialtyDrugMaximumCoinsurance STRING,
StandardComponentId STRING,
StateCode STRING,
StateCode2 STRING,
TEHBCombInnOonFamilyMOOP STRING,
TEHBCombInnOonFamilyPerGroupMOOP STRING,
TEHBCombInnOonFamilyPerPersonMOOP STRING,
TEHBCombInnOonIndividualMOOP STRING,
TEHBDedCombInnOonFamily STRING,
TEHBDedCombInnOonFamilyPerGroup STRING,
TEHBDedCombInnOonFamilyPerPerson STRING,
TEHBDedCombInnOonIndividual STRING,
TEHBDedInnTier1Coinsurance STRING,
TEHBDedInnTier1Family STRING,
TEHBDedInnTier1FamilyPerGroup STRING,
TEHBDedInnTier1FamilyPerPerson STRING,
TEHBDedInnTier1Individual STRING,
TEHBDedInnTier2Coinsurance STRING,
TEHBDedInnTier2Family STRING,
TEHBDedInnTier2FamilyPerGroup STRING,
TEHBDedInnTier2FamilyPerPerson STRING,
TEHBDedInnTier2Individual STRING,
TEHBDedOutOfNetFamily STRING,
TEHBDedOutOfNetFamilyPerGroup STRING,
TEHBDedOutOfNetFamilyPerPerson STRING,
TEHBDedOutOfNetIndividual STRING,
TEHBInnTier1FamilyMOOP STRING,
TEHBInnTier1FamilyPerGroupMOOP STRING,
TEHBInnTier1FamilyPerPersonMOOP STRING,
TEHBInnTier1IndividualMOOP STRING,
TEHBInnTier2FamilyMOOP STRING,
TEHBInnTier2FamilyPerGroupMOOP STRING,
TEHBInnTier2FamilyPerPersonMOOP STRING,
TEHBInnTier2IndividualMOOP STRING,
TEHBOutOfNetFamilyMOOP STRING,
TEHBOutOfNetFamilyPerGroupMOOP STRING,
TEHBOutOfNetFamilyPerPersonMOOP STRING,
TEHBOutOfNetIndividualMOOP STRING,
TIN STRING,
URLForEnrollmentPayment STRING,
URLForSummaryofBenefitsCoverage STRING,
UniquePlanDesign STRING,
VersionNum STRING,
WellnessProgramOffered STRING
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
WITH SERDEPROPERTIES (
    'separatorChar' = ',',
    'quoteChar'     = '"',
    'escapeChar'    = '\\'
)
STORED AS TEXTFILE
TBLPROPERTIES ('skip.header.line.count'='1');
```

11. Load data into `raw_planAttr`

```
LOAD DATA INPATH
'/user/aporwal/project/RawData/PlanAttributes.csv'
OVERWRITE INTO TABLE Raw_PlanAttr;
```

Step 4: Insert Clean Data into Tables and Create Views to Export

Insert cleaned data into new tables and create necessary views for data export.

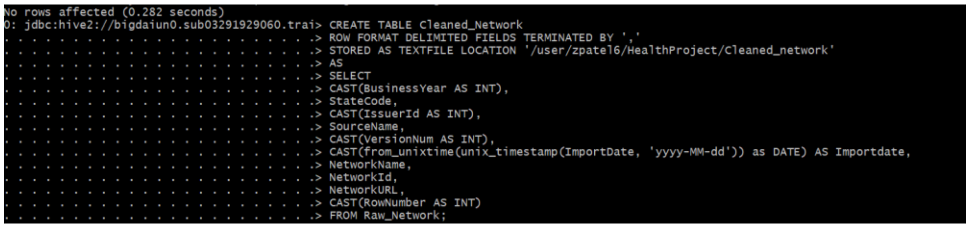
1. Create a `Cleaned_Network` table from `Raw_Network` table

2. Run the following hive query on a beeline to drop `cleaned_network` table if exists

   ```
   DROP TABLE IF EXISTS Cleaned_Network;
   ```

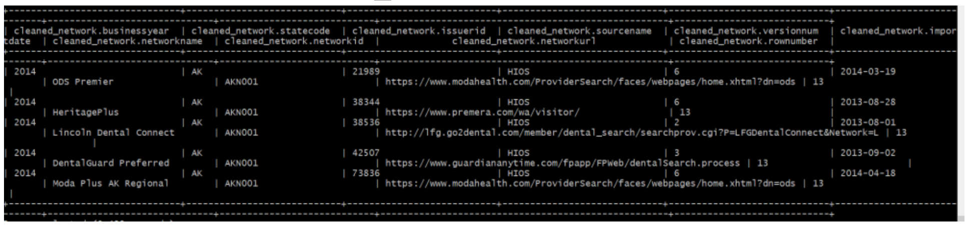
3. Execute below query to create `cleaned_network` table from `Raw_Network` table

   ```
   CREATE TABLE Cleaned_Network
   ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
   STORED AS TEXTFILE LOCATION
   '/user/zpatel6/HealthProject/Cleaned_network'
   AS
   SELECT
   CAST(BusinessYear AS INT),
   StateCode,
   CAST(IssuerId AS INT),
   SourceName,
   CAST(VersionNum AS INT),
   CAST(from_unixtime(unix_timestamp(ImportDate, 'yyyy-MM-dd')) as
   DATE) AS Importdate,
   NetworkName,
   NetworkId,
   NetworkURL,
   CAST(RowNumber AS INT)
   FROM Raw_Network;
   ```



```
No rows affected (0.282 seconds)
0: jdbc:hive2://bigdaiun0.sub03291929060.trai> CREATE TABLE Cleaned_Network
. . . . . . . . . . . . . . . . . . . . . . .> ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
. . . . . . . . . . . . . . . . . . . . . . .> STORED AS TEXTFILE LOCATION '/user/zpatel6/HealthProject/Cleaned_network'
. . . . . . . . . . . . . . . . . . . . . . .> AS
. . . . . . . . . . . . . . . . . . . . . . .> SELECT
. . . . . . . . . . . . . . . . . . . . . . .> CAST(BusinessYear AS INT),
. . . . . . . . . . . . . . . . . . . . . . .> StateCode,
. . . . . . . . . . . . . . . . . . . . . . .> CAST(IssuerId AS INT),
. . . . . . . . . . . . . . . . . . . . . . .> SourceName,
. . . . . . . . . . . . . . . . . . . . . . .> CAST(VersionNum AS INT),
. . . . . . . . . . . . . . . . . . . . . . .> CAST(from_unixtime(unix_timestamp(ImportDate, 'yyyy-MM-dd')) as DATE) AS Importdate,
. . . . . . . . . . . . . . . . . . . . . . .> NetworkName,
. . . . . . . . . . . . . . . . . . . . . . .> NetworkId,
. . . . . . . . . . . . . . . . . . . . . . .> NetworkURL,
. . . . . . . . . . . . . . . . . . . . . . .> CAST(RowNumber AS INT)
. . . . . . . . . . . . . . . . . . . . . . .> FROM Raw_Network;
```

   ```
   SELECT * FROM Cleaned_Network LIMIT 5;
   ```



```
| cleaned_network.businessyear  | cleaned_network.statecode  | cleaned_network.issuerid  | cleaned_network.sourcename  | cleaned_network.versionnum  | cleaned_network.impor
tdate  | cleaned_network.networkname  | cleaned_network.networkid  |           cleaned_network.networkurl           | cleaned_network.rownumber  |
| 2014                          | AK                         | 21989                     | HIOS                        | 6                           | 2014-03-19
       | ODS Premier                  | AKN001                     | https://www.modahealth.com/ProviderSearch/faces/webpages/home.xhtml?dn=ods | 13
|
| 2014                          | AK                         | 38344                     | HIOS                        | 6                           | 2013-08-28
       | HeritagePlus                 | AKN001                     | https://www.premera.com/wa/visitor/             | 13                         |
| 2014                          | AK                         | 38536                     | HIOS                        | 2                           | 2013-08-01
       | Lincoln Dental Connect       | AKN001                     | http://lfg.go2dental.com/member/dental_search/searchprov.cgi?P=LFGDentalConnect&Network=L | 13
             |
| 2014                          | AK                         | 42507                     | HIOS                        | 3                           | 2013-09-02
       | DentalGuard Preferred        | AKN001                     | https://www.guardiananytime.com/fpapp/FPWeb/dentalSearch.process | 13                 |
| 2014                          | AK                         | 73836                     | HIOS                        | 6                           | 2014-04-18
       | Moda Plus AK Regional        | AKN001                     | https://www.modahealth.com/ProviderSearch/faces/webpages/home.xhtml?dn=ods | 13
|
```

4. Create `Cleaned_Rates` table from the `Rates` table

   ```
   drop table if exists Cleaned_Rate;
   ```

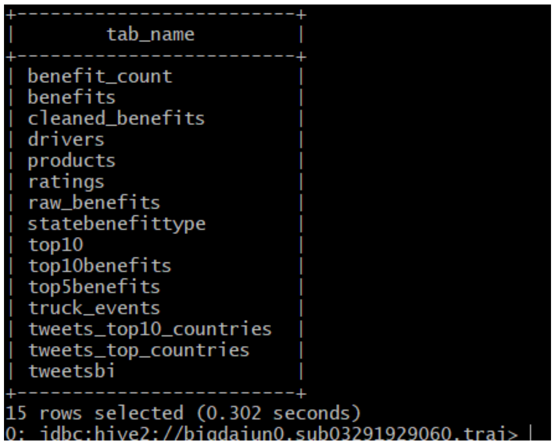
```
CREATE TABLE Cleaned_Rate
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
STORED AS TEXTFILE LOCATION
'/user/clin22/project/CleanedData/Rate'
AS
SELECT
CAST(BusinessYear AS INT),
StateCode,
CAST(IssuerId AS INT),
SourceName,
CAST(VersionNum AS INT),
CAST(from_unixtime(unix_timestamp(ImportDate, 'yyyy-MM-dd')) as
DATE) AS Importdate,
CAST(IssuerId2 AS INT),
FederalTIN,
CAST(from_unixtime(unix_timestamp(RateEffectiveDate, 'yyyy-MM-
dd')) as DATE) AS RateEffectiveDate,
CAST(from_unixtime(unix_timestamp(RateExpirationDate, 'yyyy-MM-
dd')) as DATE) AS RateExpirationDate,
PlanId,
RatingAreaId,
Tobacco,
Age,
CAST(IndividualRate AS DECIMAL),
IndividualTobaccoRate,
CAST(Couple AS DECIMAL(7,2)),
CAST(PrimarySubscriberAndOneDependent AS DECIMAL(7,2)),
CAST(PrimarySubscriberAndTwoDependents AS DECIMAL(7,2)),
CAST(PrimarySubscriberAndThreeOrMoreDependents AS DECIMAL(7,2)),
CAST(CoupleAndOneDependent AS DECIMAL(7,2)),
CAST(CoupleAndTwoDependents AS DECIMAL(7,2)),
CAST(CoupleAndThreeOrMoreDependents AS DECIMAL(7,2)),
CAST(RowNumber AS INT)
FROM Rate;
```

5. Create `Cleaned_Benefits` table definition

```
CREATE EXTERNAL TABLE Cleaned_Benefits(
BenefitName STRING,
BusinessYear INT,
EHBVarReason STRING,
Exclusions STRING,
Explanation STRING,
ImportDate DATE,
IsCovered BOOLEAN,
IsEHB BOOLEAN,
IsStateMandate BOOLEAN,
IsSubjToDedTier1 BOOLEAN,
IssuerId INT,
LimitQty INT,
LimitUnit STRING,
PlanId STRING,
RowNumber INT,
SourceName STRING,
StandardComponentId STRING,
StateCode STRING,

VersionNum INT);
```

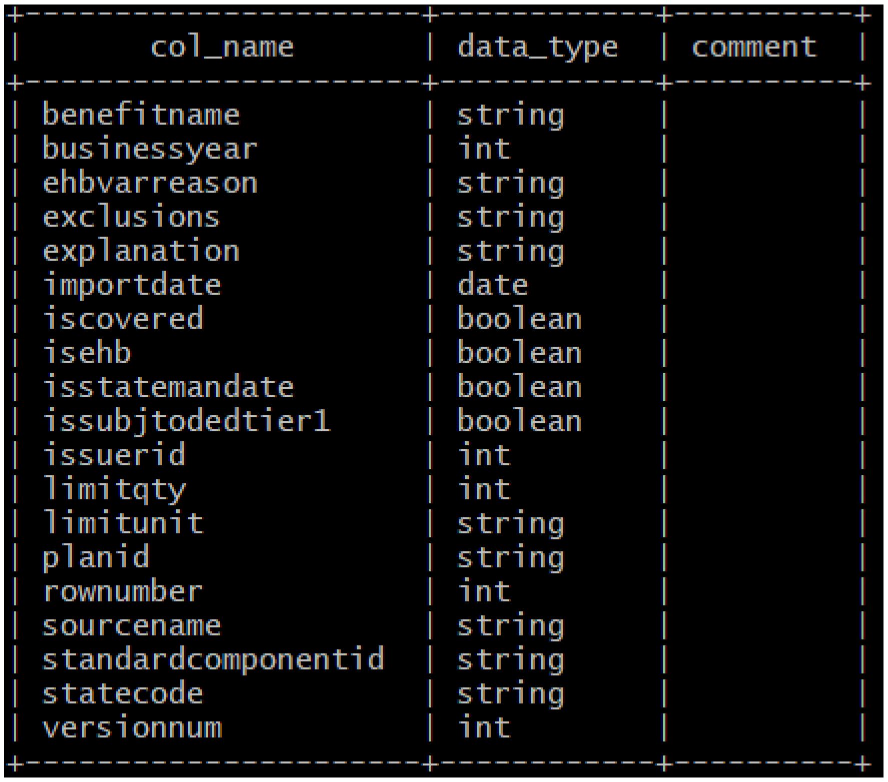
6. Insert data into `Cleaned_Benefits`

```
INSERT INTO TABLE Cleaned_Benefits

SELECT
    BenefitName,
    CAST(BusinessYear AS INT),
    EHBVarReason,
    Exclusions,
    Explanation,
    CAST(from_unixtime(unix_timestamp(ImportDate, 'yyyy-MM-dd'))
as DATE) AS Importdate,
    CAST(IsCovered AS BOOLEAN),
    CAST(IsEHB AS BOOLEAN),
    CAST(IsStateMandate AS BOOLEAN),
    CAST(IsSubjToDedTier1 AS BOOLEAN),
    CAST(IssuerId AS INT),
    CAST(LimitQty AS INT),
    LimitUnit,
    PlanId,
    CAST(RowNumber AS INT),
    SourceName,
    StandardComponentId,
    StateCode,
    CAST(VersionNum AS INT)
FROM Raw_Benefits;


show tables;
```



```
+-------------------------+
|        tab_name         |
+-------------------------+
| benefit_count           |
| benefits                |
| cleaned_benefits        |
| drivers                 |
| products                |
| ratings                 |
| raw_benefits            |
| statebenefittype        |
| top10                   |
| top10benefits           |
| top5benefits            |
| truck_events            |
| tweets_top10_countries  |
| tweets_top_countries    |
| tweetsbi                |
+-------------------------+
15 rows selected (0.302 seconds)
0: jdbc:hive2://bigdaiun0.sub03291929060.trai> 
```

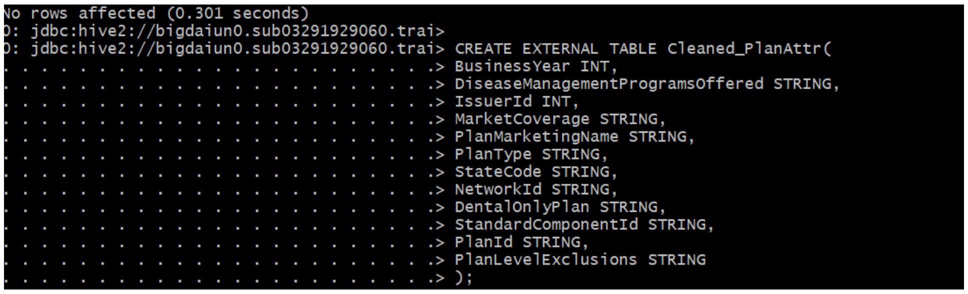
```
describe Cleaned_Benefits;
```



```
+-----------------------+------------+----------+
|       col_name        | data_type  | comment  |
+-----------------------+------------+----------+
| benefitname           | string     |          |
| businessyear          | int        |          |
| ehbvarreason          | string     |          |
| exclusions            | string     |          |
| explanation           | string     |          |
| importdate            | date       |          |
| iscovered             | boolean    |          |
| isehb                 | boolean    |          |
| isstatemandate        | boolean    |          |
| issubjtodedtier1      | boolean    |          |
| issuerid              | int        |          |
| limitqty              | int        |          |
| limitunit             | string     |          |
| planid                | string     |          |
| rownumber             | int        |          |
| sourcename            | string     |          |
| standardcomponentid   | string     |          |
| statecode             | string     |          |
| versionnum            | int        |          |
+-----------------------+------------+----------+
```

7. Create an external `Cleaned_PlanAttr` table by executing below query

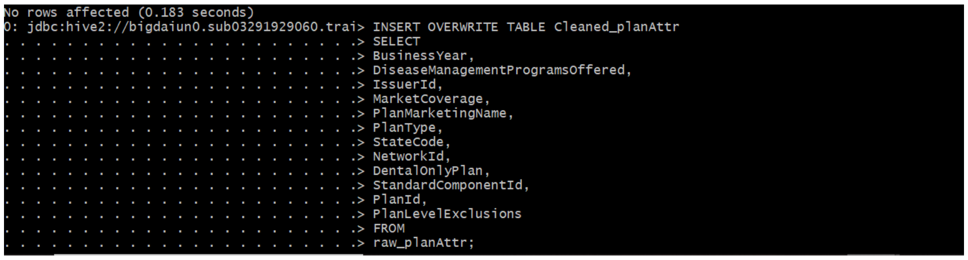
```
CREATE EXTERNAL TABLE Cleaned_PlanAttr(
BusinessYear INT,
DiseaseManagementProgramsOffered STRING,
IssuerId INT,
MarketCoverage STRING,
PlanMarketingName STRING,
PlanType STRING,
StateCode STRING,
NetworkId STRING,
DentalOnlyPlan STRING,
StandardComponentId STRING,
PlanId STRING,
PlanLevelExclusions STRING
);
```



```
No rows affected (0.301 seconds)
0: jdbc:hive2://bigdaiun0.sub03291929060.trai>
0: jdbc:hive2://bigdaiun0.sub03291929060.trai> CREATE EXTERNAL TABLE Cleaned_PlanAttr(
. . . . . . . . . . . . . . . . . . . . . . .> BusinessYear INT,
. . . . . . . . . . . . . . . . . . . . . . .> DiseaseManagementProgramsOffered STRING,
. . . . . . . . . . . . . . . . . . . . . . .> IssuerId INT,
. . . . . . . . . . . . . . . . . . . . . . .> MarketCoverage STRING,
. . . . . . . . . . . . . . . . . . . . . . .> PlanMarketingName STRING,
. . . . . . . . . . . . . . . . . . . . . . .> PlanType STRING,
. . . . . . . . . . . . . . . . . . . . . . .> StateCode STRING,
. . . . . . . . . . . . . . . . . . . . . . .> NetworkId STRING,
. . . . . . . . . . . . . . . . . . . . . . .> DentalOnlyPlan STRING,
. . . . . . . . . . . . . . . . . . . . . . .> StandardComponentId STRING,
. . . . . . . . . . . . . . . . . . . . . . .> PlanId STRING,
. . . . . . . . . . . . . . . . . . . . . . .> PlanLevelExclusions STRING
. . . . . . . . . . . . . . . . . . . . . . .> );
```

8. Insert data in the `Cleaned_PlanAttr` table from `Raw_PlanAttr` table

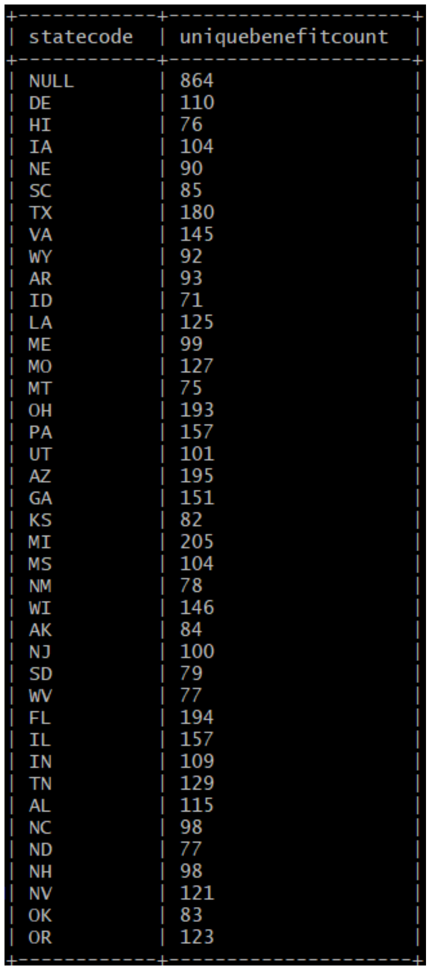
```
INSERT OVERWRITE TABLE Cleaned_planAttr
SELECT
BusinessYear,
DiseaseManagementProgramsOffered,
IssuerId,
MarketCoverage,
PlanMarketingName,
PlanType,
StateCode,
NetworkId,
DentalOnlyPlan,
StandardComponentId,
PlanId,
PlanLevelExclusions
FROM
raw_planAttr;
```



```
No rows affected (0.183 seconds)
0: jdbc:hive2://bigdaiun0.sub03291929060.trai> INSERT OVERWRITE TABLE Cleaned_planAttr
. . . . . . . . . . . . . . . . . . . . . . .> SELECT
. . . . . . . . . . . . . . . . . . . . . . .> BusinessYear,
. . . . . . . . . . . . . . . . . . . . . . .> DiseaseManagementProgramsOffered,
. . . . . . . . . . . . . . . . . . . . . . .> IssuerId,
. . . . . . . . . . . . . . . . . . . . . . .> MarketCoverage,
. . . . . . . . . . . . . . . . . . . . . . .> PlanMarketingName,
. . . . . . . . . . . . . . . . . . . . . . .> PlanType,
. . . . . . . . . . . . . . . . . . . . . . .> StateCode,
. . . . . . . . . . . . . . . . . . . . . . .> NetworkId,
. . . . . . . . . . . . . . . . . . . . . . .> DentalOnlyPlan,
. . . . . . . . . . . . . . . . . . . . . . .> StandardComponentId,
. . . . . . . . . . . . . . . . . . . . . . .> PlanId,
. . . . . . . . . . . . . . . . . . . . . . .> PlanLevelExclusions
. . . . . . . . . . . . . . . . . . . . . . .> FROM
. . . . . . . . . . . . . . . . . . . . . . .> raw_planAttr;
```

9. Create `Benefit_Count` view

```
CREATE VIEW Benefit_Count AS
SELECT StateCode, COUNT(DISTINCT BenefitName) AS
UniqueBenefitCount
FROM Cleaned_Benefits
GROUP BY StateCode;
```



```
+-----------+--------------------+
| statecode | uniquebenefitcount |
+-----------+--------------------+
| NULL      | 864                |
| DE        | 110                |
| HI        | 76                 |
| IA        | 104                |
| NE        | 90                 |
| SC        | 85                 |
| TX        | 180                |
| VA        | 145                |
| WY        | 92                 |
| AR        | 93                 |
| ID        | 71                 |
| LA        | 125                |
| ME        | 99                 |
| MO        | 127                |
| MT        | 75                 |
| OH        | 193                |
| PA        | 157                |
| UT        | 101                |
| AZ        | 195                |
| GA        | 151                |
| KS        | 82                 |
| MI        | 205                |
| MS        | 104                |
| NM        | 78                 |
| WI        | 146                |
| AK        | 84                 |
| NJ        | 100                |
| SD        | 79                 |
| WV        | 77                 |
| FL        | 194                |
| IL        | 157                |
| IN        | 109                |
| TN        | 129                |
| AL        | 115                |
| NC        | 98                 |
| ND        | 77                 |
| NH        | 98                 |
| NV        | 121                |
| OK        | 83                 |
| OR        | 123                |
+-----------+--------------------+
```

10. Create `Network_Rate` view

```
drop view if exists Network_Rate;

CREATE VIEW Network_Rate AS
SELECT CK.BusinessYear AS NETWORK_YEAR, CK.StateCode AS
NETWORK_STATE, CK.NetworkName AS NETWORK_NAME, CR.*
FROM Cleaned_Network CK INNER JOIN Cleaned_Rate CR
ON CK.IssuerID = CR.IssuerID
```

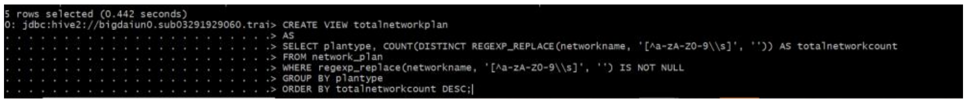
11. Create `PlantypeState` view

```
DROP VIEW IF EXISTS PlantypeState;

-- Create view to calculate plan count per state and plan type
CREATE VIEW PlantypeState
AS
SELECT statecode, plantype, COUNT(*) as plancount
FROM network_plan
```

```
GROUP BY statecode, plantype
ORDER BY statecode;
```

12. Create `totalnetworkplan` view

```
CREATE VIEW totalnetworkplan
AS
SELECT plantype, COUNT(DISTINCT REGEXP_REPLACE(networkname, '[^a-
zA-Z0-9\\s]', '')) AS totalnetworkcount
FROM network_plan
WHERE regexp_replace(networkname, '[^a-zA-Z0-9\\s]', '') IS NOT
NULL
GROUP BY plantype
ORDER BY totalnetworkcount DESC;
```



```
5 rows selected (0.442 seconds)
0: jdbc:hive2://bigdaiun0.sub03291929060.trai> CREATE VIEW totalnetworkplan
. . . . . . . . . . . . . . . . . . . . . . .> AS
. . . . . . . . . . . . . . . . . . . . . . .> SELECT plantype, COUNT(DISTINCT REGEXP_REPLACE(networkname, '[^a-zA-Z0-9\\s]', '')) AS totalnetworkcount
. . . . . . . . . . . . . . . . . . . . . . .> FROM network_plan
. . . . . . . . . . . . . . . . . . . . . . .> WHERE regexp_replace(networkname, '[^a-zA-Z0-9\\s]', '') IS NOT NULL
. . . . . . . . . . . . . . . . . . . . . . .> GROUP BY plantype
. . . . . . . . . . . . . . . . . . . . . . .> ORDER BY totalnetworkcount DESC;
```

13. Create top 5 benefits view

```
CREATE VIEW top5Benefits AS
WITH RankedBenefitCounts AS (
    SELECT
        BusinessYear,
        BenefitName,
        COUNT(*) AS BenefitCount,
        RANK() OVER (PARTITION BY BusinessYear ORDER BY COUNT(*)
DESC) AS BenefitRank
    FROM
        Cleaned_Benefits WHERE BusinessYear IS NOT NULL
    GROUP BY
        BusinessYear, BenefitName
)

SELECT
    BusinessYear,
    BenefitName,
    BenefitCount
FROM
    RankedBenefitCounts
WHERE
    BenefitRank <= 5;
```

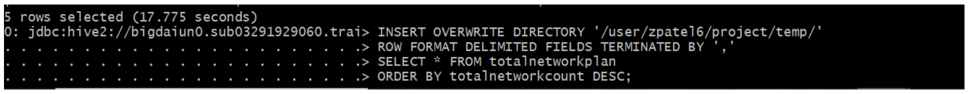
Step 5: Export Data

1. Create temporary directory

```
hdfs dfs -mkdir project/temp;
```

2. Export all data to a CSV file in the temporary directory

```
INSERT OVERWRITE DIRECTORY '/user/aporwal/project/temp/'
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
SELECT * FROM Benefit_Count
ORDER BY StateCode;


INSERT OVERWRITE DIRECTORY '/user/zpatel6/project/temp/'
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
SELECT * FROM totalnetworkplan
ORDER BY totalnetworkcount DESC;
```



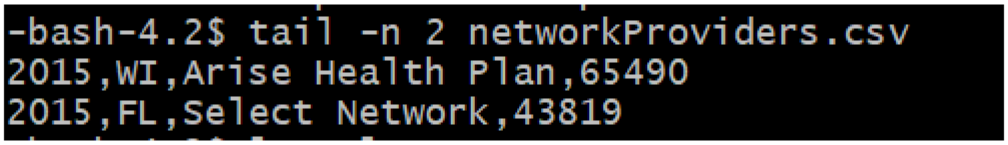
```
5 rows selected (17.775 seconds)
0: jdbc:hive2://bigdaiun0.sub03291929060.trai> INSERT OVERWRITE DIRECTORY '/user/zpatel6/project/temp/'
. . . . . . . . . . . . . . . . . . . . . . .> ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
. . . . . . . . . . . . . . . . . . . . . . .> SELECT * FROM totalnetworkplan
. . . . . . . . . . . . . . . . . . . . . . .> ORDER BY totalnetworkcount DESC;
```

```
INSERT OVERWRITE DIRECTORY '/user/clin22/project/temp/'
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
SELECT BUSINESSYEAR, AVG(INDIVIDUALRATE) as INDIVIDUAL,
AVG(COUPLE) AS COUPLE,
AVG(primarysubscriberandonedependent) AS DEPENDENT,
AVG(primarysubscriberandtwodependents) AS DEPENDENT2,
AVG(primarysubscriberandthreeormoredependents) AS DEPENDENT3,
AVG(coupleandonedependent) CDEPENDENT,
AVG(coupleandtwodependents) CDEPENDENT2,
AVG(coupleandthreeormoredependents) AS CDEPENDENT3
FROM Network_Rate
WHERE INDIVIDUALRATE IS NOT NULL AND COUPLE IS NOT NULL
GROUP BY BUSINESSYEAR
ORDER BY BUSINESSYEAR;

INSERT OVERWRITE DIRECTORY '/user/kbhanda3/proj5200/temp1/'
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
SELECT businessyear, statecode, networkname, count_of_values
FROM (
    SELECT businessyear, statecode, networkname,
COUNT(networkname) AS count_of_values,
           ROW_NUMBER() OVER (PARTITION BY businessyear ORDER BY
COUNT(REGEXP_REPLACE(networkname, '[^a-zA-Z0-9\\s]', '')) DESC)
AS row_num
    FROM tempNetPlanBenefit
    WHERE networkname != plantype
    GROUP BY businessyear, networkname, statecode
) AS counts_per_year
WHERE row_num IN (1,2,3);

INSERT OVERWRITE DIRECTORY '/user/zpatel6/project/temp/'
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
SELECT * FROM PlantypeState
```

```
ORDER BY statecode;

INSERT OVERWRITE DIRECTORY '/user/aporwal/project/temp/'
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
```

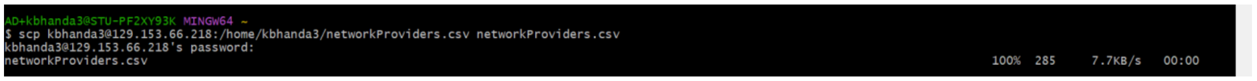
3. Download files to local machine

```
hdfs dfs -get project/temp/000000_0 networkcountplantype.csv

scp
zpatel6@129.153.66.218:/home/zpatel6/networkcountplantype.csv .

hdfs dfs -get project/temp/000000_0 plantypestate.csv

scp zpatel6@129.153.66.218:/home/zpatel6/plantypestate.csv .

hdfs dfs -ls proj5200/temp1

hdfs dfs -getmerge proj5200/temp1 networkproviders.csv

tail -n 2 networkProviders.csv
```



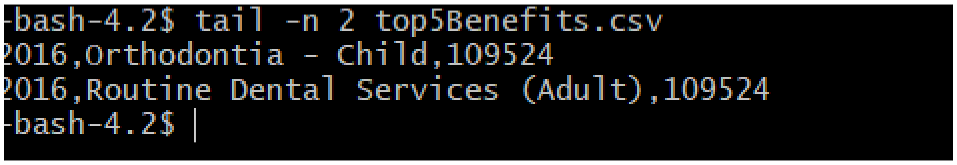
```
-bash-4.2$ tail -n 2 networkProviders.csv
2015,WI,Arise Health Plan,65490
2015,FL,Select Network,43819
```

```
scp kbhanda3@129.153.66.218:/home/kbhanda3/networkProviders.csv
networkProviders.csv
```



```
AD+kbhanda3@STU-PF2XY93K MINGW64 ~
$ scp kbhanda3@129.153.66.218:/home/kbhanda3/networkProviders.csv networkProviders.csv
kbhanda3@129.153.66.218's password:
networkProviders.csv                                          100%  285     7.7KB/s   00:00
```

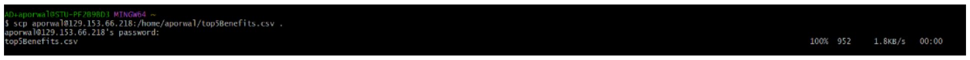
```
hdfs dfs -get project/temp/000000_0 top5Benefits.csv

tail -n 2 top5Benefits.csv
```



```
-bash-4.2$ tail -n 2 top5Benefits.csv
2016,Orthodontia - Child,109524
2016,Routine Dental Services (Adult),109524
-bash-4.2$
```

\

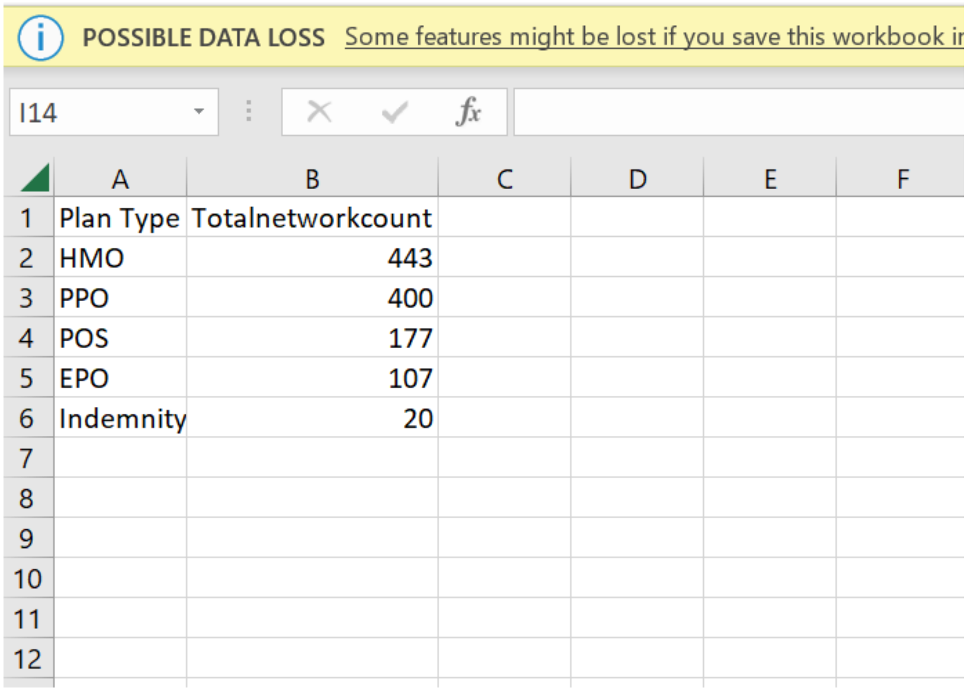
```
scp aporwal@129.153.66.218:/home/aporwal/top5Benefits.csv .
```



```
AD+aporwal@STU-PF2B9BD3 MINGW64 ~
$ scp aporwal@129.153.66.218:/home/aporwal/top5Benefits.csv .
aporwal@129.153.66.218's password:
top5Benefits.csv                                          100%  952     1.8KB/s   00:00
```

Step 6: Create Visualization

1. Create network count plan type visualization
2. Open 'networkcountplantype.csv' file in excel
3. Insert a row at top and give title as 'Plan type' and 'Totalnetworkcount'

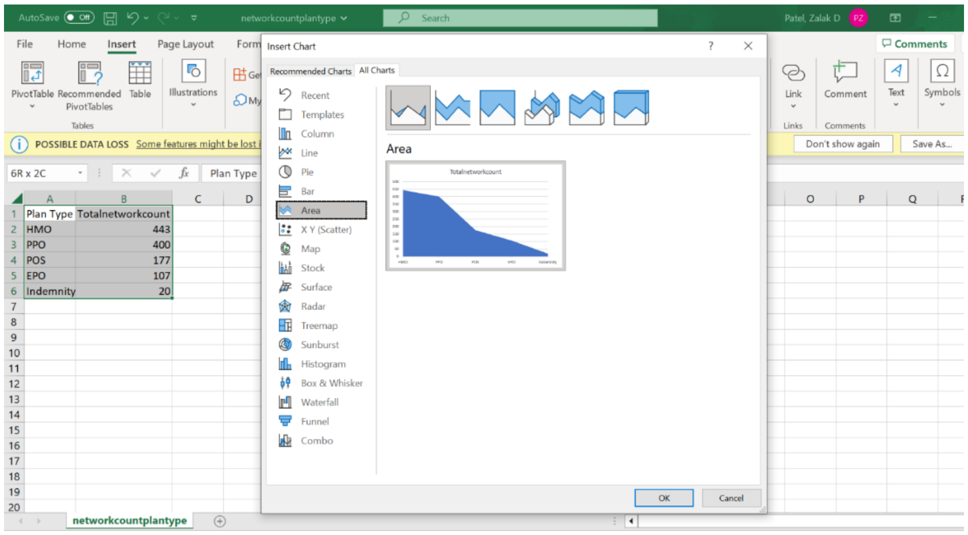


POSSIBLE DATA LOSS Some features might be lost if you save this workbook i

I14

	A	B
1	Plan Type	Totalnetworkcount
2	HMO	443
3	PPO	400
4	POS	177
5	EPO	107
6	Indemnity	20

4. Select data, tap on insert, and select ‘Area’ graph as shown below Screenshot

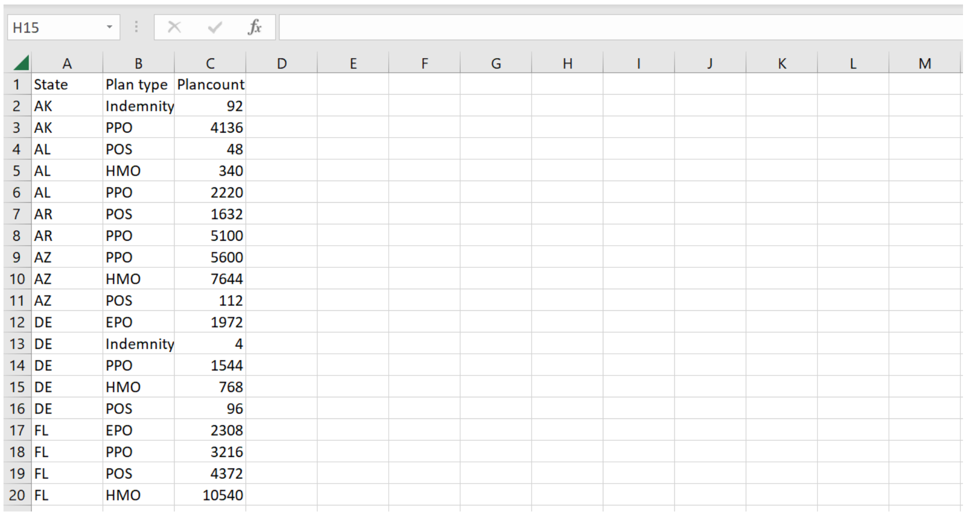


POSSIBLE DATA LOSS Some features might be lost i

	A	B
1	Plan Type	Totalnetworkcount
2	HMO	443
3	PPO	400
4	POS	177
5	EPO	107
6	Indemnity	20

5. After selecting the chart, you can select a variety of chart styles from the char design as shown in the below Screenshot
6. Open ‘plantypestate.csv’ file in excel and save it as excel format as a 3D map can only show in excel
7. Now open newly saved ‘plantypestate.xlsx’ file in excel

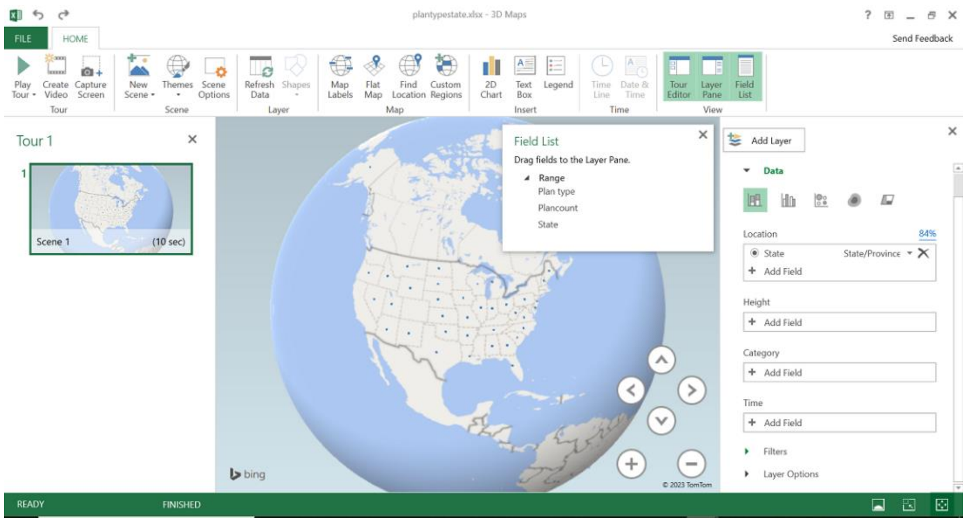
8. Insert a row at the top and add header ‘State’, ‘Plan Type’, and ‘Plancount’



H15

	A	B	C
1	State	Plan type	Plancount
2	AK	Indemnity	92
3	AK	PPO	4136
4	AL	POS	48
5	AL	HMO	340
6	AL	PPO	2220
7	AR	POS	1632
8	AR	PPO	5100
9	AZ	PPO	5600
10	AZ	HMO	7644
11	AZ	POS	112
12	DE	EPO	1972
13	DE	Indemnity	4
14	DE	PPO	1544
15	DE	HMO	768
16	DE	POS	96
17	FL	EPO	2308
18	FL	PPO	3216
19	FL	POS	4372
20	FL	HMO	10540

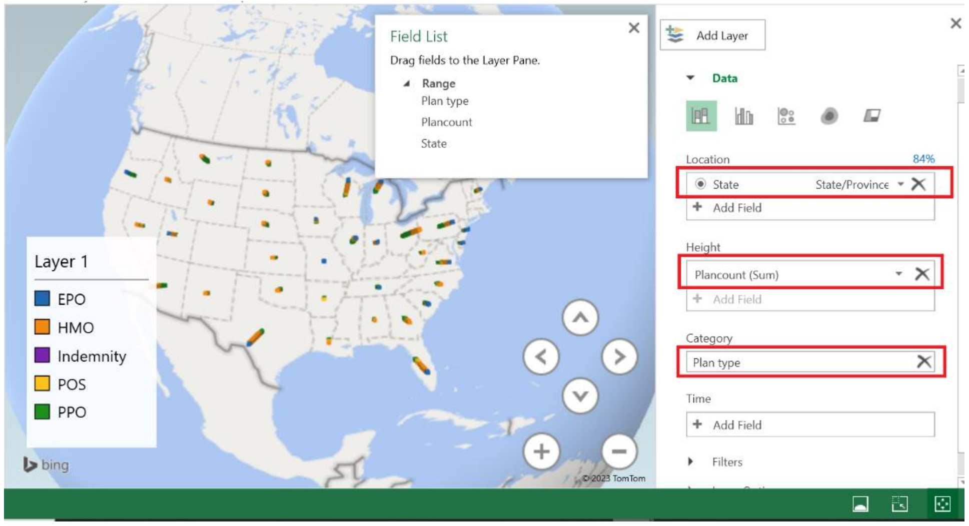
9. Now, select data > tap on insert > Open 3D map



Field List
Drag fields to the Layer Pane.
Range
Plan type
Plancount
State

Add Layer
Data
Location 84%
State State/Province
Add Field
Height
Add Field
Category
Add Field
Time
Add Field
Filters
Layer Options

10. Select the properties and values in the layer as follows.

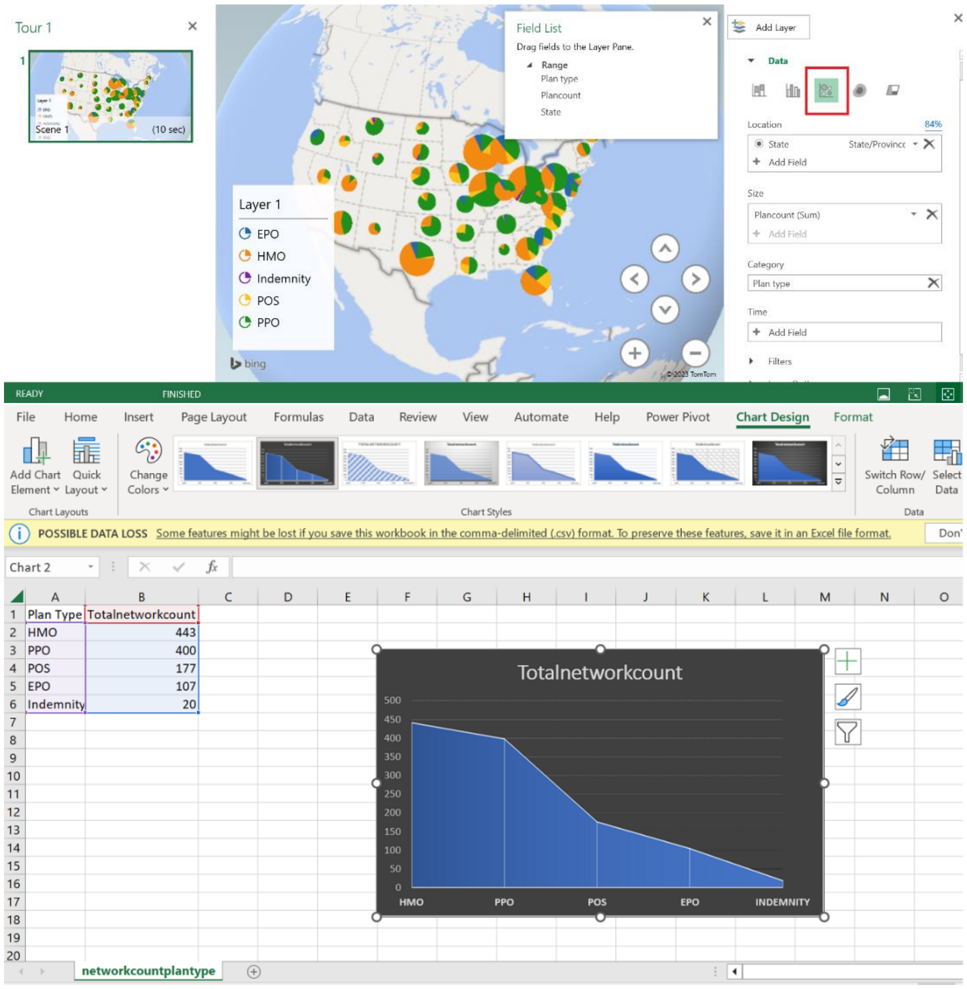


Field List
Drag fields to the Layer Pane.
Range
Plan type
Plancount
State

Layer 1
🟦 EPO
🟧 HMO
🟪 Indemnity
🟨 POS
🟩 PPO

Add Layer
Data
Location 84%
State State/Province
Add Field
Height
Plancount (Sum)
Add Field
Category
Plan type
Time
Add Field
Filters

11. Finally, change the visualization to a bubble. Then you can drag the earth and rotate it to observe different types across state of USA

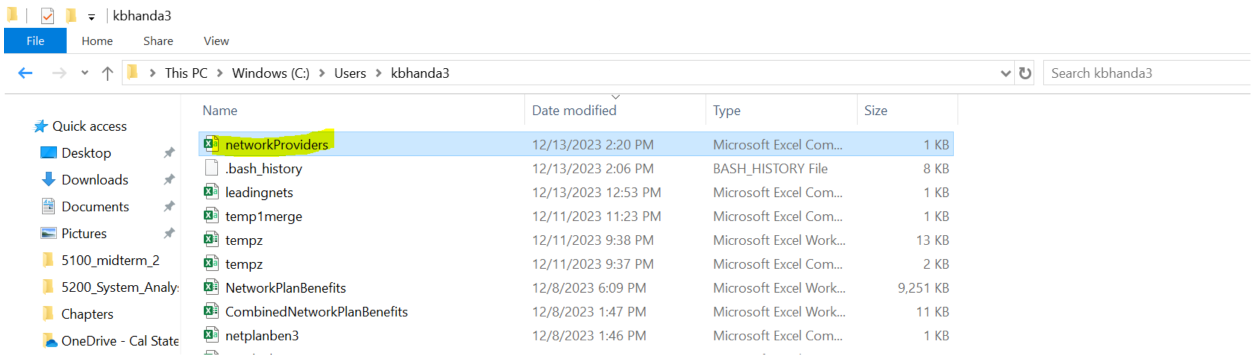


POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format.

Plan Type	Totalnetworkcount
HMO	443
PPO	400
POS	177
EPO	107
Indemnity	20

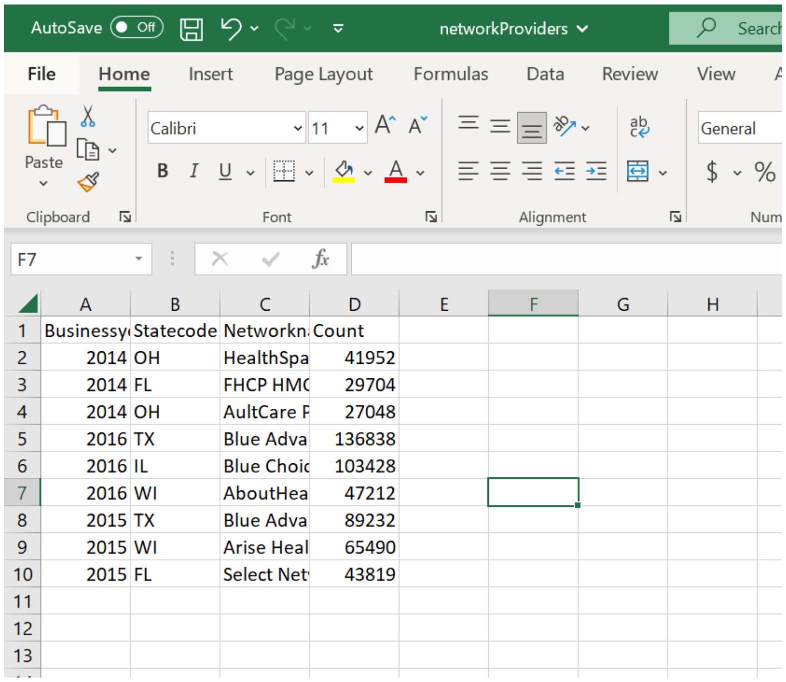
12. Create network providers visualization.

13. Open .csv file and add column names. Save the file in excel format. You can use the excel to perform visualization as well, but for this analysis, we will use Tableau.



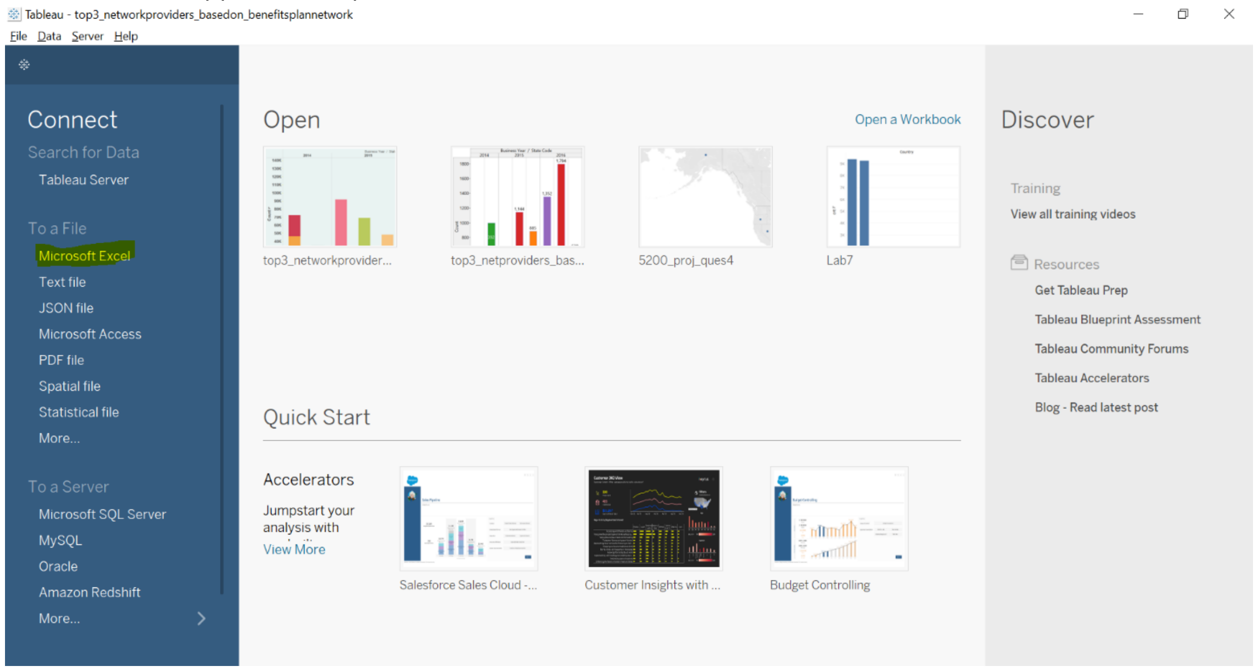
Name	Date modified	Type	Size
networkProviders	12/13/2023 2:20 PM	Microsoft Excel Com...	1 KB
.bash_history	12/13/2023 2:06 PM	BASH_HISTORY File	8 KB
leadingnets	12/13/2023 12:53 PM	Microsoft Excel Com...	1 KB
temp1merge	12/11/2023 11:23 PM	Microsoft Excel Com...	1 KB
tempz	12/11/2023 9:38 PM	Microsoft Excel Work...	13 KB
tempz	12/11/2023 9:37 PM	Microsoft Excel Com...	2 KB
NetworkPlanBenefits	12/8/2023 6:09 PM	Microsoft Excel Work...	9,251 KB
CombinedNetworkPlanBenefits	12/8/2023 1:47 PM	Microsoft Excel Work...	11 KB
netplanben3	12/8/2023 1:46 PM	Microsoft Excel Com...	1 KB

14. Add column names and save as ".xls":

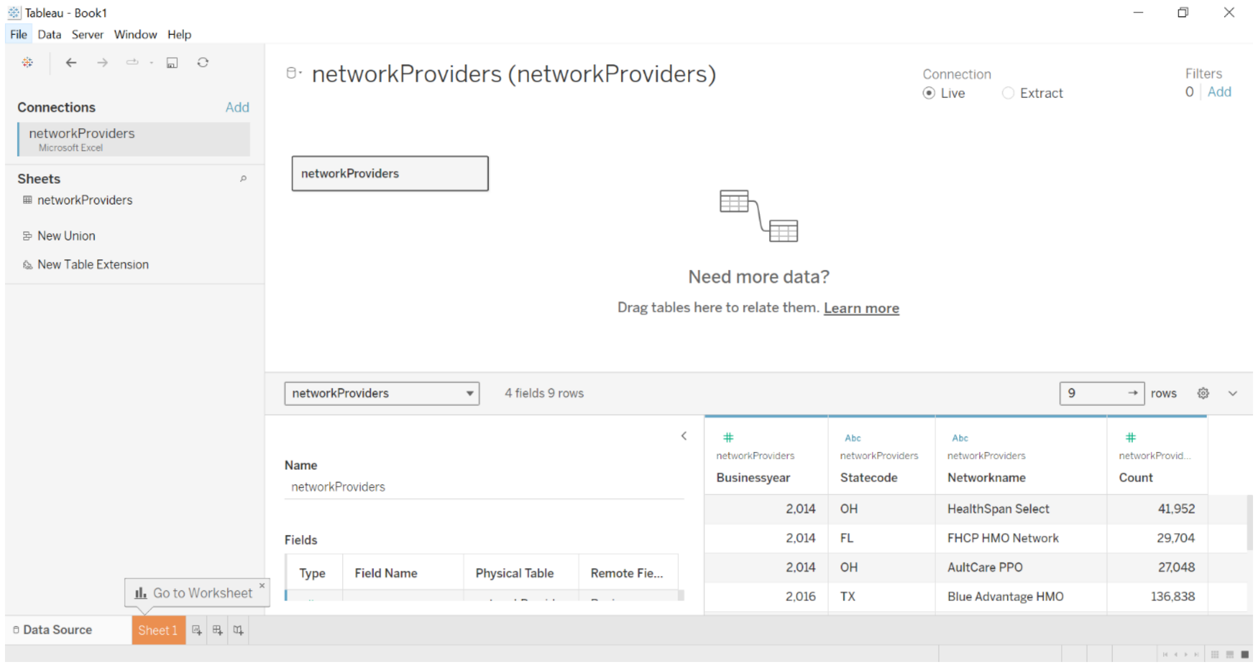


	A	B	C	D	E	F	G	H
1	Businessy	Statecode	Networkn	Count				
2	2014	OH	HealthSpa	41952				
3	2014	FL	FHCP HMC	29704				
4	2014	OH	AultCare F	27048				
5	2016	TX	Blue Adva	136838				
6	2016	IL	Blue Choic	103428				
7	2016	WI	AboutHea	47212				
8	2015	TX	Blue Adva	89232				
9	2015	WI	Arise Heal	65490				
10	2015	FL	Select Net	43819				
11								
12								
13								

15. On the tableau application, open the excel file.



16. Click on “Sheet 1”:



Connection: (●) Live ( ) Extract

networkProviders — 4 fields 9 rows — 9 rows

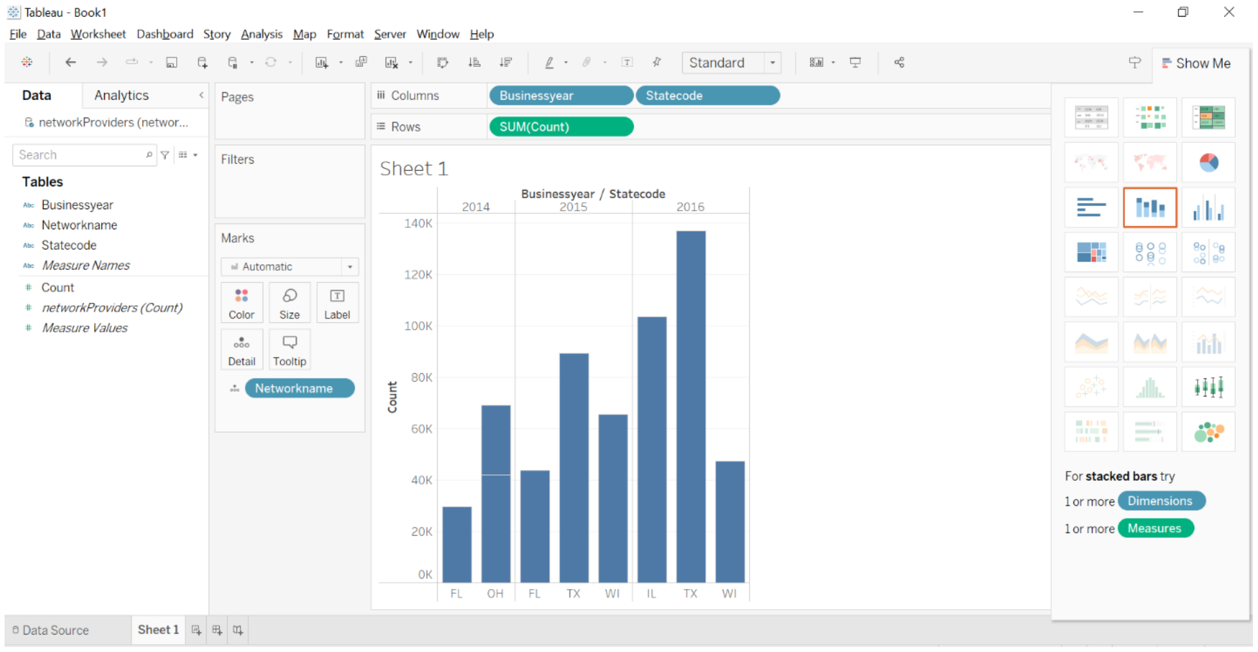
Name: networkProviders

Fields

Type	Field Name	Physical Table	Remote Fie...

networkProviders Businessyear	networkProviders Statecode	networkProviders Networkname	networkProvid... Count
2,014	OH	HealthSpan Select	41,952
2,014	FL	FHCP HMO Network	29,704
2,014	OH	AultCare PPO	27,048
2,016	TX	Blue Advantage HMO	136,838

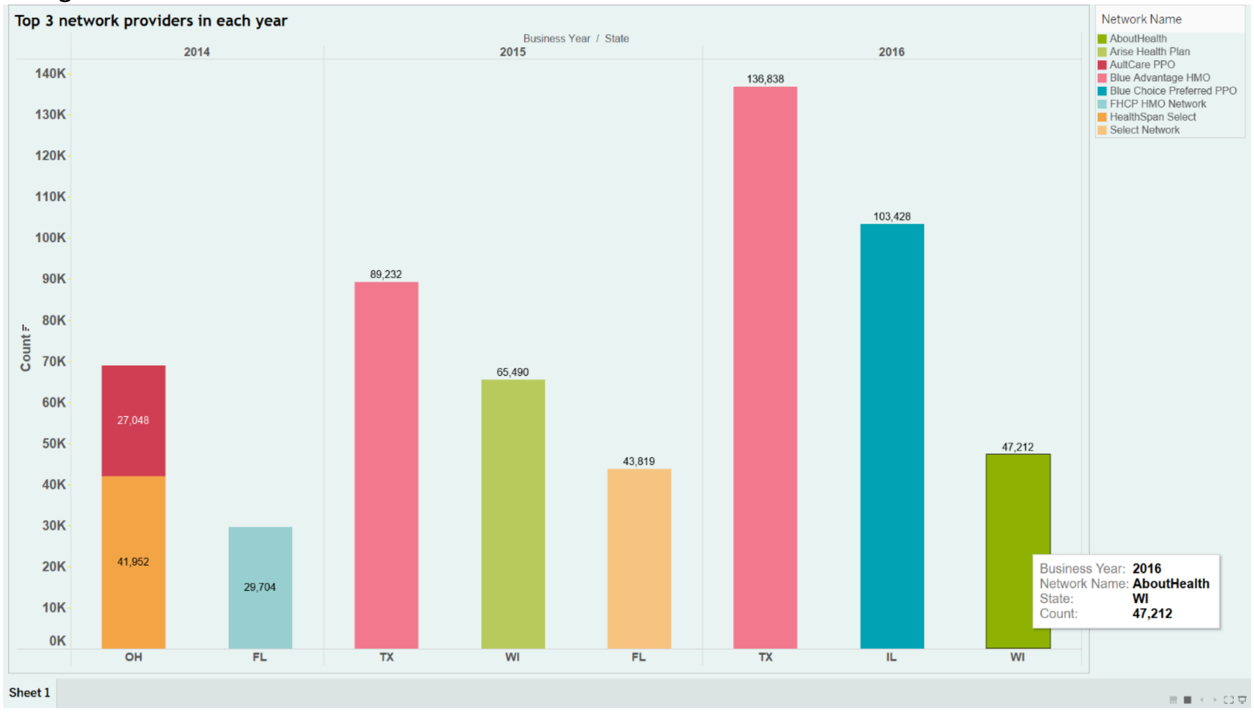
17. Right click on “business year” and convert it to dimension.



Columns: Businessyear, Statecode
Rows: SUM(Count)
Marks: Detail — Networkname

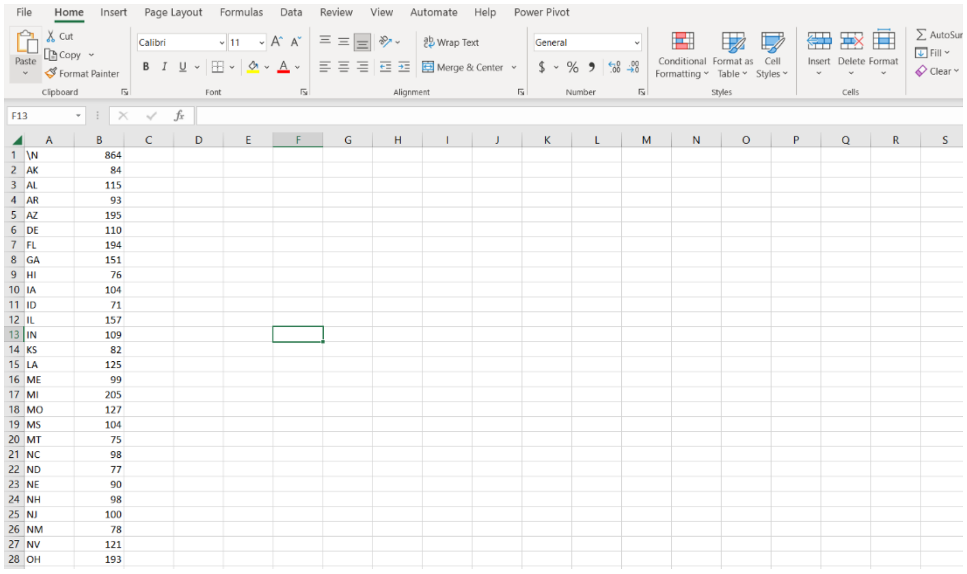
18. Select ‘Stacked bars’ as an option from “Show Me” section, then drag Business year and State from “Tables” as Columns and drag “SUM(Count)” as Rows. Change the color,

background and title:



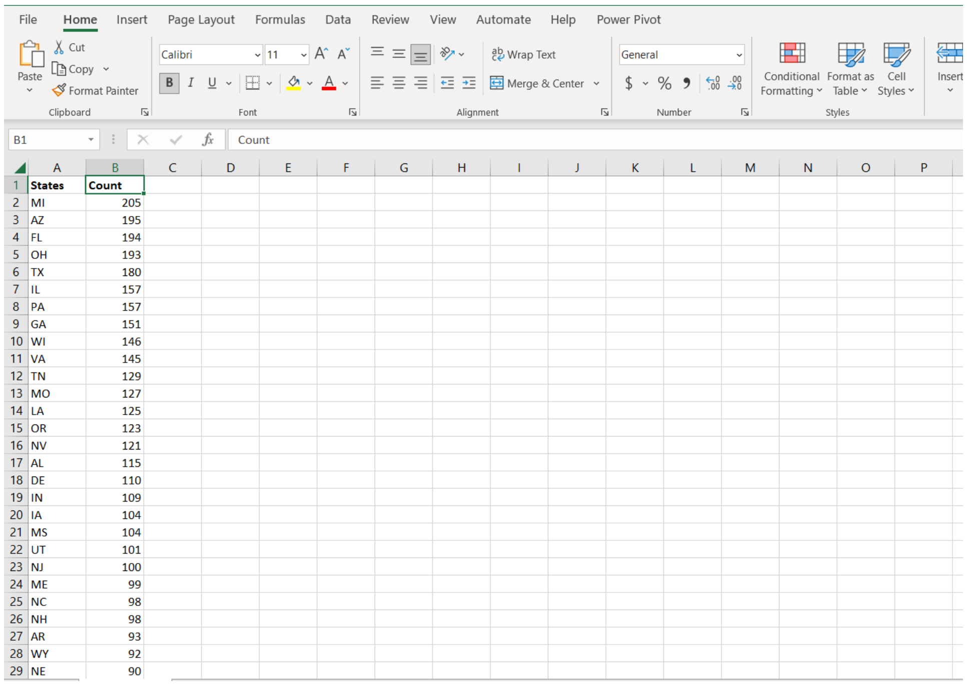
19. Create visualization for states with the most benefits

20. Open the downloaded data in excel



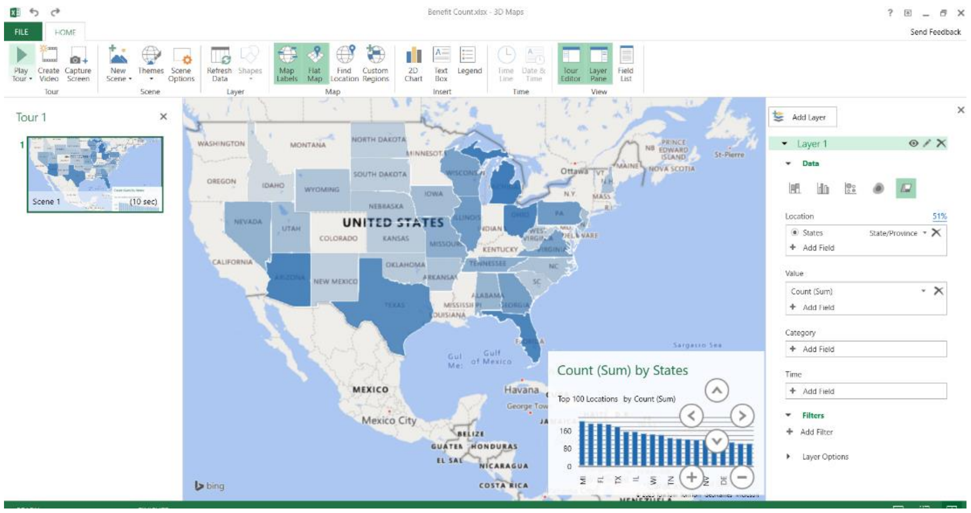
	A	B
1	\N	864
2	AK	84
3	AL	115
4	AR	93
5	AZ	195
6	DE	110
7	FL	194
8	GA	151
9	HI	76
10	IA	104
11	ID	71
12	IL	157
13	IN	109
14	KS	82
15	LA	125
16	ME	99
17	MI	205
18	MO	127
19	MS	104
20	MT	75
21	NC	98
22	ND	77
23	NE	90
24	NH	98
25	NJ	100
26	NM	78
27	NV	121
28	OH	193

21. Rearranged them based on the number of counts:

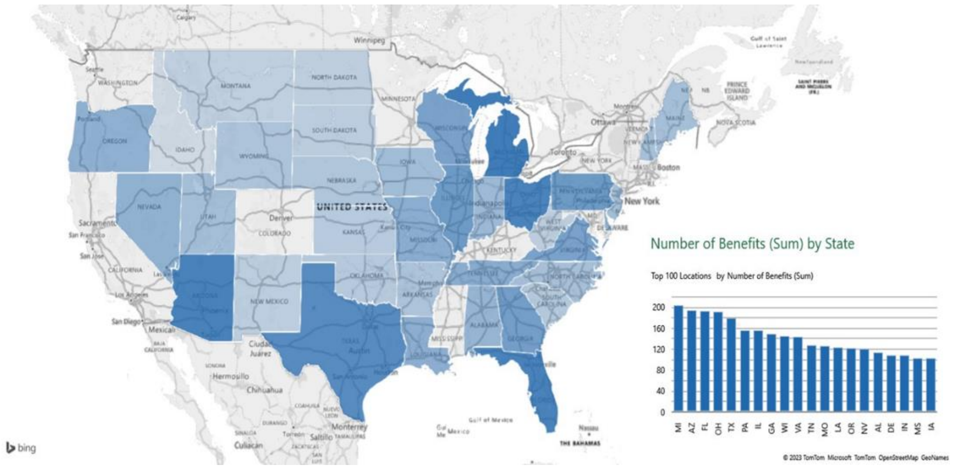


States	Count
MI	205
AZ	195
FL	194
OH	193
TX	180
IL	157
PA	157
GA	151
WI	146
VA	145
TN	129
MO	127
LA	125
OR	123
NV	121
AL	115
DE	110
IN	109
IA	104
MS	104
UT	101
NJ	100
ME	99
NC	98
NH	98
AR	93
WY	92
NE	90

22. Saved it in excel format and created a 3D Map
    For the 3D Map, we selected multiple options **Map Labels, Flat Map.**
    Along with spatial analysis, we also inserted a 2D Chart to display the count for different states.

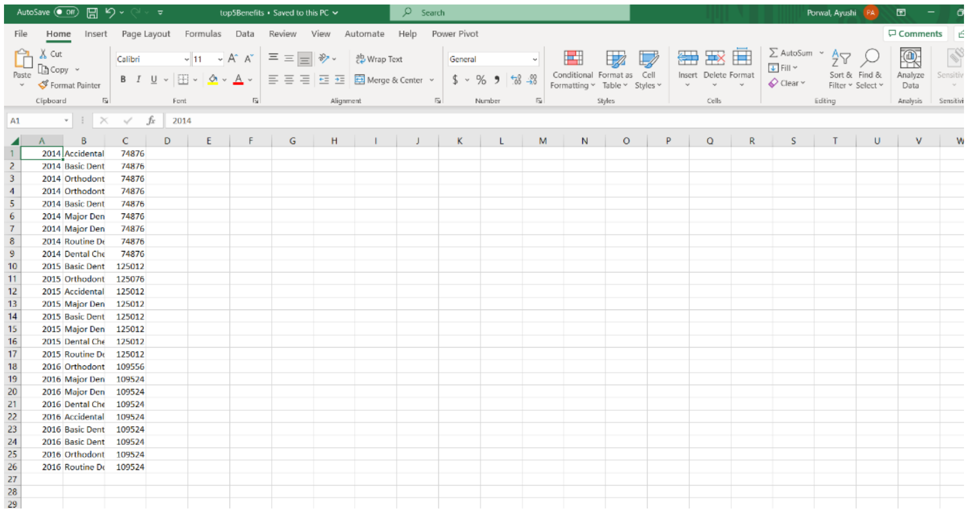


23. For the final output, changed the aesthetic and the output looks like this:


Number of Benefits (Sum) by State
Top 100 Locations by Number of Benefits (Sum)
200
160
120
80
40
0
MI
AZ
FL
OH
TX
PA
IL
GA
WI
VA
TN
MO
LA
OR
NV
AL
DE
IN
MS
IA
UNITED STATES
bing
© 2023 TomTom Microsoft TomTom OpenStreetMap GeoNames

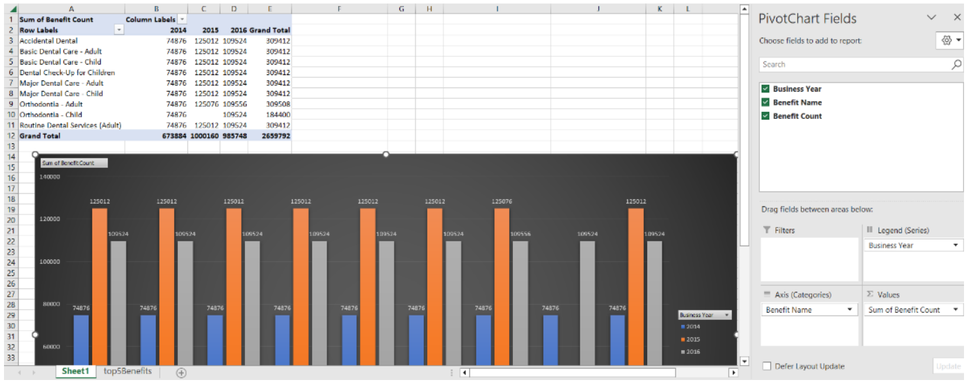
24. Create top 5 benefits visualization

25. Open the downloaded data in excel



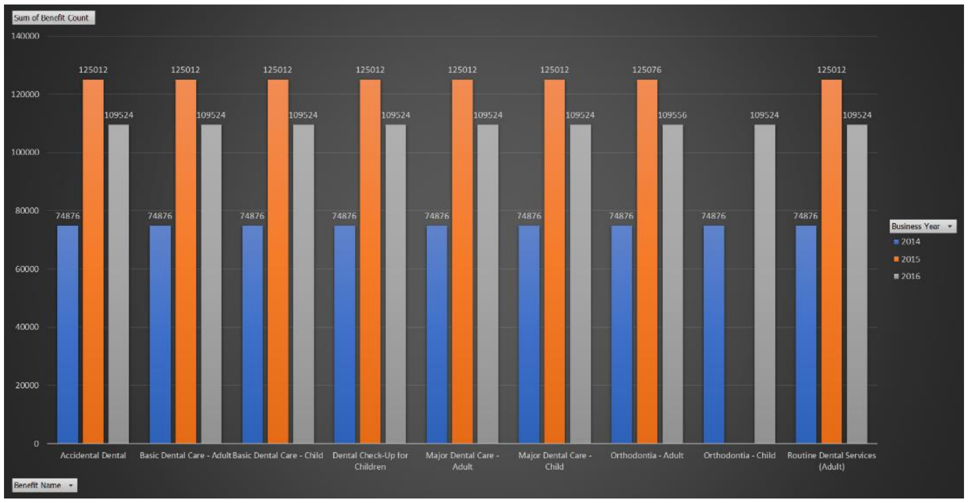
	A	B	C
1	2014	Accidental	74876
2	2014	Basic Dent	74876
3	2014	Orthodont	74876
4	2014	Orthodont	74876
5	2014	Basic Dent	74876
6	2014	Major Den	74876
7	2014	Major Den	74876
8	2014	Routine De	74876
9	2014	Dental Che	74876
10	2015	Basic Dent	125012
11	2015	Orthodont	125076
12	2015	Accidental	125012
13	2015	Major Den	125012
14	2015	Basic Dent	125012
15	2015	Major Den	125012
16	2015	Dental Che	125012
17	2015	Routine De	125012
18	2016	Orthodont	109556
19	2016	Major Den	109524
20	2016	Major Den	109524
21	2016	Dental Che	109524
22	2016	Accidental	109524
23	2016	Basic Dent	109524
24	2016	Basic Dent	109524
25	2016	Orthodont	109524
26	2016	Routine De	109524

26. Create a pivot chart with legend as **Business Year,** Axis as **Benefit Name,** Values as **Sum of Benefit Count**



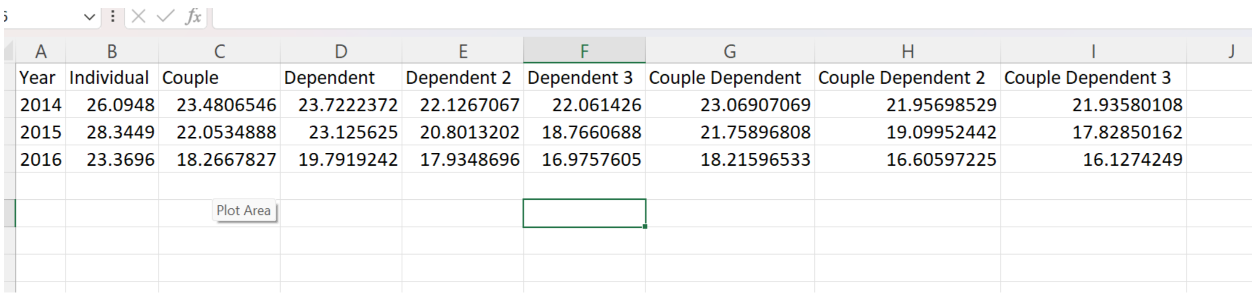
Sum of Benefit Count	Column Labels			
Row Labels	2014	2015	2016	Grand Total
Accidental Dental	74876	125012	109524	309412
Basic Dental Care - Adult	74876	125012	109524	309412
Basic Dental Care - Child	74876	125012	109524	309412
Dental Check-Up for Children	74876	125012	109524	309412
Major Dental Care - Adult	74876	125012	109524	309412
Major Dental Care - Child	74876	125012	109524	309412
Orthodontia - Adult	74876	125076	109556	309508
Orthodontia - Child	74876		109524	184400
Routine Dental Services (Adult)	74876	125012	109524	309412
Grand Total	673884	1000160	985748	2659792

27. The graph:



28. Create the Average Cost per Person visualization.

29. Open downloaded CSV file in excel.



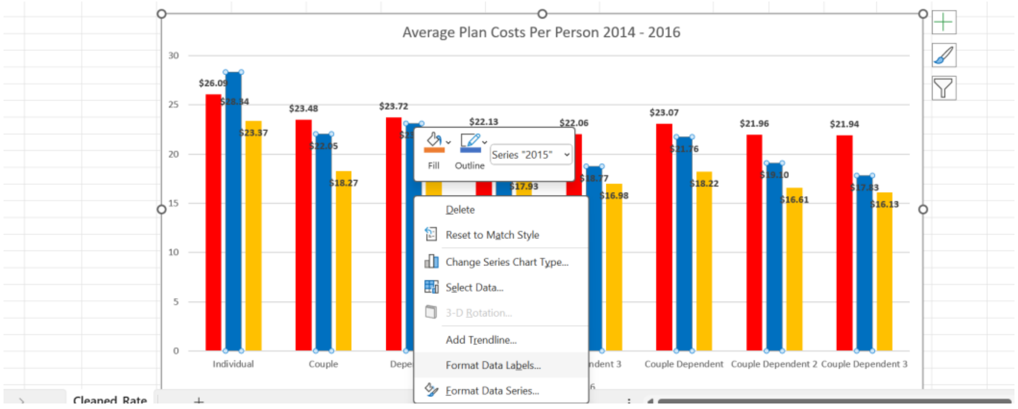
A	B	C	D	E	F	G	H	I	J
Year	Individual	Couple	Dependent	Dependent 2	Dependent 3	Couple Dependent	Couple Dependent 2	Couple Dependent 3	
2014	26.0948	23.4806546	23.7222372	22.1267067	22.061426	23.06907069	21.95698529	21.93580108	
2015	28.3449	22.0534888	23.125625	20.8013202	18.7660688	21.75896808	19.09952442	17.82850162	
2016	23.3696	18.2667827	19.7919242	17.9348696	16.9757605	18.21596533	16.60597225	16.1274249	

Plot Area

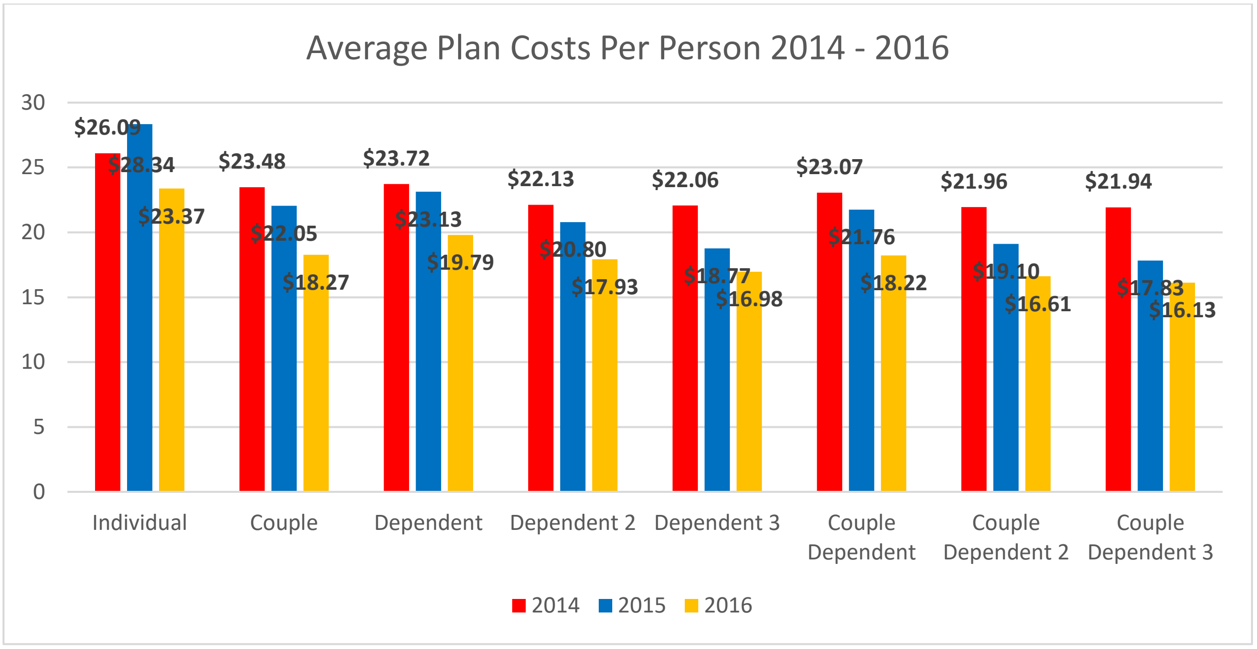
30. Change the column names to the figure above.

31. Highlight the data and select bar chart.

32. Change the coloring, and create labels by right clicking and select Format Data Labels



33. Resulting Chart



References

1. URL of Data Source, http://www.calstatela.edu
2. URL of your GitHub, https://github.com/ayushiporwal13/HealthInsuranceAnalysis
3. URL of References, https://www.kaggle.com/code/shelars1985/exploring-health-insurance-marketplace