

Advanced SQL Operations Report

William Stults

1/12/2023

Western Governors University

1. Alignment With Needs

Specific needs outlined in the data dictionary for the churn data set include the following:

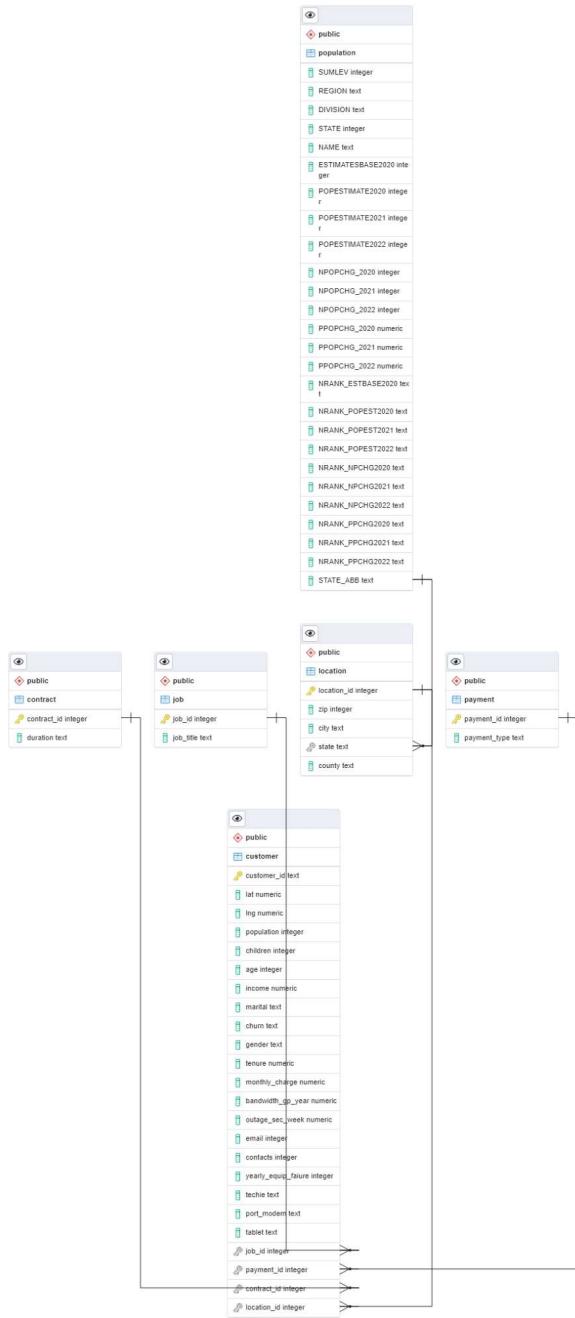
- Recruitment and retention
- Customer outreach and promotions

My churn analysis dashboard presents data that can be used to address customer recruitment by illustrating which states exhibit the largest percentage of non-customer residents, along with the total estimated population for those states (Lowest Subscriber to Population Ratios worksheet). This information can be used to identify target markets for more aggressive sales strategies and promotions.

The dashboard also identifies states with the largest degree of churn, cross-referenced with each state's total estimated population (Churn % With Total Population worksheet). The Churn % By State worksheet allows the user to view churn percentages for individual states. This is useful for identifying markets where retention should be more closely managed.

2. Analysis of Database Tables

An ERD illustrating the database tables and their relations is shown below.



The "customer" table has dependencies referencing columns in the other four tables native to the "churn" database: "contract", "job", "location", and "payment". These four tables would have been created first, in any order, specifying their primary key which is then referenced by the "customer" table.

When the customer table is created, foreign keys are added for the four "id" columns in the existing four tables, establishing the relational link between the tables and enforcing referential integrity.

Lastly, the "population" table was created and populated with data. While a foreign key / primary key link is not present, it does share a one to many relationship with the location table, as illustrated in the ERD.

3. Selection of Business Intelligence Tool

Tableau Desktop is the BI tool utilized for the project. It is a fitting choice due to its variety of visualizations, the ease with which they can be created, and the ability to work with large amounts of data (much more than needed for this operation). It is considered an industry standard tool for business-related data visualization.

4. Cleaning and Preparing Data

The existing "churn" database required no preparation, cleaning or alteration.

Before utilizing the second data source, I needed to do some preparation. This involved creating a table in the churn database to house the data from NST-EST2022-POPCHG2020_2022.csv, importing the data, and then executing SQL statements to prepare the data for use.

The create table script for the new population table is shown below.

```
-- Table: public.population

-- DROP TABLE public.population;

CREATE TABLE public.population
(
    "SUMLEV" integer NOT NULL,
    "REGION" text COLLATE pg_catalog."default" NOT NULL,
    "DIVISION" text COLLATE pg_catalog."default" NOT NULL,
    "STATE" integer NOT NULL,
    "NAME" text COLLATE pg_catalog."default" NOT NULL,
    "ESTIMATESBASE2020" integer NOT NULL,
    "POPESTIMATE2020" integer NOT NULL,
    "POPESTIMATE2021" integer NOT NULL,
    "POPESTIMATE2022" integer NOT NULL,
    "NPOPCHG_2020" integer NOT NULL,
    "NPOPCHG_2021" integer NOT NULL,
    "NPOPCHG_2022" integer NOT NULL,
    "PPOPCHG_2020" numeric NOT NULL,
    "PPOPCHG_2021" numeric NOT NULL,
    "PPOPCHG_2022" numeric NOT NULL,
    "NRANK_ESTBASE2020" text COLLATE pg_catalog."default" NOT NULL,
    "NRANK_POPEST2020" text COLLATE pg_catalog."default" NOT NULL,
    "NRANK_POPEST2021" text COLLATE pg_catalog."default" NOT NULL,
    "NRANK_POPEST2022" text COLLATE pg_catalog."default" NOT NULL,
    "NRANK_NPCHG2020" text COLLATE pg_catalog."default" NOT NULL,
    "NRANK_NPCHG2021" text COLLATE pg_catalog."default" NOT NULL,
    "NRANK_NPCHG2022" text COLLATE pg_catalog."default" NOT NULL,
    "NRANK_PPCHG2020" text COLLATE pg_catalog."default" NOT NULL,
    "NRANK_PPCHG2021" text COLLATE pg_catalog."default" NOT NULL,
    "NRANK_PPCHG2022" text COLLATE pg_catalog."default" NOT NULL,
)
```

```
TABLESPACE pg_default;  
  
ALTER TABLE public.population  
OWNER to postgres;
```

--

Once done, I right-clicked the new table and selected "import/export" from the menu to utilize PgAdmin's built in tool to import the data from NST-EST2022-POPCHG2020_2022.csv.

I trimmed the table by deleting rows containing data on regions rather than states:

```
DELETE FROM public.population WHERE "STATE" = 0
```

Then I created a new column "STATE_ABB" and assigned it the values in the "NAME" column:

```
UPDATE population SET "STATE_ABB" = "NAME"
```

Lastly, I executed the update statements below to transform state names in the STATE_ABB column to state name abbreviations.

```
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'Alabama',  
'AL');  
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'Alaska', 'AK');  
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'Arizona',  
'AZ');  
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'Arkansas',  
'AR');  
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'California',  
'CA');  
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'Canal Zone',  
'CZ');  
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'Colorado',  
'CO');  
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'Connecticut',  
'CT');  
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'Delaware',  
'DE');  
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'District of  
Columbia', 'DC');  
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'Florida',  
'FL');  
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'Georgia',  
'GA');  
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'Guam', 'GU');  
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'Hawaii', 'HI');  
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'Idaho', 'ID');  
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'Illinois',  
'IL');  
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'Indiana',  
'IN');  
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'Iowa', 'IA');
```

```
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'Kansas', 'KS');
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'Kentucky',
'KY');
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'Louisiana',
'LA');
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'Maine', 'ME');
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'Maryland',
'MD');
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'Massachusetts',
'MA');
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'Michigan',
'MI');
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'Minnesota',
'MN');
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'Mississippi',
'MS');
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'Missouri',
'MO');
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'Montana',
'MT');
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'Nebraska',
'NE');
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'Nevada', 'NV');
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'New Hampshire',
'NH');
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'New Jersey',
'NJ');
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'New Mexico',
'NM');
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'New York',
'NY');
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'North
Carolina', 'NC');
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'North Dakota',
'ND');
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'Ohio', 'OH');
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'Oklahoma',
'OK');
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'Oregon', 'OR');
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'Pennsylvania',
'PA');
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'Puerto Rico',
'PR');
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'Rhode Island',
'RI');
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'South
Carolina', 'SC');
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'South Dakota',
'SD');
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'Tennessee',
'TN');
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'Texas', 'TX');
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'Utah', 'UT');
```

```
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'Vermont',  
'VT');  
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'Virgin  
Islands', 'VI');  
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'Virginia',  
'VA');  
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'Washington',  
'WA');  
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'West VA',  
'WV');  
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'Wisconsin',  
'WI');  
UPDATE population SET "STATE_ABB" = replace("STATE_ABB", 'Wyoming',  
'WY');
```

5. Creating the Dashboard

The process below was the one I used to create my churn analysis dashboard.

Launch the Tableau application. In the left-hand panel of the screen which is colored blue, look for the section named "To a Server". Click the More... option and select PostgreSQL. Enter the appropriate details and click "Sign In". Drag the "customer" table to the right hand pane.

With the "customer" table selected in the right pane, open the "Data" menu on the top toolbar and select "Convert to Custom SQL". Paste in the SQL shown below, then click "Ok".

```
SELECT "customer"."age" AS "age",  
"customer"."bandwidth_gp_year" AS "bandwidth_gp_year",  
"customer"."children" AS "children",  
CAST("customer"."churn" AS TEXT) AS "churn",  
"customer"."contacts" AS "contacts",  
"customer"."contract_id" AS "contract_id",  
CAST("customer"."customer_id" AS TEXT) AS "customer_id",  
"customer"."email" AS "email",  
CAST("customer"."gender" AS TEXT) AS "gender",  
"customer"."income" AS "income",  
"customer"."job_id" AS "job_id",  
"customer"."lat" AS "lat",  
"customer"."lng" AS "lng",  
"customer"."location_id" AS "location_id",  
CAST("customer"."marital" AS TEXT) AS "marital",  
"customer"."monthly_charge" AS "monthly_charge",  
"customer"."outage_sec_week" AS "outage_sec_week",  
"customer"."payment_id" AS "payment_id",  
"customer"."population" AS "population",  
CAST("customer"."port_modem" AS TEXT) AS "port_modem",  
CAST("customer"."tablet" AS TEXT) AS "tablet",  
CAST("customer"."techie" AS TEXT) AS "techie",  
"customer"."tenure" AS "tenure",  
"customer"."yearly_equip_faiure" AS "yearly_equip_faiure",  
CAST("location"."city" AS TEXT) AS "city",
```

```

CAST("location"."county" AS TEXT) AS "county",
"location"."location_id" AS "location_id (location)",
CAST("location"."state" AS TEXT) AS "state",
"location"."zip" AS "zip",
"population"."POPESTIMATE2022" AS "POPESTIMATE2022",
CAST("population"."STATE_ABB" AS TEXT) AS "STATE_ABB"
FROM "public"."customer" "customer"
INNER JOIN "public"."location"
ON "location"."location_id" = "customer"."location_id"
INNER JOIN "public"."population"
ON "population"."STATE_ABB" = "location"."state"

```

Click "Update Now" in the lower right pane to view the results of the join.

Once all the data was imported I created some calculated fields in Tableau.

For each calculated field, a brief description and the code needed to create the field are included.

Churn Percent

The churn percent field is a calculation of the percentage of customers that churned.

```
SUM([Churn Yes Count])/COUNT([Churn])
```

Churn Yes Count

The churn yes count field gives us a count of the customers that churned.

```

IF [Churn]='Yes' THEN 1
END

```

Pct of State

The pct of state field gives us a percent value for how many customers in that state are subscribers compared to the total population estimate.

```
COUNT([State])/MAX([POPESTIMATE2022])
```

Below I have noted the process I used for creating each worksheet, and finally, the dashboard itself.

Worksheet 1 - Lowest Subscriber to Population Ratios

1. Drag "State" to the columns shelf. From its dropdown menu, select Measure > Count.
2. Drag "Pct of State" to the first position on the columns shelf.
3. Drag "State" to the rows shelf.
4. Drag "State" to the Filters shelf. From the "Top" tab, enable the "By field:" radio button. From the first dropdown menu, select bottom. In the field to the right, type "20". From the second dropdown menu, select "Pct of State". Then click Apply, and OK.

5. Drag "State" to the Color tile on the marks shelf. Click the Color tile and choose "Edit Colors".

From the "Select Color Palette" dropdown, choose "Color Blind". Click "Assign Palette", then Apply, and then OK.

The underlying SQL for Worksheet 1 is shown below.

```
SELECT COUNT(CAST("Custom SQL Query1"."state" AS TEXT)) AS
"TEMP(Calculation_242912952012296194)(858378402)(0)",
CAST("Custom SQL Query1"."state" AS TEXT) AS "state"
FROM (
    SELECT CAST("location"."city" AS TEXT) AS "city",
    CAST("location"."county" AS TEXT) AS "county",
    "location"."location_id" AS "location_id (location)",
    CAST("location"."state" AS TEXT) AS "state",
    "location"."zip" AS "zip"
    FROM "public"."location" "location"
) "Custom SQL Query1"
INNER JOIN (
    SELECT "t0"."state" AS "state",
    (CASE WHEN "t2"."__measure_1" = 0 THEN NULL ELSE
    CAST(COALESCE("t1"."__measure_0", 0) AS DOUBLE PRECISION) /
    "t2"."__measure_1" END) AS "$__alias_0"
    FROM (
        SELECT CAST("Custom SQL Query1"."state" AS TEXT) AS "state"
        FROM (
            SELECT CAST("location"."city" AS TEXT) AS "city",
            CAST("location"."county" AS TEXT) AS "county",
            "location"."location_id" AS "location_id (location)",
            CAST("location"."state" AS TEXT) AS "state",
            "location"."zip" AS "zip"
            FROM "public"."location" "location"
        ) "Custom SQL Query1"
        GROUP BY 1
    ) "t0"
    LEFT JOIN (
        SELECT CAST("Custom SQL Query1"."state" AS TEXT) AS "state",
        COUNT(CAST("Custom SQL Query1"."state" AS TEXT)) AS "__measure_0"
        FROM (
            SELECT CAST("location"."city" AS TEXT) AS "city",
            CAST("location"."county" AS TEXT) AS "county",
            "location"."location_id" AS "location_id (location)",
            CAST("location"."state" AS TEXT) AS "state",
            "location"."zip" AS "zip"
            FROM "public"."location" "location"
        ) "Custom SQL Query1"
        GROUP BY 1
    ) "t1" ON ("t0"."state" IS NOT DISTINCT FROM "t1"."state")
    LEFT JOIN (
        SELECT CAST("Custom SQL Query1"."state" AS TEXT) AS "state",
        MAX("Custom SQL Query1"."POPESTIMATE2022") AS "__measure_1"
        FROM (
```

```

SELECT "population"."POPESTIMATE2022" AS "POPESTIMATE2022",
       CAST("population"."STATE_ABB" AS TEXT) AS "STATE_ABB"
  FROM "public"."population" "population"
) "Custom SQL Query"
LEFT JOIN (
  SELECT CAST("location"."city" AS TEXT) AS "city",
         CAST("location"."county" AS TEXT) AS "county",
         "location"."location_id" AS "location_id (location)",
         CAST("location"."state" AS TEXT) AS "state",
         "location"."zip" AS "zip"
    FROM "public"."location" "location"
) "Custom SQL Query1" ON (CAST("Custom SQL Query". "STATE_ABB" AS
TEXT) = CAST("Custom SQL Query1". "state" AS TEXT))
 GROUP BY 1
) "t2" ON ("t0". "state" IS NOT DISTINCT FROM "t2". "state")
ORDER BY 2 ASC NULLS FIRST,
  1 ASC NULLS FIRST
LIMIT 20
) "t3" ON (CAST("Custom SQL Query1". "state" AS TEXT) IS NOT DISTINCT FROM
"t3". "state")
GROUP BY 2

```

```

SELECT MAX("Custom SQL Query". "POPESTIMATE2022") AS
"TEMP(Calculation_242912952012296194)(3105768215)(0)",
       CAST("Custom SQL Query1". "state" AS TEXT) AS "state"
  FROM (
    SELECT "population"."POPESTIMATE2022" AS "POPESTIMATE2022",
           CAST("population"."STATE_ABB" AS TEXT) AS "STATE_ABB"
      FROM "public"."population" "population"
) "Custom SQL Query"
LEFT JOIN (
  SELECT CAST("location"."city" AS TEXT) AS "city",
         CAST("location"."county" AS TEXT) AS "county",
         "location"."location_id" AS "location_id (location)",
         CAST("location"."state" AS TEXT) AS "state",
         "location"."zip" AS "zip"
    FROM "public"."location" "location"
) "Custom SQL Query1" ON (CAST("Custom SQL Query". "STATE_ABB" AS TEXT) =
CAST("Custom SQL Query1". "state" AS TEXT))
 INNER JOIN (
  SELECT "t0". "state" AS "state",
        (CASE WHEN "t2". "__measure_1" = 0 THEN NULL ELSE
          CAST(COALESCE("t1". "__measure_0", 0) AS DOUBLE PRECISION) /
          "t2". "__measure_1" END) AS "$__alias_0"
    FROM (
      SELECT CAST("Custom SQL Query1". "state" AS TEXT) AS "state"
    FROM (
      SELECT CAST("location"."city" AS TEXT) AS "city",
             CAST("location"."county" AS TEXT) AS "county",
             "location"."location_id" AS "location_id (location)",
             CAST("location"."state" AS TEXT) AS "state",

```

```

        "location"."zip" AS "zip"
    FROM "public"."location" "location"
) "Custom SQL Query1"
GROUP BY 1
) "t0"
LEFT JOIN (
    SELECT CAST("Custom SQL Query1"."state" AS TEXT) AS "state",
    COUNT(CAST("Custom SQL Query1"."state" AS TEXT)) AS "__measure_0"
FROM (
    SELECT CAST("location"."city" AS TEXT) AS "city",
    CAST("location"."county" AS TEXT) AS "county",
    "location"."location_id" AS "location_id (location)",
    CAST("location"."state" AS TEXT) AS "state",
    "location"."zip" AS "zip"
    FROM "public"."location" "location"
) "Custom SQL Query1"
GROUP BY 1
) "t1" ON ("t0"."state" IS NOT DISTINCT FROM "t1"."state")
LEFT JOIN (
    SELECT CAST("Custom SQL Query1"."state" AS TEXT) AS "state",
    MAX("Custom SQL Query1"."POPESTIMATE2022") AS "__measure_1"
FROM (
    SELECT "population"."POPESTIMATE2022" AS "POPESTIMATE2022",
    CAST("population"."STATE_ABB" AS TEXT) AS "STATE_ABB"
    FROM "public"."population" "population"
) "Custom SQL Query"
LEFT JOIN (
    SELECT CAST("location"."city" AS TEXT) AS "city",
    CAST("location"."county" AS TEXT) AS "county",
    "location"."location_id" AS "location_id (location)",
    CAST("location"."state" AS TEXT) AS "state",
    "location"."zip" AS "zip"
    FROM "public"."location" "location"
) "Custom SQL Query1" ON (CAST("Custom SQL Query"."STATE_ABB" AS
TEXT) = CAST("Custom SQL Query1"."state" AS TEXT))
GROUP BY 1
) "t2" ON ("t0"."state" IS NOT DISTINCT FROM "t2"."state")
ORDER BY 2 ASC NULLS FIRST,
    1 ASC NULLS FIRST
LIMIT 20
) "t3" ON (CAST("Custom SQL Query1"."state" AS TEXT) IS NOT DISTINCT FROM
"t3"."state")
GROUP BY 2

```

```

SELECT "t0"."state" AS "state",
    (CASE WHEN "t2"."__measure_1" = 0 THEN NULL ELSE
    CAST(COALESCE("t1"."__measure_0", 0) AS DOUBLE PRECISION) /
    "t2"."__measure_1" END) AS "$__alias_0"
FROM (
    SELECT CAST("Custom SQL Query1"."state" AS TEXT) AS "state"
    FROM (

```

```

SELECT CAST("location"."city" AS TEXT) AS "city",
       CAST("location"."county" AS TEXT) AS "county",
       "location"."location_id" AS "location_id (location)",
       CAST("location"."state" AS TEXT) AS "state",
       "location"."zip" AS "zip"
    FROM "public"."location" "location"
) "Custom SQL Query1"
GROUP BY 1
) "t0"
LEFT JOIN (
SELECT CAST("Custom SQL Query1"."state" AS TEXT) AS "state",
       COUNT(CAST("Custom SQL Query1"."state" AS TEXT)) AS "__measure_0"
  FROM (
SELECT CAST("location"."city" AS TEXT) AS "city",
       CAST("location"."county" AS TEXT) AS "county",
       "location"."location_id" AS "location_id (location)",
       CAST("location"."state" AS TEXT) AS "state",
       "location"."zip" AS "zip"
    FROM "public"."location" "location"
) "Custom SQL Query1"
GROUP BY 1
) "t1" ON ("t0"."state" IS NOT DISTINCT FROM "t1"."state")
LEFT JOIN (
SELECT CAST("Custom SQL Query1"."state" AS TEXT) AS "state",
       MAX("Custom SQL Query"."POPESTIMATE2022") AS "__measure_1"
  FROM (
SELECT "population"."POPESTIMATE2022" AS "POPESTIMATE2022",
       CAST("population"."STATE_ABB" AS TEXT) AS "STATE_ABB"
    FROM "public"."population" "population"
) "Custom SQL Query"
LEFT JOIN (
SELECT CAST("location"."city" AS TEXT) AS "city",
       CAST("location"."county" AS TEXT) AS "county",
       "location"."location_id" AS "location_id (location)",
       CAST("location"."state" AS TEXT) AS "state",
       "location"."zip" AS "zip"
    FROM "public"."location" "location"
) "Custom SQL Query1" ON (CAST("Custom SQL Query"."STATE_ABB" AS TEXT)
= CAST("Custom SQL Query1"."state" AS TEXT))
GROUP BY 1
) "t2" ON ("t0"."state" IS NOT DISTINCT FROM "t2"."state")
ORDER BY 2 ASC NULLS FIRST,
        1 ASC NULLS FIRST
LIMIT 20

```

Worksheet 2 - Churn % By State

1. Drag "Churn" to the "Columns" shelf.
2. Drag "Churn" to the "Rows" shelf. From its dropdown menu, highlight "Measure" and select "Count".

3. Click "Show Me" in the top right corner, and select "pie charts".
4. In the "Marks" panel, click the "Size" icon (two circles) next to the second "CNT(Churn)" and choose Label. Open that same "CNT(Churn)" dropdown menu, highlight "Quick Table Calculation" and choose "Percent of Total".
5. Drag "Churn" to the "Label" tile in the "Marks" panel.
6. Drag "State" to the "Filters" shelf in the first position. From the filter's settings menu, choose "Single Value (dropdown)".

The underlying SQL for Worksheet 2 is shown below.

```

SELECT CAST("Custom SQL Query2"."churn" AS TEXT) AS "churn",
       COUNT(CAST("Custom SQL Query2"."churn" AS TEXT)) AS "cnt:churn:ok"
  FROM (
    SELECT "customer"."age" AS "age",
           "customer"."bandwidth_gp_year" AS "bandwidth_gp_year",
           "customer"."children" AS "children",
           CAST("customer"."churn" AS TEXT) AS "churn",
           "customer"."contacts" AS "contacts",
           "customer"."contract_id" AS "contract_id",
           CAST("customer"."customer_id" AS TEXT) AS "customer_id",
           "customer"."email" AS "email",
           CAST("customer"."gender" AS TEXT) AS "gender",
           "customer"."income" AS "income",
           "customer"."job_id" AS "job_id",
           "customer"."lat" AS "lat",
           "customer"."lng" AS "lng",
           "customer"."location_id" AS "location_id",
           CAST("customer"."marital" AS TEXT) AS "marital",
           "customer"."monthly_charge" AS "monthly_charge",
           "customer"."outage_sec_week" AS "outage_sec_week",
           "customer"."payment_id" AS "payment_id",
           "customer"."population" AS "population",
           CAST("customer"."port_modem" AS TEXT) AS "port_modem",
           CAST("customer"."tablet" AS TEXT) AS "tablet",
           CAST("customer"."techie" AS TEXT) AS "techie",
           "customer"."tenure" AS "tenure",
           "customer"."yearly_equip_faiure" AS "yearly_equip_faiure",
           CAST("location"."city" AS TEXT) AS "city",
           CAST("location"."county" AS TEXT) AS "county",
           "location"."location_id" AS "location_id (location)",
           CAST("location"."state" AS TEXT) AS "state",
           "location"."zip" AS "zip",
           "population"."POPESTIMATE2022" AS "POPESTIMATE2022",
           CAST("population"."STATE_ABB" AS TEXT) AS "STATE_ABB"
  FROM "public"."customer" "customer"
  INNER JOIN "public"."location"
  ON "location"."location_id" = "customer"."location_id"
  INNER JOIN "public"."population"
  ON "population"."STATE_ABB" = "location"."state"

```

```

) "Custom SQL Query2"
GROUP BY 1

SELECT CAST("Custom SQL Query1"."state" AS TEXT) AS "state"
FROM (
    SELECT "customer"."age" AS "age",
        "customer"."bandwidth_gp_year" AS "bandwidth_gp_year",
        "customer"."children" AS "children",
        CAST("customer"."churn" AS TEXT) AS "churn",
        "customer"."contacts" AS "contacts",
        "customer"."contract_id" AS "contract_id",
        CAST("customer"."customer_id" AS TEXT) AS "customer_id",
        "customer"."email" AS "email",
        CAST("customer"."gender" AS TEXT) AS "gender",
        "customer"."income" AS "income",
        "customer"."job_id" AS "job_id",
        "customer"."lat" AS "lat",
        "customer"."lng" AS "lng",
        "customer"."location_id" AS "location_id",
        CAST("customer"."marital" AS TEXT) AS "marital",
        "customer"."monthly_charge" AS "monthly_charge",
        "customer"."outage_sec_week" AS "outage_sec_week",
        "customer"."payment_id" AS "payment_id",
        "customer"."population" AS "population",
        CAST("customer"."port_modem" AS TEXT) AS "port_modem",
        CAST("customer"."tablet" AS TEXT) AS "tablet",
        CAST("customer"."techie" AS TEXT) AS "techie",
        "customer"."tenure" AS "tenure",
        "customer"."yearly_equip_faiure" AS "yearly_equip_faiure",
        CAST("location"."city" AS TEXT) AS "city",
        CAST("location"."county" AS TEXT) AS "county",
        "location"."location_id" AS "location_id (location)",
        CAST("location"."state" AS TEXT) AS "state",
        "location"."zip" AS "zip",
        "population"."POPESTIMATE2022" AS "POPESTIMATE2022",
        CAST("population"."STATE_ABB" AS TEXT) AS "STATE_ABB"
    FROM "public"."customer" "customer"
    INNER JOIN "public"."location"
    ON "location"."location_id" = "customer"."location_id"
    INNER JOIN "public"."population"
    ON "population"."STATE_ABB" = "location"."state"
) "Custom SQL Query2"
LEFT JOIN (
    SELECT CAST("location"."city" AS TEXT) AS "city",
        CAST("location"."county" AS TEXT) AS "county",
        "location"."location_id" AS "location_id (location)",
        CAST("location"."state" AS TEXT) AS "state",
        "location"."zip" AS "zip"
    FROM "public"."location" "location"
) "Custom SQL Query1" ON ("Custom SQL Query2"."location_id" = "Custom SQL
Query1"."location_id (location)")

```

```
WHERE ("Custom SQL Query1"."location_id (location)" IS NULL)
LIMIT 1
```

```
SELECT CAST("Custom SQL Query2"."churn" AS TEXT) AS "churn"
FROM (
    SELECT "customer"."age" AS "age",
        "customer"."bandwidth_gp_year" AS "bandwidth_gp_year",
        "customer"."children" AS "children",
        CAST("customer"."churn" AS TEXT) AS "churn",
        "customer"."contacts" AS "contacts",
        "customer"."contract_id" AS "contract_id",
        CAST("customer"."customer_id" AS TEXT) AS "customer_id",
        "customer"."email" AS "email",
        CAST("customer"."gender" AS TEXT) AS "gender",
        "customer"."income" AS "income",
        "customer"."job_id" AS "job_id",
        "customer"."lat" AS "lat",
        "customer"."lng" AS "lng",
        "customer"."location_id" AS "location_id",
        CAST("customer"."marital" AS TEXT) AS "marital",
        "customer"."monthly_charge" AS "monthly_charge",
        "customer"."outage_sec_week" AS "outage_sec_week",
        "customer"."payment_id" AS "payment_id",
        "customer"."population" AS "population",
        CAST("customer"."port_modem" AS TEXT) AS "port_modem",
        CAST("customer"."tablet" AS TEXT) AS "tablet",
        CAST("customer"."techie" AS TEXT) AS "techie",
        "customer"."tenure" AS "tenure",
        "customer"."yearly_equip_faiure" AS "yearly_equip_faiure",
        CAST("location"."city" AS TEXT) AS "city",
        CAST("location"."county" AS TEXT) AS "county",
        "location"."location_id" AS "location_id (location)",
        CAST("location"."state" AS TEXT) AS "state",
        "location"."zip" AS "zip",
        "population"."POPESTIMATE2022" AS "POPESTIMATE2022",
        CAST("population"."STATE_ABB" AS TEXT) AS "STATE_ABB"
    FROM "public"."customer" "customer"
    INNER JOIN "public"."location"
    ON "location"."location_id" = "customer"."location_id"
    INNER JOIN "public"."population"
    ON "population"."STATE_ABB" = "location"."state"
) "Custom SQL Query2"
GROUP BY 1
```

```
SELECT CAST("Custom SQL Query1"."state" AS TEXT) AS "state"
FROM (
    SELECT CAST("location"."city" AS TEXT) AS "city",
        CAST("location"."county" AS TEXT) AS "county",
        "location"."location_id" AS "location_id (location)",
```

```

    CAST("location"."state" AS TEXT) AS "state",
    "location"."zip" AS "zip"
  FROM "public"."location" "location"
) "Custom SQL Query1"
GROUP BY 1
ORDER BY 1 ASC NULLS FIRST

```

```

SELECT COUNT(CAST("Custom SQL Query2"."churn" AS TEXT)) AS "cnt:churn:ok"
FROM (
  SELECT "customer"."age" AS "age",
    "customer"."bandwidth_gp_year" AS "bandwidth_gp_year",
    "customer"."children" AS "children",
    CAST("customer"."churn" AS TEXT) AS "churn",
    "customer"."contacts" AS "contacts",
    "customer"."contract_id" AS "contract_id",
    CAST("customer"."customer_id" AS TEXT) AS "customer_id",
    "customer"."email" AS "email",
    CAST("customer"."gender" AS TEXT) AS "gender",
    "customer"."income" AS "income",
    "customer"."job_id" AS "job_id",
    "customer"."lat" AS "lat",
    "customer"."lng" AS "lng",
    "customer"."location_id" AS "location_id",
    CAST("customer"."marital" AS TEXT) AS "marital",
    "customer"."monthly_charge" AS "monthly_charge",
    "customer"."outage_sec_week" AS "outage_sec_week",
    "customer"."payment_id" AS "payment_id",
    "customer"."population" AS "population",
    CAST("customer"."port_modem" AS TEXT) AS "port_modem",
    CAST("customer"."tablet" AS TEXT) AS "tablet",
    CAST("customer"."techie" AS TEXT) AS "techie",
    "customer"."tenure" AS "tenure",
    "customer"."yearly_equip_faiure" AS "yearly_equip_faiure",
    CAST("location"."city" AS TEXT) AS "city",
    CAST("location"."county" AS TEXT) AS "county",
    "location"."location_id" AS "location_id (location)",
    CAST("location"."state" AS TEXT) AS "state",
    "location"."zip" AS "zip",
    "population"."POPESTIMATE2022" AS "POPESTIMATE2022",
    CAST("population"."STATE_ABB" AS TEXT) AS "STATE_ABB"
  FROM "public"."customer" "customer"
  INNER JOIN "public"."location"
  ON "location"."location_id" = "customer"."location_id"
  INNER JOIN "public"."population"
  ON "population"."STATE_ABB" = "location"."state"
) "Custom SQL Query2"
HAVING (COUNT(1) > 0)

```

Worksheet 3 - States: Churn % With Total Population

1. Drag "Churn Percent" to the "Columns" shelf.
2. Drag "POPESTIMATE2022" to the "Rows" shelf.
3. Drag "State" to the "Color" tile on the "Marks" shelf. If prompted, choose "Add all members". Click the "Color" tile and choose "Edit Colors". From the "Select Color Palette" dropdown, choose "Color Blind". Click "Assign Palette", then Apply, and then OK.
4. From the main dropdown menu in the "Marks" shelf, choose "Circle".
5. Click the "Size" tile in the "Marks" shelf and slide the slider to the halfway point.

The underlying SQL for Worksheet 3 is shown below.

```

SELECT "t1"."state" AS "state",
       "t1"."sum:POPESTIMATE2022:ok" AS "sum:POPESTIMATE2022:ok"
  FROM (
    SELECT CAST("t0"."state" AS TEXT) AS "state",
           SUM("Custom SQL Query"."POPESTIMATE2022") AS "sum:POPESTIMATE2022:ok"
      FROM (
        SELECT "population"."POPESTIMATE2022" AS "POPESTIMATE2022",
               CAST("population"."STATE_ABB" AS TEXT) AS "STATE_ABB"
          FROM "public"."population" "population"
      ) "Custom SQL Query"
     LEFT JOIN (
      SELECT CAST("Custom SQL Query1"."state" AS TEXT) AS "state"
        FROM (
          SELECT CAST("location"."city" AS TEXT) AS "city",
                 CAST("location"."county" AS TEXT) AS "county",
                 "location"."location_id" AS "location_id (location)",
                 CAST("location"."state" AS TEXT) AS "state",
                 "location"."zip" AS "zip"
                FROM "public"."location" "location"
      ) "Custom SQL Query1"
     GROUP BY 1
    ) "t0" ON (CAST("Custom SQL Query"."STATE_ABB" AS TEXT) = "t0"."state")
   GROUP BY 1
  ) "t1"
 INNER JOIN (
  SELECT CAST("Custom SQL Query1"."state" AS TEXT) AS "state",
         SUM((CASE WHEN (CAST("Custom SQL Query2"."churn" AS TEXT) = 'Yes')
            THEN 1 ELSE NULL END)) AS "__measure_0",
         COUNT(CAST("Custom SQL Query2"."churn" AS TEXT)) AS "__measure_1"
    FROM (
      SELECT "customer"."age" AS "age",
             "customer"."bandwidth_gp_year" AS "bandwidth_gp_year",
             "customer"."children" AS "children",
             CAST("customer"."churn" AS TEXT) AS "churn",
             "customer"."contacts" AS "contacts",
             "customer"."contract_id" AS "contract_id",
             CAST("customer"."customer_id" AS TEXT) AS "customer_id",
             "customer"."email" AS "email",

```

```

        CAST("customer"."gender" AS TEXT) AS "gender",
        "customer"."income" AS "income",
        "customer"."job_id" AS "job_id",
        "customer"."lat" AS "lat",
        "customer"."lng" AS "lng",
        "customer"."location_id" AS "location_id",
        CAST("customer"."marital" AS TEXT) AS "marital",
        "customer"."monthly_charge" AS "monthly_charge",
        "customer"."outage_sec_week" AS "outage_sec_week",
        "customer"."payment_id" AS "payment_id",
        "customer"."population" AS "population",
        CAST("customer"."port_modem" AS TEXT) AS "port_modem",
        CAST("customer"."tablet" AS TEXT) AS "tablet",
        CAST("customer"."techie" AS TEXT) AS "techie",
        "customer"."tenure" AS "tenure",
        "customer"."yearly_equip_faiure" AS "yearly_equip_faiure",
        CAST("location"."city" AS TEXT) AS "city",
        CAST("location"."county" AS TEXT) AS "county",
        "location"."location_id" AS "location_id (location)",
        CAST("location"."state" AS TEXT) AS "state",
        "location"."zip" AS "zip",
        "population"."POPESTIMATE2022" AS "POPESTIMATE2022",
        CAST("population"."STATE_ABB" AS TEXT) AS "STATE_ABB"
    FROM "public"."customer" "customer"
    INNER JOIN "public"."location"
    ON "location"."location_id" = "customer"."location_id"
    INNER JOIN "public"."population"
    ON "population"."STATE_ABB" = "location"."state"
) "Custom SQL Query2"
LEFT JOIN (
    SELECT CAST("location"."city" AS TEXT) AS "city",
        CAST("location"."county" AS TEXT) AS "county",
        "location"."location_id" AS "location_id (location)",
        CAST("location"."state" AS TEXT) AS "state",
        "location"."zip" AS "zip"
    FROM "public"."location" "location"
) "Custom SQL Query1" ON ("Custom SQL Query2"."location_id" = "Custom
SQL Query1"."location_id (location)")
GROUP BY 1
HAVING (NOT ((CASE WHEN COUNT(CAST("Custom SQL Query2"."churn" AS
TEXT)) = 0 THEN NULL ELSE CAST(SUM((CASE WHEN (CAST("Custom SQL
Query2"."churn" AS TEXT) = 'Yes') THEN 1 ELSE NULL END)) AS DOUBLE
PRECISION) / COUNT(CAST("Custom SQL Query2"."churn" AS TEXT)) END) IS
NULL))
) "t2" ON ("t1"."state" IS NOT DISTINCT FROM "t2"."state")

```

```

SELECT "t0"."state" AS "state"
FROM (
    SELECT CAST("Custom SQL Query1"."state" AS TEXT) AS "state"
    FROM (
        SELECT CAST("location"."city" AS TEXT) AS "city",

```

```

        CAST("location"."county" AS TEXT) AS "county",
        "location"."location_id" AS "location_id (location)",
        CAST("location"."state" AS TEXT) AS "state",
        "location"."zip" AS "zip"
    FROM "public"."location" "location"
) "Custom SQL Query1"
GROUP BY 1
) "t0"
INNER JOIN (
    SELECT CAST("Custom SQL Query1"."state" AS TEXT) AS "state",
        SUM((CASE WHEN (CAST("Custom SQL Query2"."churn" AS TEXT) = 'Yes')
    THEN 1 ELSE NULL END)) AS "__measure_0",
        COUNT(CAST("Custom SQL Query2"."churn" AS TEXT)) AS "__measure_1"
FROM (
    SELECT "customer"."age" AS "age",
        "customer"."bandwidth_gp_year" AS "bandwidth_gp_year",
        "customer"."children" AS "children",
        CAST("customer"."churn" AS TEXT) AS "churn",
        "customer"."contacts" AS "contacts",
        "customer"."contract_id" AS "contract_id",
        CAST("customer"."customer_id" AS TEXT) AS "customer_id",
        "customer"."email" AS "email",
        CAST("customer"."gender" AS TEXT) AS "gender",
        "customer"."income" AS "income",
        "customer"."job_id" AS "job_id",
        "customer"."lat" AS "lat",
        "customer"."lng" AS "lng",
        "customer"."location_id" AS "location_id",
        CAST("customer"."marital" AS TEXT) AS "marital",
        "customer"."monthly_charge" AS "monthly_charge",
        "customer"."outage_sec_week" AS "outage_sec_week",
        "customer"."payment_id" AS "payment_id",
        "customer"."population" AS "population",
        CAST("customer"."port_modem" AS TEXT) AS "port_modem",
        CAST("customer"."tablet" AS TEXT) AS "tablet",
        CAST("customer"."techie" AS TEXT) AS "techie",
        "customer"."tenure" AS "tenure",
        "customer"."yearly_equip_faiure" AS "yearly_equip_faiure",
        CAST("location"."city" AS TEXT) AS "city",
        CAST("location"."county" AS TEXT) AS "county",
        "location"."location_id" AS "location_id (location)",
        CAST("location"."state" AS TEXT) AS "state",
        "location"."zip" AS "zip",
        "population"."POPESTIMATE2022" AS "POPESTIMATE2022",
        CAST("population"."STATE_ABB" AS TEXT) AS "STATE_ABB"
    FROM "public"."customer" "customer"
    INNER JOIN "public"."location"
    ON "location"."location_id" = "customer"."location_id"
    INNER JOIN "public"."population"
    ON "population"."STATE_ABB" = "location"."state"
) "Custom SQL Query2"
LEFT JOIN (
    SELECT CAST("location"."city" AS TEXT) AS "city",

```

```

        CAST("location"."county" AS TEXT) AS "county",
        "location"."location_id" AS "location_id (location)",
        CAST("location"."state" AS TEXT) AS "state",
        "location"."zip" AS "zip"
    FROM "public"."location" "location"
) "Custom SQL Query1" ON ("Custom SQL Query2"."location_id" = "Custom
SQL Query1"."location_id (location)")
GROUP BY 1
HAVING (NOT ((CASE WHEN COUNT(CAST("Custom SQL Query2"."churn" AS
TEXT)) = 0 THEN NULL ELSE CAST(SUM((CASE WHEN (CAST("Custom SQL
Query2"."churn" AS TEXT) = 'Yes') THEN 1 ELSE NULL END)) AS DOUBLE
PRECISION) / COUNT(CAST("Custom SQL Query2"."churn" AS TEXT)) END) IS
NULL))
) "t1" ON ("t0"."state" IS NOT DISTINCT FROM "t1"."state")

```

```

SELECT COUNT(CAST("Custom SQL Query2"."churn" AS TEXT)) AS
"TEMP(Calculation_242912952012042241)(3605431932)(0)",
SUM((CASE WHEN (CAST("Custom SQL Query2"."churn" AS TEXT) = 'Yes') THEN
1 ELSE NULL END)) AS "TEMP(Calculation_242912952012042241)(670882048)
(0)",
CAST("Custom SQL Query1"."state" AS TEXT) AS "state"
FROM (
    SELECT "customer"."age" AS "age",
        "customer"."bandwidth_gp_year" AS "bandwidth_gp_year",
        "customer"."children" AS "children",
        CAST("customer"."churn" AS TEXT) AS "churn",
        "customer"."contacts" AS "contacts",
        "customer"."contract_id" AS "contract_id",
        CAST("customer"."customer_id" AS TEXT) AS "customer_id",
        "customer"."email" AS "email",
        CAST("customer"."gender" AS TEXT) AS "gender",
        "customer"."income" AS "income",
        "customer"."job_id" AS "job_id",
        "customer"."lat" AS "lat",
        "customer"."lng" AS "lng",
        "customer"."location_id" AS "location_id",
        CAST("customer"."marital" AS TEXT) AS "marital",
        "customer"."monthly_charge" AS "monthly_charge",
        "customer"."outage_sec_week" AS "outage_sec_week",
        "customer"."payment_id" AS "payment_id",
        "customer"."population" AS "population",
        CAST("customer"."port_modem" AS TEXT) AS "port_modem",
        CAST("customer"."tablet" AS TEXT) AS "tablet",
        CAST("customer"."techie" AS TEXT) AS "techie",
        "customer"."tenure" AS "tenure",
        "customer"."yearly_equip_faiure" AS "yearly_equip_faiure",
        CAST("location"."city" AS TEXT) AS "city",
        CAST("location"."county" AS TEXT) AS "county",
        "location"."location_id" AS "location_id (location)",
        CAST("location"."state" AS TEXT) AS "state",
        "location"."zip" AS "zip",

```

```

"population"."POPESTIMATE2022" AS "POPESTIMATE2022",
CAST("population"."STATE_ABB" AS TEXT) AS "STATE_ABB"
FROM "public"."customer" "customer"
INNER JOIN "public"."location"
ON "location"."location_id" = "customer"."location_id"
INNER JOIN "public"."population"
ON "population"."STATE_ABB" = "location"."state"
) "Custom SQL Query2"
LEFT JOIN (
SELECT CAST("location"."city" AS TEXT) AS "city",
CAST("location"."county" AS TEXT) AS "county",
"location"."location_id" AS "location_id (location)",
CAST("location"."state" AS TEXT) AS "state",
"location"."zip" AS "zip"
FROM "public"."location" "location"
) "Custom SQL Query1" ON ("Custom SQL Query2"."location_id" = "Custom SQL
Query1"."location_id (location)")
GROUP BY 3
HAVING (NOT ((CASE WHEN COUNT(CAST("Custom SQL Query2"."churn" AS TEXT))
= 0 THEN NULL ELSE CAST(SUM((CASE WHEN (CAST("Custom SQL Query2"."churn"
AS TEXT) = 'Yes') THEN 1 ELSE NULL END)) AS DOUBLE PRECISION) /
COUNT(CAST("Custom SQL Query2"."churn" AS TEXT)) END) IS NULL))

```

Creating the Dashboard

Click the "new dashboard" button along the bottom toolbar. You may drag each worksheet from the sheets panel over to the right side to fill the space in the dashboard.

I made the state filter for the "Churn % by State" worksheet a floating object by opening the settings dropdown for the filter and choosing "floating", then dragging the filter where I wanted it to appear in the dashboard.

6. Results of Data Analysis

Based on the analysis of the geographical data when combined with churn data, there are some states in the country that represent missed opportunities for growth and retention. States with larger populations such as California, Texas and Florida have a low subscriber ratios when compared to the total number of residents in those states. It would be beneficial to explore marketing campaigns and onboarding initiatives in those states in order to better capitalize on the opportunities they present.

7. Limitations of Data Analysis

While the data available gives us ample opportunity to hypothesize where growth opportunities might exist, it is limited in scope and missing some additional information that would be beneficial. These might include known competitors, their subscriber base in each location, and popular products to name just a few things.

With regard to the tools used, Tableau Desktop makes visualizing data easy, but it does have some drawbacks. Data must be cleaned and prepared prior to import, as the options available within Tableau are slim. The same can be said for Tableau's data modeling capabilities (To, 2021).

SQL is a versatile and well-documented data language, and the only limitations I see with using it would be strictly due to the user's lack of familiarity with it.

8. Web Sources

<https://blog.openbridge.com/7-steps-to-export-sql-statements-from-tableau-7e51a2fd4277>

<https://stackoverflow.com/questions/6308594/how-can-i-copy-data-from-one-column-to-another-in-the-same-table>

9. Works Cited

To, K. (2021, September 3). *Tableau Limitations and Top 16 Alternatives*. Holistics Blog.

<https://www.holistics.io/blog/tableau-limitations-and-top-alternatives/>