D211 Advanced Data Acquisition

By: Kevin Sandoval

**A1.**

Both datasets are provided in the submission.

**A2.**

As I’m not sure what the evaluators have access to when trying to view the dashboard, I will try outline steps necessary to viewing the dashboard in the labs on demand environment.

PATH 1

1. If the user is in the labs on demand environment while checking my submission, continue in path 1. If the user is on their own machine, continue to path 2.
2. The user must first download the .twbx file provided in the submission
3. The user now must enter the labs on demand environment
4. Open the Tableau program on the desktop. Use the regular Tableau and not Tableau Prep.
5. Click the file dropdown and then click on open. Navigate to the download section of the file manager and the .twbx should be there.
6. Open the .twbx file and everything should load
7. The user may be prompted with login requirements for the localhost. The requirements are here:
   1. Server: localhost
   2. Port: 5432
   3. Database: churn
   4. Username: postgres
   5. Password: Passw0rd!

PATH 2

1. Download the .twbx file from the submission
2. Open an email and email the .twbx file found in the downloads folder to yourself
3. Open the labs on demand environment
4. Open Microsoft edge and log into the email that you used to send the .twbx file to yourself
5. Download the file from the email.
6. Open the Tableau program on the desktop. Use the regular Tableau and not Tableau Prep.
7. Click the file dropdown and then click on open. Navigate to the download section of the file manager and the .twbx should be there.
8. Open the .twbx file and everything should load
9. The user may be prompted with login requirements for the localhost. The requirements are here:
   1. Server: localhost
   2. Port: 5432
   3. Database: churn
   4. Username: postgres
   5. Password: Passw0rd!

I know that the second Path is not ideal but I believe it is the easiest way to access the .twbx file as well as the localhost that stores the data being used.

**A3.**

The dashboard is relatively simple and straightforward. The user can click on any of the tabs at the bottom of the screen to see the worksheets that were created. If the user scrolls all the way to the right, using the arrows in the bottom right, the user can then select the story. After selecting the story, the user can click on either the boxes below the title, or the arrows on either side of the boxes to change the pages in the dashboard. There are 4 pages which include the Intro, Demographic information, Actionable Insights, and the Conclusion.

* **Intro:** There are no interactive elements in the introduction for the user to use. Progressing to the next page can be accessed by the boxes above or the right arrow.
* **Demographics:** There are two filters on this page. The user can select or deselect and then click apply on the gender filter to filter by a gender. The age filter is a slider range. The user can change the data using this filter by sliding the filter to create a range. For example, by deselecting “Female”, clicking apply, and moving the slider to have 30 on the low end and 45 on the high end, we can see the data for Males aged 30-45. The user can also hover over an individual pie chart slice, or bar in the bar graph. If the user selects a bar in the bar graph, they may see an option box pop up to “Keep-Only” or “Exclude.” This is another way of filtering. If the user edits the data and cannot figure out how to revert to the original, there is a small revert button above the “Demographic” box above the graphs.
* **Actionable Insights:** There are two filters on this page. The user can select or deselect and then click apply on the gender filter to filter by a gender. The age filter is a slider range. The user can change the data using this filter by sliding the filter to create a range. For example, by deselecting “Female”, clicking apply, and moving the slider to have 30 on the low end and 45 on the high end, we can see the data for Males aged 30-45. The user can also hover over one of the rectangle data points in the Monthly Charge graphs, or a rectangle data point in the Contract Churn charts to see more information. If the user selects a rectangle data point, they may see an option box pop up to “Keep-Only” or “Exclude.” This is another way of filtering. While not interactable, there is a gradient scale to the side that shows how the color of the graph changes with the churn rate, or monthly charge. If the user edits the data and cannot figure out how to revert to the original, there is a small revert button above the “Actionable Insights” box above the graphs.
* **Closing:** There are no interactive elements in the closing for the user to use. Progressing to the previous page can be accessed by the boxes above or the left arrow.

**A4.**

SQL Code for creating the table:

ALTER TABLE public.location ALTER COLUMN zip TYPE varchar(10);

DROP TABLE public.california;

CREATE TABLE public.california (

      zip varchar (10),

      age varchar (5),

      gender varchar(10),

      churn varchar (5),

      contract varchar (20),

      paperlessbilling varchar(20),

      tenure varchar(10),

      monthlycharge varchar(20),

      bandwidth\_gb\_month varchar(20),

      location\_id int NOT NULL,

      CONSTRAINT california\_pkey PRIMARY KEY (location\_id),

      CONSTRAINT california\_location\_id\_fkey FOREIGN KEY (location\_id)

            REFERENCES public.location (location\_id) MATCH SIMPLE

            ON UPDATE CASCADE

            ON DELETE RESTRICT

);

--command " "\\copy public.california (zip, age, gender, churn, contract, paperlessbilling, tenure, monthlycharge, bandwidth\_gb\_month, location\_id) FROM 'C:/Users/LabUser/DESKTO~1/CHURN\_~1.CSV' DELIMITER ',' CSV QUOTE '\"' ESCAPE '''';""

SQL Code for creating the joined data.

SELECT "Custom SQL Query"."age (customer)" AS "age (customer)",

  "Custom SQL Query"."age" AS "age",

  "Custom SQL Query"."bandwidth\_gb\_month" AS "bandwidth\_gb\_month",

  "Custom SQL Query"."bandwidth\_gp\_year" AS "bandwidth\_gp\_year",

  "Custom SQL Query"."children" AS "children",

  CAST("Custom SQL Query"."churn (customer)" AS TEXT) AS "churn (customer)",

  "Custom SQL Query"."churn" AS "churn",

  CAST("Custom SQL Query"."city" AS TEXT) AS "city",

  "Custom SQL Query"."contacts" AS "contacts",

  "Custom SQL Query"."contract" AS "contract",

  "contract"."contract\_id" AS "contract\_id (contract)",

  "Custom SQL Query"."contract\_id" AS "contract\_id",

  CAST("Custom SQL Query"."county" AS TEXT) AS "county",

  CAST("Custom SQL Query"."customer\_id" AS TEXT) AS "customer\_id",

  CAST("contract"."duration" AS TEXT) AS "duration",

  "Custom SQL Query"."email" AS "email",

  CAST("Custom SQL Query"."gender (customer)" AS TEXT) AS "gender (customer)",

  "Custom SQL Query"."gender" AS "gender",

  "Custom SQL Query"."income" AS "income",

  "Custom SQL Query"."job\_id" AS "job\_id",

  "Custom SQL Query"."lat" AS "lat",

  "Custom SQL Query"."lng" AS "lng",

  "Custom SQL Query"."location\_id (customer)" AS "location\_id (customer)",

  "Custom SQL Query"."location\_id" AS "location\_id",

  CAST("Custom SQL Query"."marital" AS TEXT) AS "marital",

  "Custom SQL Query"."monthly\_charge" AS "monthly\_charge",

  "Custom SQL Query"."monthlycharge" AS "monthlycharge",

  "Custom SQL Query"."outage\_sec\_week" AS "outage\_sec\_week",

  "Custom SQL Query"."paperlessbilling" AS "paperlessbilling",

  "Custom SQL Query"."payment\_id" AS "payment\_id",

  "Custom SQL Query"."population" AS "population",

  CAST("Custom SQL Query"."port\_modem" AS TEXT) AS "port\_modem",

  CAST("Custom SQL Query"."state" AS TEXT) AS "state",

  CAST("Custom SQL Query"."tablet" AS TEXT) AS "tablet",

  CAST("Custom SQL Query"."techie" AS TEXT) AS "techie",

  "Custom SQL Query"."tenure (customer)" AS "tenure (customer)",

  "Custom SQL Query"."tenure" AS "tenure",

  "Custom SQL Query"."yearly\_equip\_faiure" AS "yearly\_equip\_faiure",

  "Custom SQL Query"."zip (california)" AS "zip (california)",

  "Custom SQL Query"."zip" AS "zip"

FROM (

  SELECT DISTINCT ON ("california"."zip")

    "customer"."age" AS "age (customer)",

    "california"."age" AS "age",

    "california"."bandwidth\_gb\_month" AS "bandwidth\_gb\_month",

    "customer"."bandwidth\_gp\_year" AS "bandwidth\_gp\_year",

    "customer"."children" AS "children",

    CAST("customer"."churn" AS TEXT) AS "churn (customer)",

    "california"."churn" AS "churn",

    CAST("location"."city" AS TEXT) AS "city",

    "customer"."contacts" AS "contacts",

    "california"."contract" AS "contract",

    "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)",

    "california"."gender" 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",

    "california"."monthlycharge" AS "monthlycharge",

    "customer"."outage\_sec\_week" AS "outage\_sec\_week",

    "california"."paperlessbilling" AS "paperlessbilling",

    "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)",

    "california"."tenure" AS "tenure",

    "customer"."yearly\_equip\_faiure" AS "yearly\_equip\_faiure",

    "california"."zip" AS "zip (california)",

    "location"."zip" AS "zip"

  FROM "public"."location" "location"

    INNER JOIN "public"."customer" "customer" ON ("location"."location\_id" = "customer"."location\_id")

    INNER JOIN "public"."california" "california" ON ("location"."zip" = "california"."zip")

) "Custom SQL Query"

  INNER JOIN "public"."contract" "contract" ON ("Custom SQL Query"."contract\_id" = "contract"."contract\_id")

**B.**

The link to the Panopto video is provided in the submission.

**C1.**

The purpose of my dashboard was to analyze the demographic information and churn rates of the WGU Churn dataset and compare it with the California Q2 2022 Telecom dataset. I looked at the age and gender demographics specifically which fits in with what the EVP is looking for. I also looked at the Monthly Charge and Contract type for each customer in an age bin of 5 years, while also examining if they churned or not. The contract type and charge that a customer pays could drive the decisions of the customer which is what the SVP is looking for in this analysis. These metrics correctly showcase what the leaders are looking and can create actionable insights for the company.

**C2.**

I used Tableau for the creation of my dashboard. Tableau is a great tool to use for this type of analysis because it allows us to connect to data sources, and then use SQL code to manipulate the data to use how we see fit. It also allows us to create interesting and engaging visuals that helps showcase what we discovered to an audience that may not be technically inclined.

**C3.**

I pulled data from Kaggle to use for this analysis. I checked the data for missing values, duplicate data, and outliers before I decided to bring it into PGAdmin for use in the analysis. I did the same type of data cleaning and preparation as we did in D206. I used boxplots to check for the outliers and isnull() for the missing values. After I cleaned all the data of any inconsistencies or abnormalities, I then created a csv of the new dataset to use for the analysis.

**C4.**

In order to create the dashboard, I needed to bring in the data to Tableau at the beginning. To do this, I entered the Labs on Demand and emailed myself the external file used to compare the data. I retrieved the dataset and downloaded it. I then loaded into PGAdmin and created a table for the new dataset. After I created the table and linked it, I imported the data from the external dataset into the newly created table. Now that I had everything in one location, I loaded up Tableau and connected to the dataset. I performed a join on the tables to make accessing the data easier as well as to look at just the overlapping data points from WGU and the California dataset. Once I had all of my data in, I copied the same tableau worksheets, dashboards, and story that I used to create my D210 dashboard. After I was finished with that the dashboard was complete and ready to save and upload.

**C5.**

From the analysis I found that the WGU dataset tends to lean older in age than the California dataset for people based in California. WGU also charges significantly higher than the other telecom company and this can be reflected in the churn rates. WGU does a decent job at retaining customers in California on the month to month contract types, but struggles immensely at retaining the two year contracts. The two year contracts had the same churn rate as the one year contracts at over 17%. The California telecom company does a significantly better job at retaining those customers. This is a key point of analysis for me as the two year customers should be the easiest to retain, and yet WGU struggles to in comparison to their contemporaries. This all culminated in showcasing the purpose and functionality of the dashboard, as the dashboard helped us understand the trends and metrics that were occurring in the data.

**C6.**

Some limitations of the analysis are that I only looked at contract type and monthly charge. There could be other factors that contribute to churn rate that aren’t shown in the analysis. The overall sample size of the data also shrank down to under 500 people for the analysis at the end, so it could be more volatile than if there were 50,000 people still in the analysis.

**D.**

Dr. Sewell, W (n.d). *D211 Webinar.* Retrieved November 13th, 2024,From D211 Course Search

Neon.tech. (n.d). *PostgreSQL SELET DISTINCT.* Retrieved November 16th, 2024 From [**https://neon.tech/postgresql/postgresql-tutorial/postgresql-select-distinct**](https://neon.tech/postgresql/postgresql-tutorial/postgresql-select-distinct)

Tableau.com. (n.d). *Connect ot a Custom SQL Query.* Retrieved November 13th, 2024 From [**https://help.tableau.com/current/pro/desktop/en-us/customsql.htm**](https://help.tableau.com/current/pro/desktop/en-us/customsql.htm)

W3Schools. (n.d). *SQL FOREIGN KEY Constraint.* Retrieved November 17th, 2024 From <https://www.w3schools.com/sql/sql_foreignkey.asp>