Marketing Dashboard using a Prototype

Summary	Building marketing analytical dashboard augmenting TPC-DS
URL	https://www.kaggle.com/jackdaoud/marketing-data
Tools	XSV, Trifacta, Snowflake, Salesforce Einstein Analytics

---

SQL Queries on the TPC-DS Dataset

Task 1 - Start with the TPC-DS Dataset from Snowflake. Design a dashboard that will leverage queries from Snowflake to Einstein Analytics to build the dashboard.

The TPC Benchmark™DS (TPC-DS) is a decision support benchmark that models several generally applicable aspects of a decision support system, including queries and data maintenance. The benchmark provides a representative evaluation of the System Under

Test's (SUT) performance as a general purpose decision support system.

This benchmark illustrates decision support systems that:
• Examine large volumes of data;
• Give answers to real-world business questions;
• Execute queries of various operational requirements and complexities (e.g., ad-hoc, reporting, iterative OLAP, data mining);
• Are characterized by high CPU and IO load;
• Are periodically synchronized with source OLTP databases through database maintenance functions.
• Run on "Big Data" solutions, such as RDBMS as well as Hadoop/Spark based systems.

To check customers who spend more via catalog than in stores. Identified prefered customers and their country of origin.

Query-

```
--Find customers who spend more money via catalog than in stores. Identify preferred customers and their country of origin.

with year_total as (
 select c_customer_id customer_id
       ,c_first_name customer_first_name
       ,c_last_name customer_last_name
       ,c_preferred_cust_flag customer_preferred_cust_flag
       ,c_birth_country customer_birth_country
       ,c_login customer_login
       ,c_email_address customer_email_address
       ,d_year dyear
       ,sum(((ss_ext_list_price-ss_ext_wholesale_cost-ss_ext_discount_amt)+ss_ext_sales_price)/2) year_total
       ,'s' sale_type
 from customer
     ,store_sales
     ,date_dim
 where c_customer_sk = ss_customer_sk
   and ss_sold_date_sk = d_date_sk
 group by c_customer_id
         ,c_first_name
         ,c_last_name
```

Output-

Row	CUSTOMER_ID	CUSTOMER_FIRST_NAME	CUSTOMER_LAST_NAME	CUSTOMER_LOGIN
1	AAAAAAAAAAAAAABA	Scott	Gordon	NULL
2	AAAAAAAAAAAAAEBA	Donna	Barnes	NULL
3	AAAAAAAAAAAAAGCA	Antonio	Carney	NULL
4	AAAAAAAAAAAAAJCA	Marian	Sweet	NULL
5	AAAAAAAAAAAAALDA	Joseph	Hart	NULL
6	AAAAAAAAAAAAANDA	Alexandria	Lancaster	NULL
7	AAAAAAAAAAAABBDA	Stanley	Oliver	NULL
8	AAAAAAAAAAAABCDA	Matthew	Simon	NULL
9	AAAAAAAAAAAABDBA	Magdalena	Fitts	NULL

To check the total web sales for customers in specific zip code, cities, countries or states, or specific items for a given year and quarter.

Query-

```
--Report the total web sales for customers in specific zip codes, cities, counties or states, or specific items for a
--given year and quarter. .

select  ca_zip, ca_city, sum(ws_sales_price)
 from web_sales, customer, customer_address, date_dim, item
 where ws_bill_customer_sk = c_customer_sk
  and c_current_addr_sk = ca_address_sk
  and ws_item_sk = i_item_sk
  and ( substr(ca_zip,1,5) in ('85669', '86197','88274','83405','86475', '85392', '85460', '80348', '81792')
        or
        i_item_id in (select i_item_id
                           from item
                           where i_item_sk in (2, 3, 5, 7, 11, 13, 17, 19, 23, 29)
                           )
      )
  and ws_sold_date_sk = d_date_sk
  and d_qoy = 1 and d_year = 1998
 group by ca_zip, ca_city
 order by ca_zip, ca_city
 limit 100;
```

Output-

Row	CA_ZIP	CA_CITY	SUM(WS_SALES_PRICE)
1	00659	Centerville	197.31
2	00669	Edgewood	65.99
3	00741	California	83.41
4	00750	Bunker Hill	199.78
5	00762	Stringtown	103.76
6	00769	Oakwood	268.43
7	00791	Belmont	12.09
8	00999	Marion	126.34
9	01011	Cedar Grove	93.65

To check for each store, the number of items in a specified month that were returned after some number of days from the day of purchase.

Query-

```sql
--For each store count the number of items in a specified month that were returned after 30, 60, 90, 120 and more
--than 120 days from the day of purchase.

select
   s_store_name
  ,s_company_id
  ,s_street_number
  ,s_street_name
  ,s_street_type
  ,s_suite_number
  ,s_city
  ,s_county
  ,s_state
  ,s_zip
  ,sum(case when (sr_returned_date_sk - ss_sold_date_sk <= 30 ) then 1 else 0 end)  as "30 days"
  ,sum(case when (sr_returned_date_sk - ss_sold_date_sk > 30) and
                 (sr_returned_date_sk - ss_sold_date_sk <= 60) then 1 else 0 end )  as "31-60 days"
  ,sum(case when (sr_returned_date_sk - ss_sold_date_sk > 60) and
                 (sr_returned_date_sk - ss_sold_date_sk <= 90) then 1 else 0 end)  as "61-90 days"
  ,sum(case when (sr_returned_date_sk - ss_sold_date_sk > 90) and
                 (sr_returned_date_sk - ss_sold_date_sk <= 120) then 1 else 0 end)  as "91-120 days"
  ,sum(case when (sr_returned_date_sk - ss_sold_date_sk  > 120) then 1 else 0 end)  as ">120 days"
from
```

Output-

Row	S_STORE_NAME	S_COMPANY_ID	S_STREET_NUM	S_STREET_NAM	S_STREET_TYPE	S_SUITE_NUMB	S_CITY	S_COUNTY	S_STATE
1	able	1	113	South	Parkway	Suite 200	Pine Hill	Contra Cost...	CA
2	able	1	115	Hickory	Circle	Suite N	Clifton	Mobile County	AL
3	able	1	128	15th	RD	Suite T	Winchester	Nuckolls Co...	NE
4	able	1	128	Church East	ST	Suite U	Oak Hill	Contra Cost...	CA
5	able	1	140	6th	Avenue	Suite 120	Buffalo	Quay County	NM
6	able	1	148	River 2nd	Road	Suite 10	Pine Hill	Lea County	NM
7	able	1	17	Main	Ct.	Suite K	Woodland	Baltimore Co...	MD
8	able	1	170	Mill	Circle	Suite J	Cross Roads	Klamath Cou...	OR

To check for each store, in a given period, the list of items with the revenue less than a certain percent of the average revenue for all the items in that store.

Query-

```sql
--In a given period, for each store, report the list of items with revenue less than 10% the average revenue for all
--the items in that store.

select
  s_store_name,
  i_item_desc,
  sc.revenue,
  i_current_price,
  i_wholesale_cost,
  i_brand
 from store, item,
     (select ss_store_sk, avg(revenue) as ave
  from
      (select  ss_store_sk, ss_item_sk,
         sum(ss_sales_price) as revenue
    from store_sales, date_dim
    where ss_sold_date_sk = d_date_sk and d_month_seq between 1200 and 1200+11
    group by ss_store_sk, ss_item_sk) sa
  group by ss_store_sk) sb,
     (select  ss_store_sk, ss_item_sk, sum(ss_sales_price) as revenue
  from store_sales, date_dim
  where ss_sold_date_sk = d_date_sk and d_month_seq between 1200 and 1200+11
  group by ss_store_sk, ss_item_sk) sc
```

Output-

Row	S_STORE_NAME	I_ITEM_DESC	REVENUE	I_CURRENT_PRICE	I_WHOLESALE_COST	I_BRAND
1	able	A	93.80	6.73	4.17	namelessbrand #5
2	able	A	11.52	6.54	2.41	exportimaxi #5
3	able	A	1.51	6.73	4.17	namelessbrand #5
4	able	A	27.93	4.87	1.70	amalgmaxi #3
5	able	A	8.66	4.87	1.70	amalgmaxi #3
6	able	A	25.14	1.84	1.50	importoscholar #1
7	able	A	9.89	4.37	3.84	amalgbrand #3
8	able	A	41.07	6.54	2.41	exportimaxi #5
9	able	A	18.97	1.84	1.50	importoscholar #1

To check web and catalog sales and profits by warehouse.

Query-

```
--Compute web and catalog sales and profits by warehouse. Report results by month for a given year during a
--given 8-hour period.

select
         w_warehouse_name
  ,w_warehouse_sq_ft
  ,w_city
  ,w_county
  ,w_state
  ,w_country
        ,ship_carriers
        ,year
  ,sum(jan_sales) as jan_sales
  ,sum(feb_sales) as feb_sales
  ,sum(mar_sales) as mar_sales
  ,sum(apr_sales) as apr_sales
  ,sum(may_sales) as may_sales
  ,sum(jun_sales) as jun_sales
  ,sum(jul_sales) as jul_sales
  ,sum(aug_sales) as aug_sales
  ,sum(sep_sales) as sep_sales
  ,sum(oct_sales) as oct_sales
  ,sum(nov_sales) as nov_sales
```

Output-

Row	W_WAREHOUSE	W_WAREHOUSE	W_CITY	W_COUNTY	W_STATE	W_COUNTRY	SHIP_CARRIERS	YEAR	JAN_SALES
1	Agricultural ...	159446	Bethel	Bronx County	NY	United States	GREAT EAST...	1998	21514541701...
2	Bad cards m...	621234	Oakland	Gage County	NE	United States	GREAT EAST...	1998	2155646785...
3	Conventiona...	977787	Shiloh	Franklin Parish	LA	United States	GREAT EAST...	1998	2160838726...
4	Doors canno	294242	Cedar Grove	Raleigh Cou...	WV	United States	GREAT EAST...	1998	2159033207...
5	Empty, midd...	198212	Cedar Grove	Daviess Cou...	MO	United States	GREAT EAST...	1998	2171842228...
6	Friendly, suit...	863277	Lakeview	Ziebach Cou...	SD	United States	GREAT EAST...	1998	21711991226...
7	Government...	621938	Liberty	Mesa County	CO	United States	GREAT EAST...	1998	2169564671...
8	Important d...	185634	Pleasant Hill	Jackson Co...	NC	United States	GREAT EAST...	1998	21577271212...

To check per customer extended sales price, extended list price and extended tax for a kind of shoppers buying from certain stores during a period of time.

Query-

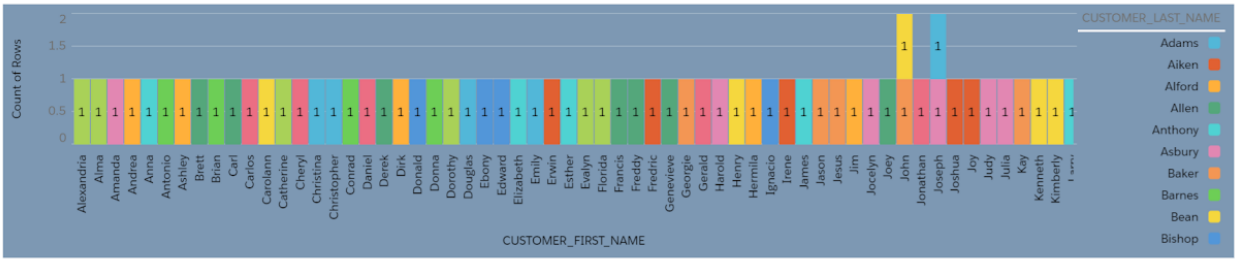
```
--Compute the per customer extended sales price, extended list price and extended tax for "out of town" shoppers
--buying from stores located in two cities in the first two days of each month of three consecutive years. Only
--consider customers with specific dependent and vehicle counts.

select  c_last_name
       ,c_first_name
       ,ca_city
       ,bought_c"SNOWFLAKE_SAMPLE_DATA"."TPCDS_SF100TCL"."PROMOTION"ity
       ,ss_ticket_number
       ,extended_price
       ,extended_tax
       ,list_price
 from (select ss_ticket_number
             ,ss_customer_sk
             ,ca_city bought_city
             ,sum(ss_ext_sales_price) extended_price
             ,sum(ss_ext_list_price) list_price
             ,sum(ss_ext_tax) extended_tax
       from store_sales
           ,date_dim
           ,store
           ,household_demographics
           ,customer_address
```

Output-

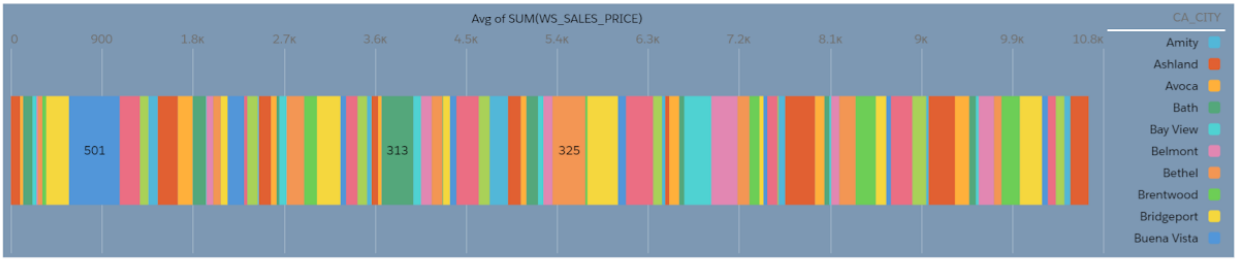
Row	C_LAST_NAME	C_FIRST_NAME	CA_CITY	BOUGHT_CITY	SS_TICKET_NUMB	EXTENDED_PRICE	EXTENDED_TAX	LIST_PRICE
1	Aaron	Patrick	Ashland	Hillcrest	197696993	17699.71	626.92	34864.64
2	Aaron	Rachel	Sunnyside	Spring Hill	247710297	24160.48	1738.73	41751.96
3	Aaron	Helen	Kingston	Green Acres	483029627	11128.44	718.92	23364.04
4	Aaron	Elizabeth	Mountain View	Jamestown	679919093	10255.25	270.72	21537.62
5	Aaron	Brandon	Lakeside	Newtown	1097129696	18685.51	715.38	50296.93
6	Aaron	Darrell	Greenwood	Harmony	1316979877	13860.71	784.64	40841.15
7	Aaron	Thomas	Brownsville	Centerville	1320067447	23219.91	1067.46	51387.86
8	Aaron	Stephanie	Empire	Union Hill	1427314726	23600.93	1574.48	29306.51
9	Aaron	Nathan	Glendale	Enterprise	1440890873	10090.36	487.86	31862.53

Dashboard of the TPC-DS Dataset

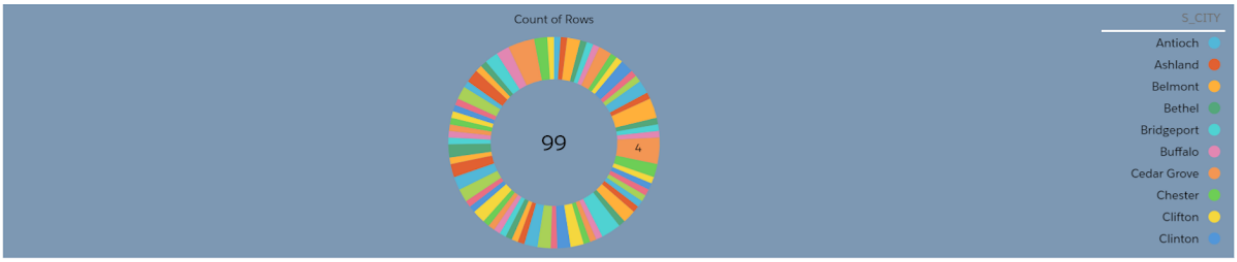
Query 4
Find customers who spend more money via catalog than in stores. Identify preferred customers and their country of origin



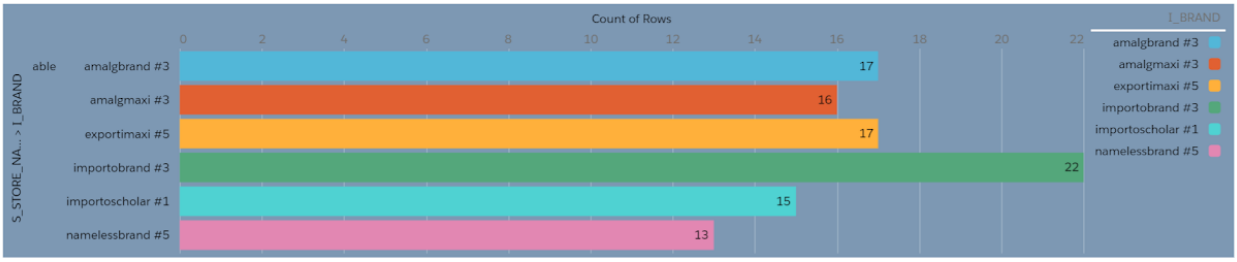
Query 45
Report the total web sales for customers in specific zip codes, cities, counties or states, or specific items for a given year and quarter



Query 50
For each store count the number of items in a specified month that were returned after 30, 60, 90, 120 and more than 120 days from the day of purchase



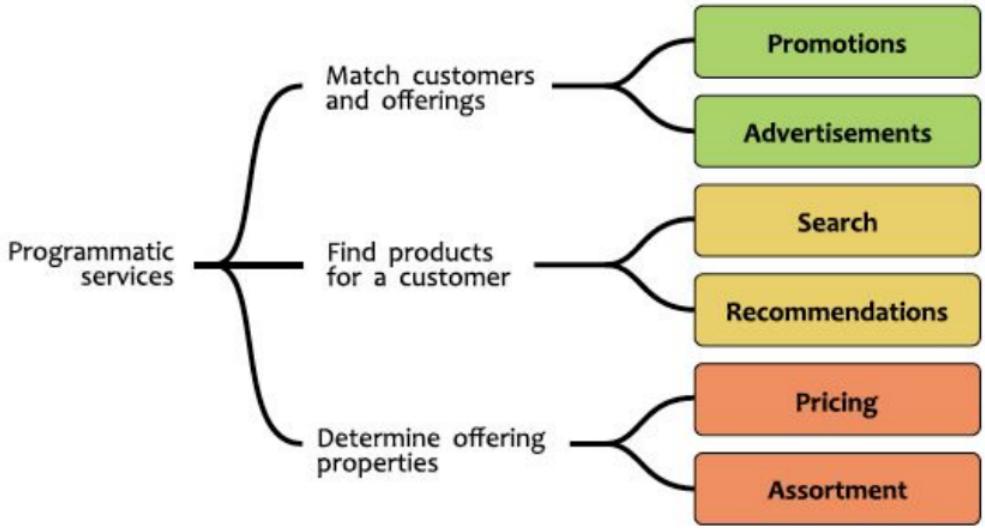
Query 65
In a given period, for each store, report the list of items with revenue less than 10% the average revenue for all the items in that store



Query 66
Compute web and catalog sales and profits by warehouse. Report results by month for a given year during a given 8-hour period.

Use-Case Accomplishment

Task 2: Discuss who is this dashboard targeted towards and the use-cases you will accomplish with it:



This dashboard will be used by Business Analyst, and Management of the company to plan for taking steps towards the growth of the organization.
Getting insights of the customer behaviour could be useful for creating promotional campaigns, do dynamic pricing, bundle products for discounts, offer recommendations and ability to search products based on keywords efficiently.

This way-

- Promotions and discounts could be aligned to match customers' choice
- Better recommendations could be given of the products which have high probability of purchase
- Company will be able to reach customers with the appropriate advertisements
- Product search could be more optimized for better results
- Pricing can be more dynamic and real-time
- Assortment offers could be more appealing

Augment with a new dataset

Task 3 - Your company wants to augment this dataset with a new dataset which will be in csv format:

i. Describe your design on how you would onboard the dataset

ii. Describe what tools (xsv, Python) will be used for data clean-up

The onboarding of a new dataset to augment the TPC-DS Dataset has to be a thoughtful process.

TPC-DS models any industry that must manage, sell and distribute products (e.g., food, electronics, furniture, music and toys etc.). It utilizes the business model of a large retail company having multiple stores located nation-wide. Beyond its brick and mortar stores, the company also sells goods through catalogs and the Internet. Along with tables to model the associated sales and returns, it includes a simple inventory system and a promotion system.

The following are examples of business processes of this retail company:

- Record customer purchases (and track customer returns) from any sales channel
- Modify prices according to promotions
- Maintain warehouse inventory
- Create dynamic web pages
- Maintain customer profiles (Customer Relationship Management)

General Implementation Guidelines

The purpose of TPC benchmarks is to provide relevant, objective performance data to industry users. To achieve that purpose, TPC benchmark specifications require benchmark tests be implemented with systems, products, technologies and pricing that:

a) Are generally available to users;

b) Are relevant to the market segment that the individual TPC benchmark models or represents (e.g., TPC-DS models and represents complex, high data volume, decision support environments);

c) Would plausibly be implemented by a significant number of users in the market segment modeled or represented by the benchmark.

In keeping with these requirements, the TPC-DS database must be implemented using commercially available data processing software, and its queries must be executed via SQL interface.

The use of new systems, products, technologies (hardware or software) and pricing is encouraged so long as they meet the requirements above. Specifically prohibited are benchmark systems, products, technologies or pricing (hereafter referred to as "implementations") whose primary purpose is performance optimization of TPC benchmark results without any corresponding applicability to real-world applications and environments. In other words, all "benchmark special" implementations, which improve benchmark results but not real-world performance or pricing, are prohibited.

Various tools for Data clean-up

TRIFACTA

XSV

- Tool is a very fast function like searching , joining for a big csv file takes ~ 6-7 sec
- Tool has a help function which gives a description of every command of xsv
- Found Commands liks frequency, stats, table really helpful to get an overview of data
- No user Interface only command line which may appear boring for few people
- Commands are limited to joining, slicing and getting statistics of data
- Limited resources available for this tool

Trifacta

- Its Interactive User Interface, Easy to use No prior coding or Technical expertise required
- It can be used to create Recipe which can be used multiple times on multiple data sets
- Its AI powered features help us in structuring, validating and cleaning data
- Data profiling feature gives us a visual downloadable report of our data within minutes to analyse the scope of our data
- Cannot download Data over 1GB on free version
- Pro - version is available is very hard since it requires the company to register

Python

- Pandas, written in python can be used for data cleanup
- It can present data in a way that is suitable for data analysis via its Series and DataFrame data structures
- The package contains multiple methods for convenient data filtering
- Pandas have the best visualization among all of them
- Requires to know programming language to get the required output

---

Prototype the application

Task 4 - Prototype your application

i. Choose a marketing related dataset from www.kaggle.com, criteo.com etc.
ii. Pre-process it with xsv, trifacta, google refine etc.
iii. Show how will you upload it to Snowflake and/or Einstein analytics
iv. Show how this data can be visualized in Einstein analytics.

Description of the dataset-

Data | Tasks (2) | Code (29) | Discussion (1) | Activity | Metadata

Download (61 KB) | New Notebook

Usability 10.0 | **License** CC0: Public Domain | **Tags** business, exploratory data analysis, regression, statistical analysis, marketing

Description

Context

This data set was provided to students for their final project in order to test their statistical analysis skills as part of a MSc. in Business Analytics.

It can be utilized for EDA, Statistical Analysis, and Visualizations. For more specific guidance on how to utilize this data set, please see the Exploratory & Statistical Analysis task.

Content

The data set `marketing_data.csv` consists of 2,240 customers of XYZ company with data on:

Data Explorer

221.73 KB

marketing_data.csv

marketing_data.csv (221.73 KB)

Detail | Compact | Column

10 of 28 columns

About this file

The is a CSV file of 2240 observations (customers) with 28 variables related to marketing data. More specifically, the variables provide insights about:

- Customer profiles
- Products purchased
- Campaign success (or failure)

ID Customer's unique identifier	Year_Birth Customer's birth year	Education Customer's education level	Marital_Status Customer's marital status	Income Customer's yearly household income
0 … 11.2k	1893 … 1996	Graduation 50% PhD 22% Other (627) 28%	Married 39% Together 26% Other (796) 36%	[null] 1% $7,500.00 1% Other (2204) 98%
1826	1970	Graduation	Divorced	$84,835.00
1	1961	Graduation	Single	$57,091.00
10476	1958	Graduation	Married	$67,267.00
1386	1967	Graduation	Together	$32,474.00
5371	1989	Graduation	Single	$21,474.00
7348	1958	PhD	Single	$71,691.00
4073	1954	2n Cycle	Married	$63,564.00
1991	1967	Graduation	Together	$44,931.00

Marketing Analytics dataset has been taken from Kaggle:
https://www.kaggle.com/jackdaoud/marketing-data

The data set `marketing_data.csv` consists of **2,240 observations** (customers) of XYZ company with **28 variables**/columns related to marketing data on:

- Customer profiles
- Product preferences
- Campaign successes/failures
- Channel performance

XSV for data pre-processing

XSV Commands on Marketing Dataset-

1. Header command to check the column header name

```
C:\Users\jshar\OneDrive\Desktop\XSV\xsv\target\release>xsv headers C:\Users\jshar\OneDrive\Desktop\snowflake_query\marke
ting_data.csv
1   C_CUSTOMER_SK
2   Year_Birth
3   Education
4   Marital_Status
5    Income
6   Kidhome
7   Teenhome
8   Dt_Customer
9   Recency
10  MntWines
11  MntFruits
12  MntMeatProducts
13  MntFishProducts
14  MntSweetProducts
15  MntGoldProds
16  NumDealsPurchases
17  NumWebPurchases
18  NumCatalogPurchases
19  NumStorePurchases
20  NumWebVisitsMonth
21  AcceptedCmp3
22  AcceptedCmp4
23  AcceptedCmp5
24  AcceptedCmp1
25  AcceptedCmp2
26  Response
27  Complain
28  Country
```

2. Count command to check the number of observations

```
C:\Users\jshar\OneDrive\Desktop\XSV\xsv\target\release>xsv count C:\Users\jshar\OneDrive\Desktop\snowflake_query\marketi
ng_data.csv
2240
```

3. Flatten command to check flattened view of CSV file

```
C:\Users\jshar\OneDrive\Desktop\XSV\xsv\target\release>xsv slice -i 5 C:\Users\jshar\OneDrive\Desktop\snowflake_query\ma
rketing_data.csv | xsv flatten
C_CUSTOMER_SK        29117862
Year_Birth           1958
Education            PhD
Marital_Status       Single
 Income              $71,691.00
Kidhome              0
Teenhome             0
Dt_Customer          3/17/14
Recency              0
MntWines             336
MntFruits            130
MntMeatProducts      411
MntFishProducts      240
MntSweetProducts     32
MntGoldProds         43
NumDealsPurchases    1
NumWebPurchases      4
NumCatalogPurchases  7
NumStorePurchases    5
NumWebVisitsMonth    2
AcceptedCmp3         0
AcceptedCmp4         0
AcceptedCmp5         0
AcceptedCmp1         0
AcceptedCmp2         0
Response             1
Complain             0
Country              SP
```

4. Stats command to check basic types and statistics of each column

```
C:\Users\jshar\OneDrive\Desktop\XSV\xsv\target\release>xsv stats C:\Users\jshar\OneDrive\Desktop\snowflake_query\marketing_data.csv | xsv table
field                type     sum          min         max          min_length  max_length  mean                    stddev
C_CUSTOMER_SK        Integer  77909305463  28831638    56401638     8           8           34780939.93883929       11144519.20342949
Year_Birth           Integer  4410125      1893        1996         4           4           1968.8058035714266      11.981394142764564
Education            Unicode               2n Cycle    PhD          3           10
Marital_Status       Unicode               Absurd      YOLO         4           8
 Income              Unicode               $1,730.00   $98,777.00   0           12
Kidhome              Integer  995          0           2            1           1           0.44419642857142827     0.5382779061720435
Teenhome             Integer  1134         0           2            1           1           0.5062499999999989      0.5444166684889168
Dt_Customer          Unicode               01-01-2013  9/30/13      7           10
Recency              Integer  110005       0           99           1           2           49.10937499999993       28.9559872534794
MntWines             Integer  680816       0           1493         1           4           303.9357142857148       336.52225087151766
MntFruits            Integer  58917        0           199          1           3           26.302232142857157      39.76455477490108
MntMeatProducts      Integer  373968       0           1725         1           4           166.94999999999985      225.66498399175742
MntFishProducts      Integer  84057        0           259          1           3           37.52544642857147       54.616784073023524
MntSweetProducts     Integer  60621        0           263          1           3           27.062946428571415      41.27128306224886
MntGoldProds         Integer  98609        0           362          1           3           44.02187499999995       52.15579309748088
NumDealsPurchases    Integer  5208         0           15           1           2           2.3249999999999984      1.9318061496951502
NumWebPurchases      Integer  9150         0           27           1           2           4.084821428571434       2.778093829454848
NumCatalogPurchases  Integer  5963         0           28           1           2           2.662053571428578       2.922448104877407
NumStorePurchases    Integer  12970        0           13           1           2           5.790178571428578       3.2502324043671025
NumWebVisitsMonth    Integer  11909        0           20           1           2           5.31651785714286        2.4261032872479698
AcceptedCmp3         Integer  163          0           1            1           1           0.0727678571428572      0.259755069459085
AcceptedCmp4         Integer  167          0           1            1           1           0.07455357142857143     0.2626696336004918
AcceptedCmp5         Integer  163          0           1            1           1           0.07276785714285718     0.2597550694590847
AcceptedCmp1         Integer  144          0           1            1           1           0.06428571428571432     0.24526121019127756
AcceptedCmp2         Integer  30           0           1            1           1           0.013392857142857135    0.11494993919271194
Response             Integer  334          0           1            1           1           0.14910714285714288     0.35619405217679107
Complain             Integer  21           0           1            1           1           0.009375000000000007    0.09636964965693309
Country              Unicode               AUS         US           2           3
```

5. Frequency command to build frequency tables of each column in CSV data

```
C:\Users\jshar\OneDrive\Desktop\XSV\xsv\target\release>xsv frequency C:\Users\jshar\OneDrive\Desktop\snowflake_query\mar
keting_data.csv
field,value,count
C_CUSTOMER_SK,28952958,1
C_CUSTOMER_SK,29067535,1
C_CUSTOMER_SK,29066191,1
C_CUSTOMER_SK,29001720,1
C_CUSTOMER_SK,28831638,1
C_CUSTOMER_SK,28852837,1
C_CUSTOMER_SK,28963098,1
C_CUSTOMER_SK,29076994,1
C_CUSTOMER_SK,56260603,1
C_CUSTOMER_SK,28955532,1
Year_Birth,1976,89
Year_Birth,1971,87
Year_Birth,1975,83
Year_Birth,1972,79
Year_Birth,1970,77
Year_Birth,1978,77
Year_Birth,1965,74
Year_Birth,1973,74
Year_Birth,1969,71
Year_Birth,1974,69
Education,Graduation,1127
Education,PhD,486
Education,Master,370
Education,2n Cycle,203
Education,Basic,54
Marital_Status,Married,864
Marital_Status,Together,580
```

XSV Commands on TPC-DS Customer Dataset-

1. Sample command to randomly draw rows from CSV data

```
C:\Users\jshar\OneDrive\Desktop\XSV\xsv\target\release>xsv sample 2240 C:\Users\jshar\OneDrive\Desktop\snowflake_query\c
ustomer.csv > updated_customer.csv
```

2. Header command to check the column header name

```
C:\Users\jshar\OneDrive\Desktop\XSV\xsv\target\release>xsv headers C:\Users\jshar\OneDrive\Desktop\snowflake_query\updat
ed_customer.csv
1   C_CUSTOMER_SK
2   C_CUSTOMER_ID
3   C_CURRENT_CDEMO_SK
4   C_CURRENT_HDEMO_SK
5   C_CURRENT_ADDR_SK
6   C_FIRST_SHIPTO_DATE_SK
7   C_FIRST_SALES_DATE_SK
8   C_SALUTATION
9   C_FIRST_NAME
10  C_LAST_NAME
11  C_PREFERRED_CUST_FLAG
12  C_BIRTH_DAY
13  C_BIRTH_MONTH
14  C_BIRTH_YEAR
15  C_BIRTH_COUNTRY
16  C_LOGIN
17  C_EMAIL_ADDRESS
18  C_LAST_REVIEW_DATE
```

3. Flatten command to check flattened view of CSV file

```
C:\Users\jshar\OneDrive\Desktop\XSV\xsv\target\release>xsv slice -i 5 C:\Users\jshar\OneDrive\Desktop\snowflake_query\up
dated_customer.csv | xsv flatten
C_CUSTOMER_SK           28969530
C_CUSTOMER_ID           AAAAAAAAKDKAKLBA
C_CURRENT_CDEMO_SK      640579
C_CURRENT_HDEMO_SK      4415
C_CURRENT_ADDR_SK       20577051
C_FIRST_SHIPTO_DATE_SK  2449719
C_FIRST_SALES_DATE_SK   2449689
C_SALUTATION            Dr.
C_FIRST_NAME            Annette
C_LAST_NAME             Nelson
C_PREFERRED_CUST_FLAG   N
C_BIRTH_DAY             19
C_BIRTH_MONTH           1
C_BIRTH_YEAR            1937
C_BIRTH_COUNTRY         PORTUGAL
C_LOGIN
C_EMAIL_ADDRESS         Annette.Nelson@cd.com
C_LAST_REVIEW_DATE      2452625
```

4. Frequency command to build frequency tables of each column in CSV data

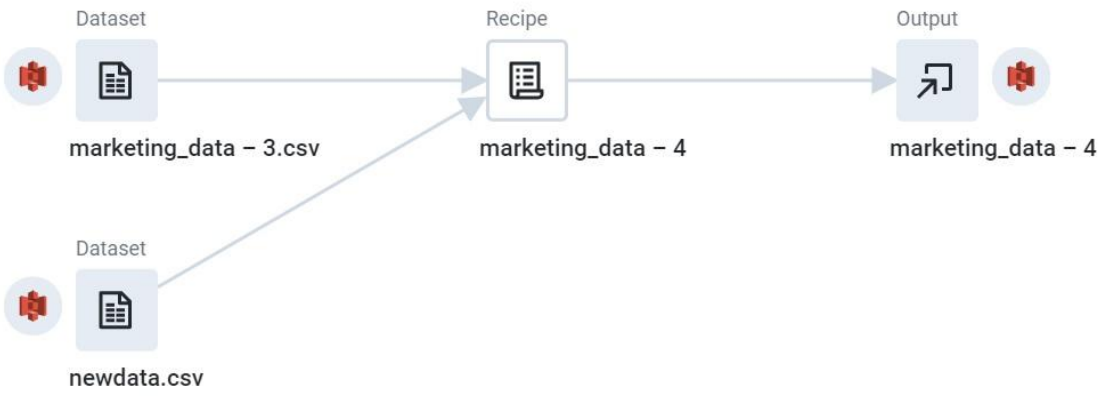
```
C:\Users\jshar\OneDrive\Desktop\XSV\xsv\target\release>xsv frequency C:\Users\jshar\OneDrive\Desktop\snowflake_query\upd
ated_customer.csv
field,value,count
C_CUSTOMER_SK,56220206,1
C_CUSTOMER_SK,29129219,1
C_CUSTOMER_SK,56226895,1
C_CUSTOMER_SK,56282666,1
C_CUSTOMER_SK,29093747,1
C_CUSTOMER_SK,28832733,1
C_CUSTOMER_SK,29034115,1
C_CUSTOMER_SK,28870350,1
C_CUSTOMER_SK,29003273,1
C_CUSTOMER_SK,28888362,1
C_CUSTOMER_ID,AAAAAAAAALKLLBA,1
C_CUSTOMER_ID,AAAAAAAACGDJKFDA,1
C_CUSTOMER_ID,AAAAAAAAGCBAKLBA,1
C_CUSTOMER_ID,AAAAAAAAKJABJLBA,1
C_CUSTOMER_ID,AAAAAAAAEMNKJFDA,1
C_CUSTOMER_ID,AAAAAAAAOFJHKLBA,1
C_CUSTOMER_ID,AAAAAAAAAJCNKFDA,1
C_CUSTOMER_ID,AAAAAAAABBCOKLBA,1
C_CUSTOMER_ID,AAAAAAAAGFCIILBA,1
C_CUSTOMER_ID,AAAAAAAAJIGNILBA,1
C_CURRENT_CDEMO_SK,(NULL),92
C_CURRENT_CDEMO_SK,1267667,2
C_CURRENT_CDEMO_SK,93667,2
C_CURRENT_CDEMO_SK,775660,1
C_CURRENT_CDEMO_SK,1234499,1
C_CURRENT_CDEMO_SK,278865,1
```

5. Stats command to check basic types and statistics of each column

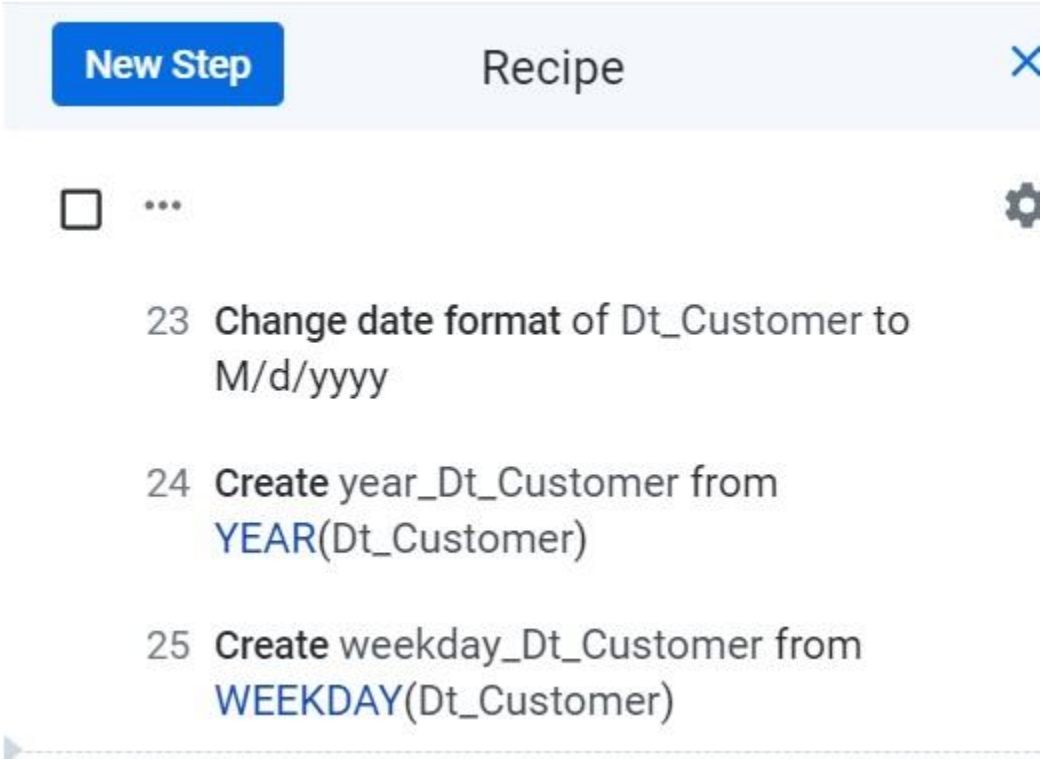
```
C:\Users\jshar\OneDrive\Desktop\XSV\xsv\target\release>xsv stats C:\Users\jshar\OneDrive\Desktop\snowflake_query\updated
_customer.csv
field,type,sum,min,max,min_length,max_length,mean,stddev
C_CUSTOMER_SK,Integer,79196667697,28831563,56401928,8,8,35355655.22187513,11523994.495715851
C_CUSTOMER_ID,Unicode,,AAAAAAAAAAAMMLBA,AAAAAAAAPPOCKLBA,16,16,,
C_CURRENT_CDEMO_SK,Integer,2053317504,1353,1919487,0,7,955920.6256983243,545767.435678365
C_CURRENT_HDEMO_SK,Integer,7784459,2,7194,0,4,3635.8986454927654,2067.547293721004
C_CURRENT_ADDR_SK,Integer,36697008252,14422,32495524,5,8,16382592.969642863,9609211.24154113
C_FIRST_SHIPTO_DATE_SK,Integer,5274156065,2449029,2452678,0,7,2450816.015334574,1051.8657999052089
C_FIRST_SALES_DATE_SK,Integer,5269189738,2448999,2452648,0,7,2450785.9246511576,1048.1818277808413
C_SALUTATION,Unicode,,Dr.,Sir,0,4,,
C_FIRST_NAME,Unicode,,Aaron,Zella,0,11,,
C_LAST_NAME,Unicode,,Aaron,Zimmerman,0,12,,
C_PREFERRED_CUST_FLAG,Unicode,,N,Y,0,1,,
C_BIRTH_DAY,Integer,32932,1,31,0,2,15.338612016767584,8.69920760097533
C_BIRTH_MONTH,Integer,14289,1,12,0,2,6.630626450116007,3.441521992008964
C_BIRTH_YEAR,Integer,4210775,1924,1992,0,4,1957.5894932589483,20.083675755892084
C_BIRTH_COUNTRY,Unicode,,AFGHANISTAN,ZIMBABWE,0,20,,
C_LOGIN,NULL,,,,0,0,,
C_EMAIL_ADDRESS,Unicode,,Aaron.Peltier@LHSVMACGukbHqmzRb.com,Zella.French@Ln8R.com,0,43,,
C_LAST_REVIEW_DATE,Integer,5297330778,2452283,2452648,0,7,2452467.9527777783,105.19618617080137
```

Data Wrangling using Trifacta

Data Flow-


Dataset
Recipe
Output
marketing_data – 3.csv
marketing_data – 4
marketing_data – 4
Dataset
newdata.csv

Recipes-


New Step
Recipe
23 Change date format of Dt_Customer to M/d/yyyy
24 Create year_Dt_Customer from YEAR(Dt_Customer)
25 Create weekday_Dt_Customer from WEEKDAY(Dt_Customer)

```
26 Create column1 from
   IF(weekday_Dt_Customer == 1,
   'Monday', IF(weekday_Dt_Customer ==
   2, 'Tuesday', IF(weekday_Dt_Customer
   == 3, 'Wednesday',
   IF(weekday_Dt_Customer == 4,
   'Thursday', IF(weekday_Dt_Customer
   == 5, 'Friday', IF(weekday_Dt_Customer
   == 6, 'Saturday',
   IF(weekday_Dt_Customer == 7,
   'Sunday', false)))))))

27 Left join with newdata.csv on
   C_CUSTOMER_SK ==
   C_CUSTOMER_SK

28 Rename column1 to 'weekofday'

29 Delete rows with missing values in
   C_CUSTOMER_SK
```

Profile-

All Data

31 columns 2,241 rows 3 data types

● 99.9% valid values ● 0.03% mismatching values ● 0.04% missing values

Job Type
Manual

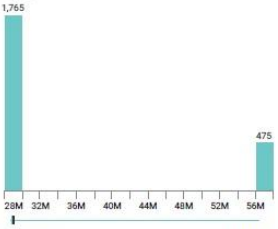
Start time
Fri, Mar 5, 2021 7:20 PM -05:00

End time
Fri, Mar 5, 2021 7:21 PM -05:00

Duration
15 sec

C_CUSTOMER_SK

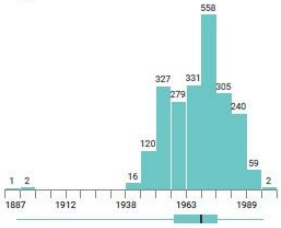
Type	Integer
Valid	2,240
Mismatched	1
Empty	0



Minimum	28,831,628
Lower quartile	28,934,450
Median	29,043,781
Upper quartile	29,154,678
Maximum	56,401,638

Year_Birth

Type	Date / Time (yyyy)
Valid	2,240
Mismatched	1
Empty	0



Minimum	1893
Lower quartile	1959
Median	1970
Upper quartile	1977
Maximum	1996

Education

Type	String
Valid	2,241
Mismatched	0
Empty	0

Top 6 values

Graduation	1,127
PhD	486
Master	370
2n Cycle	203
Basic	54
Education	1

Marital_Status

Type	String
Valid	2,241
Mismatched	0
Empty	0

Top 9 values

Married	864
Together	580
Single	480
Divorced	232
Widow	77
Alone	3
Absurd	2
YOLO	2
Marital_Status	1

Income

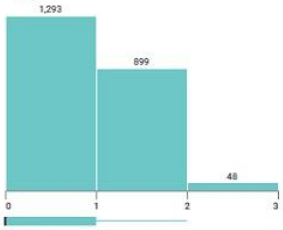
Type	String
Valid	2,217
Mismatched	0
Empty	24

Top 20 values

"$7,500.00 "	12
"$35,860.00 "	4
"$18,690.00 "	3
"$18,929.00 "	3
"$34,176.00 "	3
"$39,922.00 "	3
"$46,098.00 "	3
"$46,432.00 "	3
"$63,841.00 "	3
"$67,445.00 "	3
"$80,134.00 "	3
"$83,844.00 "	3
"$10,979.00 "	2
"$15,287.00 "	2
"$15,315.00 "	2
"$17,144.00 "	2
"$18,701.00 "	2
"$19,514.00 "	2
"$19,966.00 "	2
"$20,130.00 "	2

Kidhome

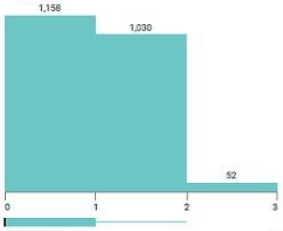
Type	Integer
Valid	2,240
Mismatched	1
Empty	0



Minimum	0
Lower quartile	0
Median	0
Upper quartile	1
Maximum	2

Teenhome

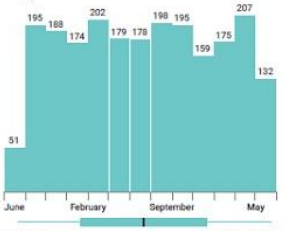
Type	Integer
Valid	2,240
Mismatched	1
Empty	0



Minimum	0
Lower quartile	0
Median	0
Upper quartile	1
Maximum	2

Dt_Customer

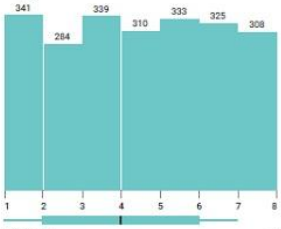
Type	Date / Time (mm*dd*yyyy)
Valid	2,240
Mismatched	0
Empty	1



Minimum	Jul 30 2012
Lower quartile	Jan 16 2013
Median	Jul 8 2013
Upper quartile	Dec 30 2013
Maximum	Jun 29 2014

weekday_Dt_Customer

Type	Integer
Valid	2,240
Mismatched	0
Empty	1



Minimum	1
Lower quartile	2
Median	4
Upper quartile	6
Maximum	7

weekofday

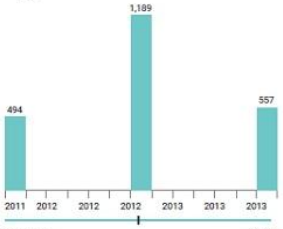
Type	String
Valid	2,241
Mismatched	0
Empty	0

Top 8 values

Monday	341
Wednesday	339
Friday	333
Saturday	325
Thursday	310
Sunday	308
Tuesday	284
false	1

year_Dt_Customer

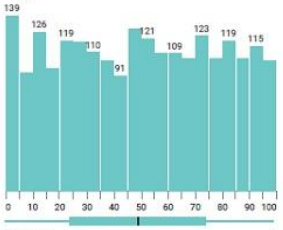
Type	Date / Time (yyyy)
Valid	2,240
Mismatched	0
Empty	1



Minimum	2012
Lower quartile	2013
Median	2013
Upper quartile	2013
Maximum	2014

Recency

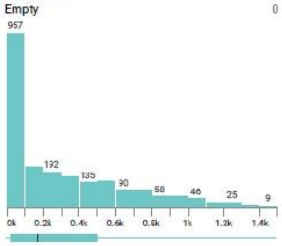
Type	Integer
Valid	2,240
Mismatched	1
Empty	0



Minimum	0
Lower quartile	24
Median	49
Upper quartile	74
Maximum	99

MntWines

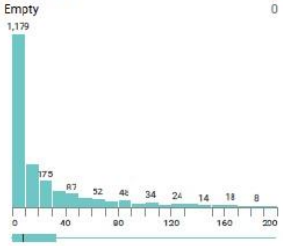
Type	Integer
Valid	2,240
Mismatched	1
Empty	0



Minimum	0
Lower quartile	24
Median	173
Upper quartile	504
Maximum	1,493

MntFruits

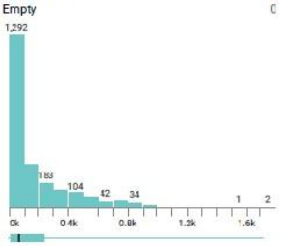
Type	Integer
Valid	2,240
Mismatched	1
Empty	0



Minimum	0
Lower quartile	1
Median	8
Upper quartile	33
Maximum	199

MntMeatProducts

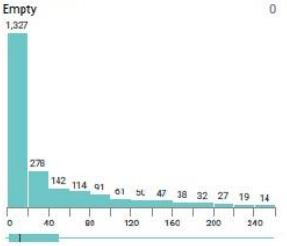
Type	Integer
Valid	2,240
Mismatched	1
Empty	0



Minimum	0
Lower quartile	16
Median	67
Upper quartile	232
Maximum	1,725

MntFishProducts

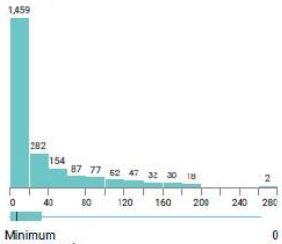
Type	Integer
Valid	2,240
Mismatched	1
Empty	0



Minimum	0
Lower quartile	3
Median	12
Upper quartile	50
Maximum	259

MntSweetProducts

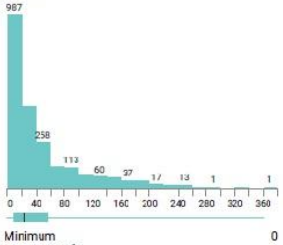
Type	Integer
Valid	2,240
Mismatched	1
Empty	0



Minimum	0
Lower quartile	1
Median	8
Upper quartile	33
Maximum	263

MntGoldProds

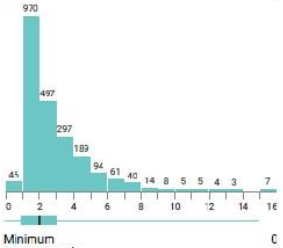
Type	Integer
Valid	2,240
Mismatched	1
Empty	0



Minimum	0
Lower quartile	9
Median	24
Upper quartile	57
Maximum	362

NumDealsPurchases

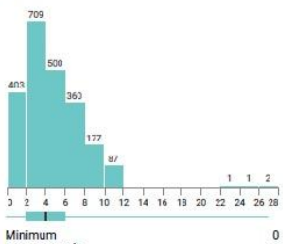
Type	Integer
Valid	2,240
Mismatched	1
Empty	0



Minimum	0
Lower quartile	1
Median	2
Upper quartile	3
Maximum	15

NumWebPurchases

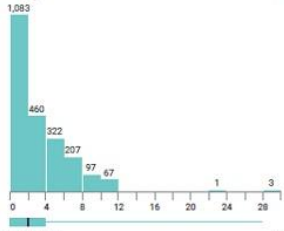
Type	Integer
Valid	2,240
Mismatched	1
Empty	0



Minimum	0
Lower quartile	2
Median	4
Upper quartile	6
Maximum	27

NumCatalogPurchases

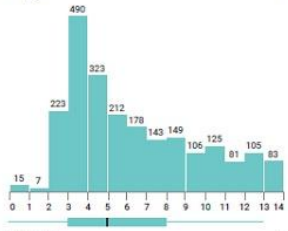
Type	Integer
Valid	2,240
Mismatched	1
Empty	0



Minimum	0
Lower quartile	0
Median	2
Upper quartile	4
Maximum	28

NumStorePurchases

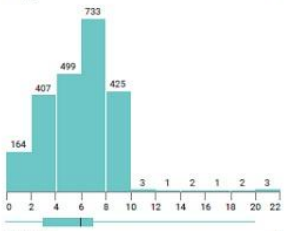
Type	Integer
Valid	2,240
Mismatched	1
Empty	0



Minimum	0
Lower quartile	3
Median	5
Upper quartile	8
Maximum	13

NumWebVisitsMonth

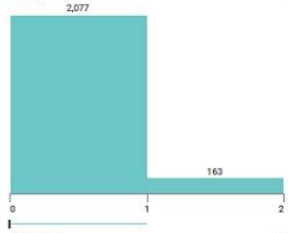
Type	Integer
Valid	2,240
Mismatched	1
Empty	0



Minimum	0
Lower quartile	3
Median	6
Upper quartile	7
Maximum	20

AcceptedCmp3

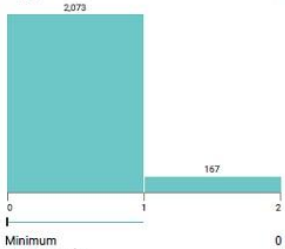
Type	Integer
Valid	2,240
Mismatched	1
Empty	0



Minimum	0
Lower quartile	0
Median	0
Upper quartile	0
Maximum	1

AcceptedCmp4

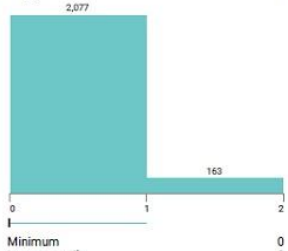
Type	Integer
Valid	2,240
Mismatched	1
Empty	0



Minimum	0
Lower quartile	0
Median	0
Upper quartile	0
Maximum	1

AcceptedCmp5

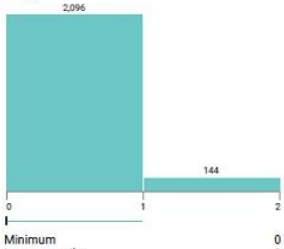
Type	Integer
Valid	2,240
Mismatched	1
Empty	0



Minimum	0
Lower quartile	0
Median	0
Upper quartile	0
Maximum	1

AcceptedCmp1

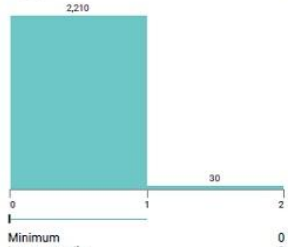
Type	Integer
Valid	2,240
Mismatched	1
Empty	0



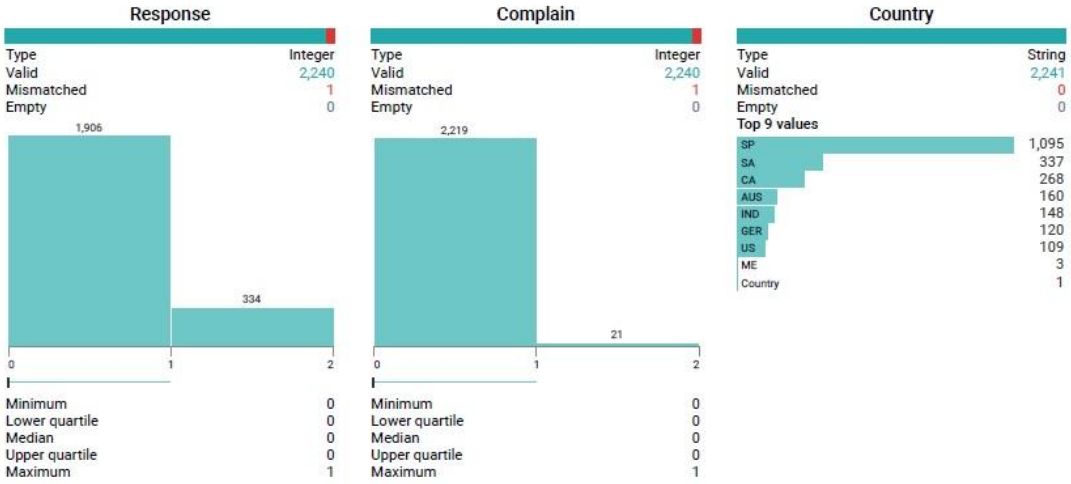
Minimum	0
Lower quartile	0
Median	0
Upper quartile	0
Maximum	1

AcceptedCmp2

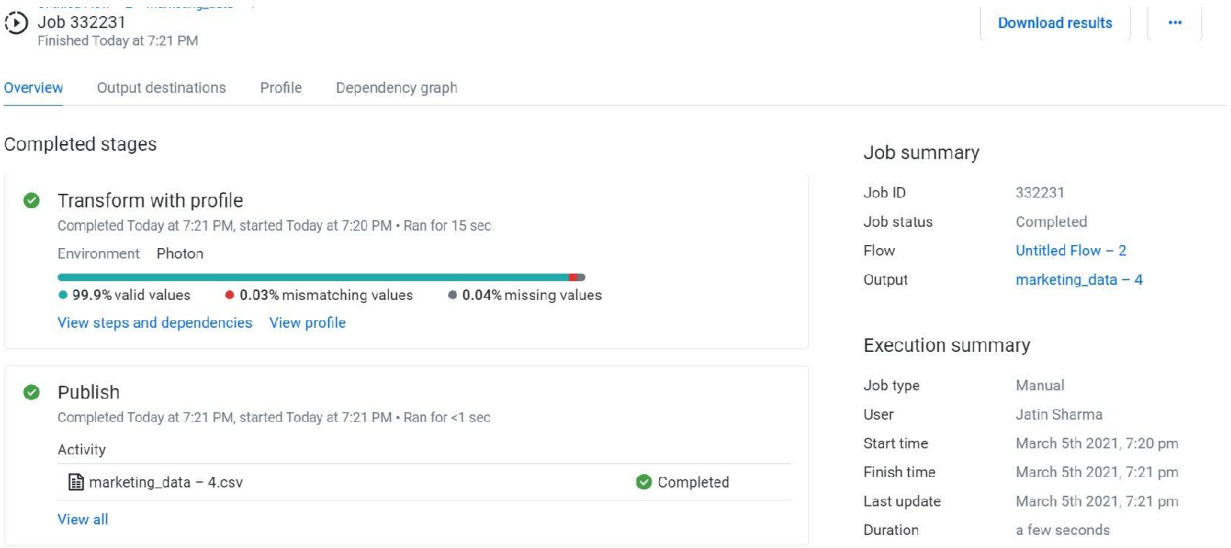
Type	Integer
Valid	2,240
Mismatched	1
Empty	0

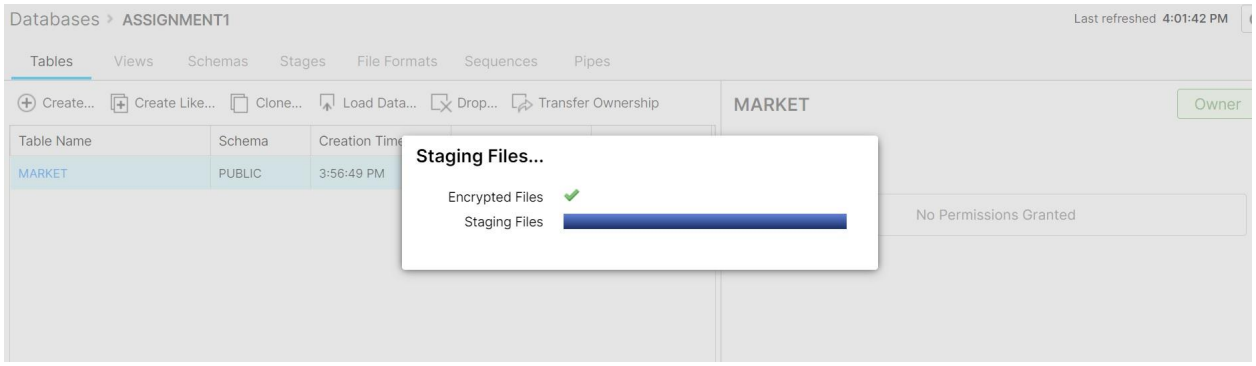


Minimum	0
Lower quartile	0
Median	0
Upper quartile	0
Maximum	1



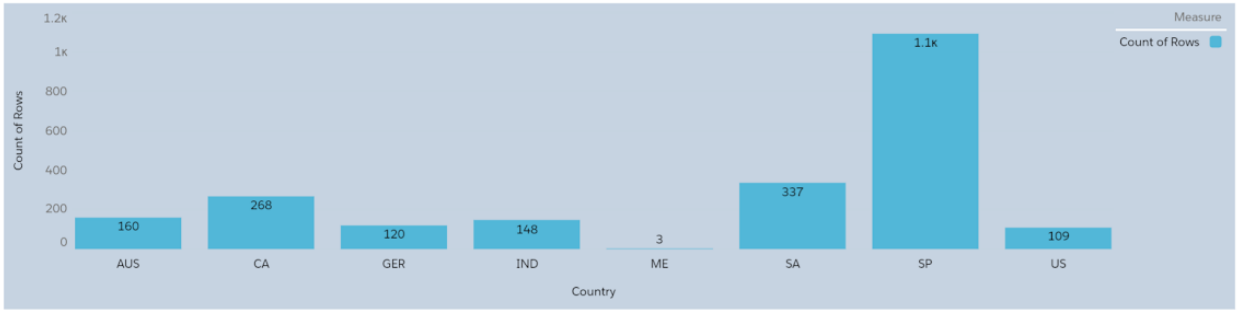
Staging on Snowflake



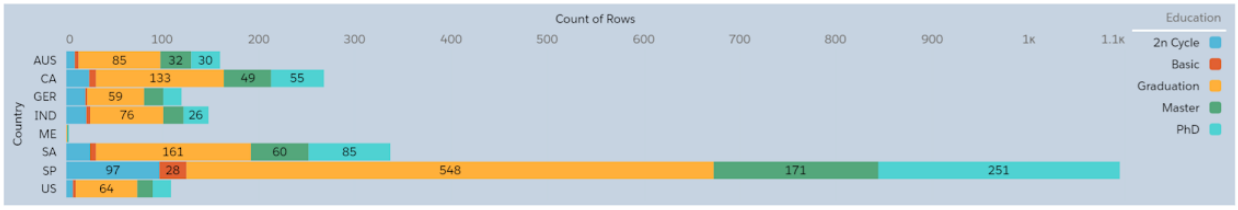


Analytical dashboard using Salesforce Einstein Analytics

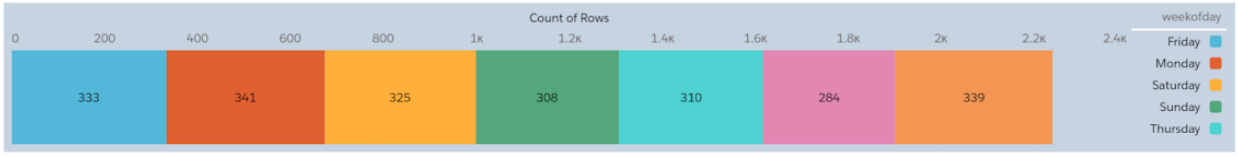
Total number of Customer



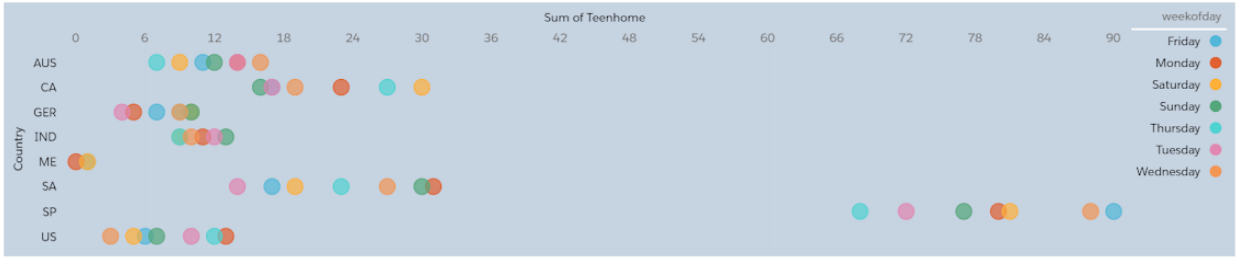
Education vs Country



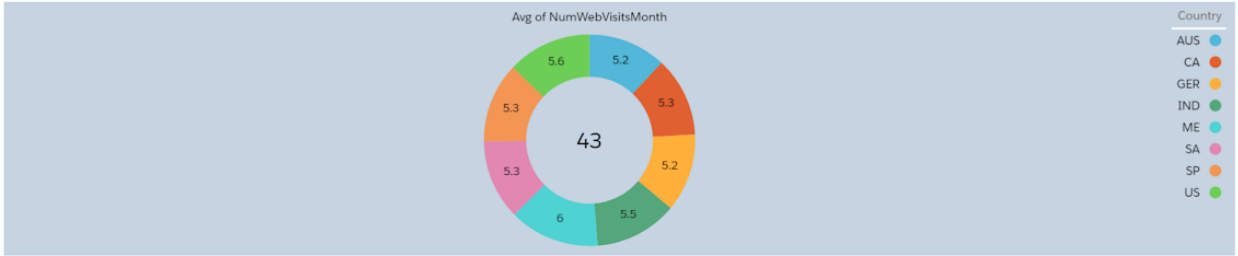
Customer purchased items according to days of week



Number of Teenhome vs Country



Average number of Wesite visit by customer per month



Total number of Kidhome vs Country