D211 Advanced Data Acquisition

SLM2 Data Analysis Performance Assessment

Hillary Osei (Student ID #011039266)

Western Governors University, College of Information Technology

Program Mentor: Dan Estes

June 16, 2025

Table of Contents

Part I: Data Dashboards

Section A1) Datasets and Dashboard File

Dashboard is titled “D211-Dashboard.twbx”

Internal dataset(s) are titled: “location.csv” and “customer.csv”
External dataset is titled: “acs2017_census_tract_data.csv”

Section A2) Dashboard Installation

The dashboard can be viewed by following these instructions:

pgAdmin Instructions

1. Download the “acs2017_census_tract_data.csv” (external dataset) and the “SQL-codes.txt” containing all the necessary SQL queries.
2. Start pgAdmin 4, click the down arrow next to Servers, then double click “churn” to start the database.
3. Right click churn and select the “Query Tool” to start the SQL query.
4. Create the “census_data" table with the first code in the “SQL-codes.txt” file (starts with the CREATE TABLE… statement and ends with OWNER to postgres) and click the play button to run it.
5. On the directory on the left, double click on “Schemas,” then scroll down and double click on “Tables” to show all the tables in the database.
6. Right click on “census_data,” then choose the “Import/Export data” option.
7. Select the Import option, set filename to “acs2017_census_tract_data.csv”, set format to csv, set encoding to UTF8, set the header option to Yes, and set the delimiter to a comma.
8. Select OK.
9. Click the “Query Tool” button and copy all the remaining code (minus the first one for creating the census_data_ table from SQL-code.txt file) and click the play button to run the queries (note: copy everything starting with the ALTER TABLE census_data…DROP COLUMN to the UPDATE census_data SET “State” = CASE” code).

Tableau Instructions

1. Download the “D211-Dashboard.twbx” file
2. Click on the Tableau 2021.4 application located on the Desktop to start
3. On the Connect screen, scroll to the “To a Server” section, select More, type in PostgreSQL and select it.
4. A window appears asking for the credentials for PostgreSQL. The Server name is “localhost.” The Port is “5432.” The database is “churn.” The authentication should be

set to “Username and Password.” The username is “postgres” and the password is “Passw0rd!.” Then, click Sign in.

5. In the churn connection screen, there should be a left sidebar that lists all the tables in the database. Drag the “location” table first to the empty space on the right side of the screen that says “Drag tables here," then drag the “census_data” table next so they are side by side.
6. There should be a line that connects location and “census_data” together. Click the line, and at the bottom there should be a section to set relationships. Set State from the “location” table equal to (=) the State1 column from the “census_data” column.
7. Return to the boxes connecting the two tables together, double click the location box.
8. In the join screen, drag the “census_data” table next to the “location” box. Then, drag the “customer” table below the “census_data” box.
9. Set the join type for “location” and “census_data” to left join and make sure the clause is set to State = State.
10. Set the join type for “location” and “customer” to inner join and make sure the clause is set to Location Id = Location Id.
11. Import the workbook by going to File > Open. Go to the Downloads folder and click on D211-Dashboard.twbx.
12. A login screen will open. Enter the password “Passw0rd!” Again and click sign-in.

Section A3) Dashboard Navigation

The first sheet in the file is Churn Rate by State, followed by CHurn Rate vs Unemployment, then Churn Rate vs Employment, Churn Rate & Income, and Churn Rate & Gender. The last tab is a dashboard containing all of these worksheets in one. When navigating the dashboard, the churn rate by state map is on the top left of the page, and the Churn Rate & Income bar chart is on the top right. Below these are the Churn Rate vs Employment Rate, Churn rate vs Unemployment Rate charts, and at the bottom right is the Churn vs Gender bar chart. In the map chart, orange represents high churn and blue represents low churn, and in the Churn vs Gender bar chart, blue represents male customers and pink represents female customers.

Section A4) SQL Code

Internal dataset table:

```
- Table: public.customer

-- DROP TABLE public.customer;

CREATE TABLE public.customer
(
   customer_id text COLLATE pg_catalog."default" NOT NULL,
   lat numeric,
   lng numeric,
   population integer,
   children integer,
   age integer,
   income numeric,
   marital text COLLATE pg_catalog."default",
   churn text COLLATE pg_catalog."default",
   gender text COLLATE pg_catalog."default",
   tenure numeric,
   monthly_charge numeric,
   bandwidth_gp_year numeric,
   outage_sec_week numeric,
   email integer,
   contacts integer,
   yearly_equip_faiure integer,
   techie text COLLATE pg_catalog."default",
   port_modem text COLLATE pg_catalog."default",
   tablet text COLLATE pg_catalog."default",
   job_id integer,
   payment_id integer,
   contract_id integer,
   location_id integer,
   CONSTRAINT customer_pkey PRIMARY KEY (customer_id),
   CONSTRAINT customer_contract_id_fkey FOREIGN KEY (contract_id)
      REFERENCES public.contract (contract_id) MATCH SIMPLE
      ON UPDATE NO ACTION
      ON DELETE NO ACTION
      NOT VALID,
   CONSTRAINT customer_job_id_fkey FOREIGN KEY (job_id)
      REFERENCES public.job (job_id) MATCH SIMPLE
      ON UPDATE NO ACTION
      ON DELETE NO ACTION
      NOT VALID,
   CONSTRAINT customer_location_id_fkey FOREIGN KEY (location_id)
      REFERENCES public.location (location_id) MATCH SIMPLE
      ON UPDATE NO ACTION
      ON DELETE NO ACTION
      NOT VALID,
   CONSTRAINT customer_payment_id_fkey FOREIGN KEY (payment_id)
```

```
        REFERENCES public.payment (payment_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
)

TABLESPACE pg_default;

ALTER TABLE public.customer
    OWNER to postgres;
```

Create external dataset table:

```
CREATE TABLE public.census_data
(
    "TractId" bigint NOT NULL,
    "State" character varying(100) COLLATE pg_catalog."default" NOT NULL,
    "County" character varying(100) COLLATE pg_catalog."default" NOT NULL,
    "TotalPop" integer,
    "Men" integer,
    "Women" integer,
    "Hispanic" double precision,
    "White" double precision,
    "Black" double precision,
    "Native" double precision,
    "Asian" double precision,
    "Pacific" double precision,
    "VotingAgeCitizen" integer,
    "Income" double precision,
    "IncomeErr" double precision,
    "IncomePerCap" double precision,
    "IncomePerCapErr" double precision,
    "Poverty" double precision,
    "ChildPoverty" double precision,
    "Professional" double precision,
    "Service" double precision,
    "Office" double precision,
    "Construction" double precision,
    "Production" double precision,
    "Drive" double precision,
    "Carpool" double precision,
    "Transit" double precision,
    "Walk" double precision,
    "OtherTransp" double precision,
    "WorkAtHome" double precision,
    "MeanCommute" double precision,
    "Employed" integer,
    "PrivateWork" double precision,
    "PublicWork" double precision,
    "SelfEmployed" double precision,
    "FamilyWork" double precision,
    "Unemployment" double precision
)

TABLESPACE pg_default;

ALTER TABLE public.census_data
    OWNER to postgres;
```

Import table data code:

```
--command " "\\copy public.census_data (\"TractId\", \"State\", \"County\", \"TotalPop\",
\"Men\", \"Women\", \"Hispanic\", \"White\", \"Black\", \"Native\", \"Asian\", \"Pacific\",
\"VotingAgeCitizen\", \"Income\", \"IncomeErr\", \"IncomePerCap\", \"IncomePerCapErr\",
\"Poverty\", \"ChildPoverty\", \"Professional\", \"Service\", \"Office\", \"Construction\",
\"Production\", \"Drive\", \"Carpool\", \"Transit\", \"Walk\", \"OtherTransp\", \"WorkAtHome\",
\"MeanCommute\", \"Employed\", \"PrivateWork\", \"PublicWork\", \"SelfEmployed\",
\"FamilyWork\", \"Unemployment\") FROM 'C:/Users/LabUser/Desktop/ACS201~1.CSV'
DELIMITER ',' CSV HEADER ENCODING 'UTF8' QUOTE '\"' ESCAPE '''';""--command "
"\\copy public.census_data (\"TractId\", \"State\", \"County\", \"TotalPop\", \"Men\", \"Women\",
\"Hispanic\", \"White\", \"Black\", \"Native\", \"Asian\", \"Pacific\", \"VotingAgeCitizen\",
\"Income\", \"IncomeErr\", \"IncomePerCap\", \"IncomePerCapErr\", \"Poverty\",
\"ChildPoverty\", \"Professional\", \"Service\", \"Office\", \"Construction\", \"Production\",
\"Drive\", \"Carpool\", \"Transit\", \"Walk\", \"OtherTransp\", \"WorkAtHome\",
\"MeanCommute\", \"Employed\", \"PrivateWork\", \"PublicWork\", \"SelfEmployed\",
\"FamilyWork\", \"Unemployment\") FROM 'C:/Users/LabUser/Desktop/ACS201~1.CSV'
DELIMITER ',' CSV HEADER ENCODING 'UTF8' QUOTE '\"' ESCAPE '''';""
```

***Drop irrelevant columns*:**

```
ALTER TABLE census_data
DROP COLUMN "TractId",
DROP COLUMN "VotingAgeCitizen",
DROP COLUMN "IncomeErr",
DROP COLUMN "IncomePerCap",
DROP COLUMN "IncomePerCapErr",
DROP COLUMN "Men",
DROP COLUMN "Women",
DROP COLUMN "Hispanic",
DROP COLUMN "White",
DROP COLUMN "Black",
DROP COLUMN "Native",
DROP COLUMN "Asian",
DROP COLUMN "Pacific";
```

Remove suffixes from County column:

```
UPDATE census_data
SET "County" = TRIM(REPLACE(REPLACE("County", ' County', ''), ' Municipio', ''));
```

FInd nulls in columns:

```
SELECT
  COUNT(*) FILTER (WHERE "State" IS NULL) AS null_state,
  COUNT(*) FILTER (WHERE "County" IS NULL) AS null_county,
  COUNT(*) FILTER (WHERE "TotalPop" IS NULL) AS null_totalpop,
  COUNT(*) FILTER (WHERE "Income" IS NULL) AS null_income,
  COUNT(*) FILTER (WHERE "Poverty" IS NULL) AS null_poverty,
  COUNT(*) FILTER (WHERE "ChildPoverty" IS NULL) AS null_childpoverty,
  COUNT(*) FILTER (WHERE "Professional" IS NULL) AS null_professional,
  COUNT(*) FILTER (WHERE "Service" IS NULL) AS null_service,
  COUNT(*) FILTER (WHERE "Office" IS NULL) AS null_office,
  COUNT(*) FILTER (WHERE "Construction" IS NULL) AS null_construction,
  COUNT(*) FILTER (WHERE "Production" IS NULL) AS null_production,
  COUNT(*) FILTER (WHERE "Drive" IS NULL) AS null_drive,
  COUNT(*) FILTER (WHERE "Carpool" IS NULL) AS null_carpool,
  COUNT(*) FILTER (WHERE "Transit" IS NULL) AS null_transit,
  COUNT(*) FILTER (WHERE "Walk" IS NULL) AS null_walk,
  COUNT(*) FILTER (WHERE "OtherTransp" IS NULL) AS null_othertransp,
  COUNT(*) FILTER (WHERE "WorkAtHome" IS NULL) AS null_workathome,
  COUNT(*) FILTER (WHERE "MeanCommute" IS NULL) AS null_meancommute,
  COUNT(*) FILTER (WHERE "Employed" IS NULL) AS null_employed,
  COUNT(*) FILTER (WHERE "PrivateWork" IS NULL) AS null_privatework,
  COUNT(*) FILTER (WHERE "PublicWork" IS NULL) AS null_publicwork,
  COUNT(*) FILTER (WHERE "SelfEmployed" IS NULL) AS null_selfemployed,
  COUNT(*) FILTER (WHERE "FamilyWork" IS NULL) AS null_familywork,
  COUNT(*) FILTER (WHERE "Unemployment" IS NULL) AS null_unemployment
FROM census_data;
```

FIll certain null columns with mean:

```
-- Poverty
UPDATE census_data
SET "Poverty" = (SELECT AVG("Poverty") FROM census_data WHERE "Poverty" IS NOT
NULL)
WHERE "Poverty" IS NULL;

-- ChildPoverty
UPDATE census_data
SET "ChildPoverty" = (SELECT AVG("ChildPoverty") FROM census_data WHERE
"ChildPoverty" IS NOT NULL)
WHERE "ChildPoverty" IS NULL;

-- Professional
UPDATE census_data
SET "Professional" = (SELECT AVG("Professional") FROM census_data WHERE
"Professional" IS NOT NULL)
WHERE "Professional" IS NULL;

-- Service
UPDATE census_data
SET "Service" = (SELECT AVG("Service") FROM census_data WHERE "Service" IS NOT
NULL)
WHERE "Service" IS NULL;

-- Office
UPDATE census_data
SET "Office" = (SELECT AVG("Office") FROM census_data WHERE "Office" IS NOT
NULL)
WHERE "Office" IS NULL;

-- Construction
UPDATE census_data
```

```
SET "Construction" = (SELECT AVG("Construction") FROM census_data WHERE
"Construction" IS NOT NULL)
WHERE "Construction" IS NULL;

-- Production
UPDATE census_data
SET "Production" = (SELECT AVG("Production") FROM census_data WHERE "Production"
IS NOT NULL)
WHERE "Production" IS NULL;

-- Drive
UPDATE census_data
SET "Drive" = (SELECT AVG("Drive") FROM census_data WHERE "Drive" IS NOT NULL)
WHERE "Drive" IS NULL;

-- Carpool
UPDATE census_data
SET "Carpool" = (SELECT AVG("Carpool") FROM census_data WHERE "Carpool" IS NOT
NULL)
WHERE "Carpool" IS NULL;

-- Transit
UPDATE census_data
SET "Transit" = (SELECT AVG("Transit") FROM census_data WHERE "Transit" IS NOT
NULL)
WHERE "Transit" IS NULL;

-- Walk
UPDATE census_data
SET "Walk" = (SELECT AVG("Walk") FROM census_data WHERE "Walk" IS NOT NULL)
WHERE "Walk" IS NULL;

-- OtherTransp
UPDATE census_data
SET "OtherTransp" = (SELECT AVG("OtherTransp") FROM census_data WHERE
"OtherTransp" IS NOT NULL)
WHERE "OtherTransp" IS NULL;

-- WorkAtHome
UPDATE census_data
```

```
SET "WorkAtHome" = (SELECT AVG("WorkAtHome") FROM census_data WHERE
"WorkAtHome" IS NOT NULL)
WHERE "WorkAtHome" IS NULL;

-- Employed
UPDATE census_data
SET "Employed" = (SELECT AVG("Employed") FROM census_data WHERE "Employed" IS
NOT NULL)
WHERE "Employed" IS NULL;

-- PrivateWork
UPDATE census_data
SET "PrivateWork" = (SELECT AVG("PrivateWork") FROM census_data WHERE
"PrivateWork" IS NOT NULL)
WHERE "PrivateWork" IS NULL;

-- PublicWork
UPDATE census_data
SET "PublicWork" = (SELECT AVG("PublicWork") FROM census_data WHERE "PublicWork"
IS NOT NULL)
WHERE "PublicWork" IS NULL;

-- SelfEmployed
UPDATE census_data
SET "SelfEmployed" = (SELECT AVG("SelfEmployed") FROM census_data WHERE
"SelfEmployed" IS NOT NULL)
WHERE "SelfEmployed" IS NULL;

-- FamilyWork
UPDATE census_data
SET "FamilyWork" = (SELECT AVG("FamilyWork") FROM census_data WHERE
"FamilyWork" IS NOT NULL)
WHERE "FamilyWork" IS NULL;
```

Fill income, unemployment, meancommute nulls with median:

```
UPDATE census_data
SET "Income" = sub.median_value
FROM (
   SELECT PERCENTILE_CONT(0.5) WITHIN GROUP (ORDER BY "Income") AS
median_value
   FROM census_data
   WHERE "Income" IS NOT NULL
) AS sub
WHERE "Income" IS NULL;

UPDATE census_data
SET "MeanCommute" = sub.median_value
FROM (
   SELECT PERCENTILE_CONT(0.5) WITHIN GROUP (ORDER BY "MeanCommute") AS
median_value
   FROM census_data
   WHERE "MeanCommute" IS NOT NULL
) AS sub
WHERE "MeanCommute" IS NULL;

UPDATE census_data
SET "Unemployment" = sub.median_value
FROM (
   SELECT PERCENTILE_CONT(0.5) WITHIN GROUP (ORDER BY "Unemployment") AS
median_value
   FROM census_data
   WHERE "Unemployment" IS NOT NULL
) AS sub
WHERE "Unemployment" IS NULL;
```

***Change states to abbreviations*:**

```
UPDATE census_data
SET "State" = CASE
   WHEN "State" = 'Alabama' THEN 'AL'
   WHEN "State" = 'Alaska' THEN 'AK'
   WHEN "State" = 'Arizona' THEN 'AZ'
   WHEN "State" = 'Arkansas' THEN 'AR'
   WHEN "State" = 'California' THEN 'CA'
   WHEN "State" = 'Colorado' THEN 'CO'
   WHEN "State" = 'Connecticut' THEN 'CT'
   WHEN "State" = 'Delaware' THEN 'DE'
   WHEN "State" = 'Florida' THEN 'FL'
   WHEN "State" = 'Georgia' THEN 'GA'
   WHEN "State" = 'Hawaii' THEN 'HI'
   WHEN "State" = 'Idaho' THEN 'ID'
   WHEN "State" = 'Illinois' THEN 'IL'
   WHEN "State" = 'Indiana' THEN 'IN'
   WHEN "State" = 'Iowa' THEN 'IA'
   WHEN "State" = 'Kansas' THEN 'KS'
   WHEN "State" = 'Kentucky' THEN 'KY'
   WHEN "State" = 'Louisiana' THEN 'LA'
   WHEN "State" = 'Maine' THEN 'ME'
   WHEN "State" = 'Maryland' THEN 'MD'
   WHEN "State" = 'Massachusetts' THEN 'MA'
   WHEN "State" = 'Michigan' THEN 'MI'
   WHEN "State" = 'Minnesota' THEN 'MN'
   WHEN "State" = 'Mississippi' THEN 'MS'
   WHEN "State" = 'Missouri' THEN 'MO'
   WHEN "State" = 'Montana' THEN 'MT'
   WHEN "State" = 'Nebraska' THEN 'NE'
   WHEN "State" = 'Nevada' THEN 'NV'
   WHEN "State" = 'New Hampshire' THEN 'NH'
   WHEN "State" = 'New Jersey' THEN 'NJ'
   WHEN "State" = 'New Mexico' THEN 'NM'
   WHEN "State" = 'New York' THEN 'NY'
   WHEN "State" = 'North Carolina' THEN 'NC'
   WHEN "State" = 'North Dakota' THEN 'ND'
   WHEN "State" = 'Ohio' THEN 'OH'
   WHEN "State" = 'Oklahoma' THEN 'OK'
   WHEN "State" = 'Oregon' THEN 'OR'
   WHEN "State" = 'Pennsylvania' THEN 'PA'
   WHEN "State" = 'Rhode Island' THEN 'RI'
   WHEN "State" = 'South Carolina' THEN 'SC'
   WHEN "State" = 'South Dakota' THEN 'SD'
   WHEN "State" = 'Tennessee' THEN 'TN'
   WHEN "State" = 'Texas' THEN 'TX'
   WHEN "State" = 'Utah' THEN 'UT'
   WHEN "State" = 'Vermont' THEN 'VT'
   WHEN "State" = 'Virginia' THEN 'VA'
   WHEN "State" = 'Washington' THEN 'WA'
   WHEN "State" = 'West Virginia' THEN 'WV'
   WHEN "State" = 'Wisconsin' THEN 'WI'
   WHEN "State" = 'Wyoming' THEN 'WY'
```

```
    WHEN "State" = 'Puerto Rico' THEN 'PR'
    WHEN "State" = 'District of Columbia' THEN 'DC'
END;
```

***Join from Tableau*:**

```
SELECT "census_data1"."Carpool" AS "Carpool (census_data1)",
  "census_data1"."ChildPoverty" AS "ChildPoverty (census_data1)",
  "census_data1"."Construction" AS "Construction (census_data1)",
  "census_data1"."County" AS "County (census_data1)",
  "census_data1"."Drive" AS "Drive (census_data1)",
  "census_data1"."Employed" AS "Employed (census_data1)",
  "census_data1"."FamilyWork" AS "FamilyWork (census_data1)",
  "census_data1"."Income" AS "Income (census_data1)",
  "census_data1"."MeanCommute" AS "MeanCommute (census_data1)",
  "census_data1"."Office" AS "Office (census_data1)",
  "census_data1"."OtherTransp" AS "OtherTransp (census_data1)",
  "census_data1"."Poverty" AS "Poverty (census_data1)",
  "census_data1"."PrivateWork" AS "PrivateWork (census_data1)",
  "census_data1"."Production" AS "Production (census_data1)",
  "census_data1"."Professional" AS "Professional (census_data1)",
  "census_data1"."PublicWork" AS "PublicWork (census_data1)",
  "census_data1"."SelfEmployed" AS "SelfEmployed (census_data1)",
  "census_data1"."Service" AS "Service (census_data1)",
  "census_data1"."State" AS "State (census_data1)",
  "census_data1"."TotalPop" AS "TotalPop (census_data1)",
  "census_data1"."Transit" AS "Transit (census_data1)",
  "census_data1"."Unemployment" AS "Unemployment (census_data1)",
  "census_data1"."Walk" AS "Walk (census_data1)",
  "census_data1"."WorkAtHome" AS "WorkAtHome (census_data1)",
  "customer"."age" AS "age",
  "customer"."bandwidth_gp_year" AS "bandwidth_gp_year",
  "customer"."children" AS "children",
  CAST("customer"."churn" AS TEXT) AS "churn",
  CAST("location"."city" AS TEXT) AS "city",
  "customer"."contacts" AS "contacts",
  "customer"."contract_id" AS "contract_id",
  CAST("location"."county" AS TEXT) AS "county",
  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 (customer)",
  "location"."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("location"."state" AS TEXT) AS "state",
  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",
  "location"."zip" AS "zip"
FROM "public"."location" "location"
  LEFT JOIN "public"."census_data" "census_data1" ON (CAST("location"."state" AS TEXT) =
"census_data1"."State")
  INNER JOIN "public"."customer" "customer" ON ("location"."location_id" =
"customer"."location_id")
```

Part II: Demonstration

Section B) Panopto Presentation

Link to the Panopto presentation is here:
https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=6334485a-1778-450c-91b0-b30
0013c10b3

Part III: Report

Section C1) Dashboard Alignment

The purpose of the dashboard is to help telecommunications stakeholders better understand the patterns and contributors to customers churning within the last month. Because churn is a key performance indicator in the telecommunications industry, the dashboard presents visual insights that explore the relationships between churn and socioeconomic factors such as income, unemployment rates, and gender. These insights allow stakeholders to identify areas for customer retention strategies, such as outreach in certain states or demographic groups with high churn rates.

Section C2) Business Intelligence Tool

Tableau was selected because it is a powerful tool for building interactive dashboards and visualizing trends across datasets to better understand customer churn. Features such as calculated fields, drag-and-drop functionality, and various chart types make it easier to find patterns, compare variables, and explain data visually (Coursera Staff, 2025). Tableau also seamlessly connects to PostgreSQL, which was used for storing, preparing, and cleaning the data.

PostgreSQL was selected as the database tool because it works well with structured data such as the external U.S. Census dataset and also supports SQL, a language used for querying and transforming data. SQL's simple syntax makes it easier to write commands for cleaning and joining data. Tables from the external and internal datasets can be managed in PostgreSQL, keeping the data organized and ready to use in Tableau.

Section C3) Data Cleaning

The external dataset was cleaned and prepared using PostgreSQL. A table "census_data" was created first, with column names and data types defined to match the census CSV file. After the table was created, the data was imported in pgAdmin, with the file configured to recognize headers and delimiters.

After importing, several columns were removed from the table using ALTER TABLE DROP COLUMN to eliminate fields that were not relevant to the analysis. The "County" column was standardized by removing suffixes like "County" and "Municipio" using the REPLACE and TRIM functions to make the data consistent for analysis.

Null values were detected in the remaining columns using COUNT(*) FILTER (WHERE…) syntax. Columns with missing values were handled based on the nature of the data. Median values were used to fill nulls in columns like Income, MeanCommute, and Unemployment since values are more likely to contain outliers that could skew the data analysis. The average was used to fill columns such as Poverty, ChildPoverty, and job sector percentages since they generally represent ratios and are less likely to be heavily skewed

Full state names were replaced with standard two-letter state abbreviations using a CASE statement. This data transformation ensures that the external data matches the internal dataset's State column format for joining the datasets later in Tableau.

Section C4) Dashboard Creation

Churn Rate by State (Map)

1. Drag State to Rows or directly into the view to activate the map.
2. Create a calculated field Churn Rate: SUM(IF [Churn] = 'Yes' THEN 1 ELSE 0 END) / COUNT([Customer Id]) * 100
3. Drag the calculated field Churn Rate to Color.
4. Set mark type to Map (automatic for geographic fields).
5. Customize the color gradient to orange-blue and then select the reversed option so the higher churn rates are orange and lower churn rates are blue.
6. Drag State to Detail

Churn Rate & Unemployment:

1. Create Unemployment to Columns
2. Drag Churn Rate to Rows.
3. Drag Churn Rate calculated field to colors
4. Drag State to Label
5. Drag State to Detail
6. Add trend line to see correlation

Churn Rate & Employment:

7. Create Employment Rate calculated field: (AVG([Employed]) / AVG([TotalPop])) * 100
8. Drag Employment Rate to Columns.
9. Drag Churn Rate to Rows.
10. Drag Churn Rate calculated field to colors
11. Drag State to Label
12. Drag State to Detail
13. Add trend line to see correlation

Churn Rate vs Income:

1. Drag State to Columns.
2. Drag Income to Rows.
3. Set aggregation for Income to Average.
4. Sort the bars descending by average income (click sort icon or right-click axis).
5. Drag Churn Rate to colors, set it to blue

Churn Rate by Gender:

1. Drag State to Columns.
2. Drag Churn Rate to Rows.
3. Drag Gender to Color to split bars by gender. Click color palette and select pink for women and blue for men
4. Drag Gender to Filters, selecting only Female and Male

Section C5) Data Analysis Results

The analysis revealed important trends related to customer churn. The state-level churn rate visualization identified areas with the highest and lowest churn, helping stakeholders prioritize regions that may need intervention. The highest churn rates were found in Delaware (38.10%), Connecticut (38.03%), Puerto Rico (35.4%), and Hawaii (34.27%) while Mississippi (15.87%), New Hampshire (17.65%), Louisiana (18.40%), and Vermont and D.C. (21.43%) had the lowest churn. This insight helps companies focus customer retention strategies in high risk areas.

There was also a clear relationship between churn and unemployment. States with higher churn also had higher unemployment, suggesting that financial instability may contribute to customers cancelling their services. For example, Puerto Rico had both the highest unemployment rate (18.20%) and one of the highest churn rates. In contrast, North Dakota, with the lowest unemployment rate (2.8%) had a much lower churn rate (26.29%). Stakeholders can use this information to tailor retention plans, such as offering flexible plans or payment assistance in economically disadvantageous areas.

The employment rate chart further supports this, showing that churn rates tend to drop as employment increases. For example, New Hampshire had the highest average employment rate at 53.57% while Puerto Rico again had the lowest average unemployment rate at 24%.

The relationship between churn and average income also revealed useful patterns. D.C. had the highest average income at $82,617 and a relatively low churn rate (21.43%) while Puerto Rico with the lowest average income at $23,664. This suggests that customers in lower-income regions may be more likely to leave due to affordability. Stakeholders can address this by implementing discounts or loyalty incentives for customers in lower-income areas.

The gender-based churn visualization showed that the highest churn rate among men were in Delaware and Rhode Island at 42.86% followed by Puerto Rico at 40%. In states like Hawaii and Connecticut, more women churned than men, Connecticut at 40.54% and Hawaii at 40%. Understanding gender-based churn behavior can help marketing teams customize messaging towards customers and outreach to improve retention.

Section C6) Analysis Limitations

Limitations of this analysis include that the external census data is from 2017, which will likely not reflect current conditions. Some states or certain areas might have changed over time in terms of income, unemployment, or population, so the results may not be fully up to date. Additionally, the analysis focused on finding patterns, not proving cause and effect. For instance, high unemployment rates and high churn rates may be related, but the data doesn't prove that one causes the other.

Section D) Web Sources

US Census Demographic Data. (n.d.). Kaggle. Retrieved June 16, 2025, from https://www.kaggle.com/datasets/muonneutrino/us-census-demographic-data

Section E) Sources

Coursera Staff. (2025, May 20). *What Is Tableau? Features, Use Cases, and More*. Coursera. Retrieved June 26, 2025, from https://www.coursera.org/articles/tableau