Using SQL to Create a Tableau Dashboard

Department of Information Technology, Western Governor's University

D211: Advanced Data Acquisition

Dr. William Sewell

June 16, 2024

A1. Datasets

For this project, I will be using the 2017-2018 NHANES dataset along with the WGU medical dataset already available in pgAdmin (“NHANES Questionnaires,” n.d.). The NHANES dataset comes as several .csv files, four of which I will actually be using. Since I am not quite clear on what exactly is required by this section, I will attach more than is probably required. The files attached are as follows:

1. NHANES dataset files:
    a. Cardio.csv – This file contains NHANES information related to cardiac conditions. **You will need this file for dashboard installation.**
    b. Conditions.csv – This file contains NHANES information related to a myriad of other conditions not befitting of the other NHANES files such as cardio.csv. **You will need this file for dashboard installation.**
    c. Demographics.csv – This file contains NHANES information related to patient demographics. **You will need this file for dashboard installation.**
    d. Diabetes.csv – This file contains NHANES information related to diabetes. **You will need this file for dashboard installation.**
2. Master data files:
    a. Nhanes_master.csv – This file is the NHANES dataset after its constituent .csv files have been cleaned up and married together in pgAdmin. This is the “external dataset.”
    b. Tableau_table.csv – This file is an export of the database table that my Tableau dashboard uses. This is a UNION of the external dataset, nhanes_master, and

several columns from the patient and servicesaddon tables already present in WGU's medical_data database.

3. SQL Code files:
    a. SQL for D211.txt – This file contains all of the SQL code required to get the pgAdmin database to look like the database I used to construct the dashboard. It should create all the tables you need for the dashboard. **You will need this file for dashboard installation.**
4. Tableau Dashboard files:
    a. D211 Dashboard.twbx – This file is the packaged workbook version of the dashboard I created. **You will need this file for dashboard installation.**

A2. Dashboard Instructions

The dashboard is provided in a .twbx format, and the SQL required to get the pgAdmin database to look like the database I used to construct the dashboard is provided as a .txt file. Below are the instructions on how to set up the dashboard for grading purposes. If you are instead looking for detailed, step-by-step instructions on how the dashboard itself and its visualizations were created, please refer to section C4.

1. Take the SQL for D211.txt file included with this submission and open it on the virtual machine (otherwise known as Labs on Demand) using the program called Notepad. You should be able to just download the file to the virtual machine, then just double-click the file to open it. It will open in Notepad by default.
2. Take the four .csv files included with this submission and paste them into the folder located at the following path: C:\Users\Public\Documents. You should be able to just

download these files to the virtual machine, then select them all and drag them to this folder once you find it.

3. Open pgAdmin4 by double-clicking the program icon on the desktop.
4. Expand “Servers,” then expand “PostgreSQL 13,” then expand “Databases (3),” then expand “medical_data.” Please note that after you expand “Servers,” the rest may automatically expand.
5. Right-click “medical_data,” and in the menu that pops up, select “Query Tool.”
6. Go to the .txt file from before—the one in Notepad. Click “Edit” in the top left, then select “Select All” from the menu that appears.
7. On your keyboard, press Ctrl and C at the same time to copy. Alternatively, right-click on the blue, selected text in Notepad and select “Copy” from the menu that appears.
8. Go back to pgAdmin and paste this text into the query editor. You can paste by pressing Ctrl and V at the same time on your keyboard after clicking on the query editor. Alternatively, you can right-click in the query editor and select “Paste” from the menu that appears.
9. Press the “play button” just above the query editor to run the query. This will create all the tables needed for the dashboard, as well as perform the necessary data cleaning.
10. Double-click the Tableau icon on the desktop that has the white background to open Tableau.
11. Download the .twbx file submitted with this assignment to the virtual machine.
12. In the top left of Tableau, click “File,” then “Open,” and find where you downloaded the .twbx file. Highlight the file and click “Open.”
13. When prompted for a password, type “Passw0rd!” into the box and click “Sign In.”

14. The dashboard should appear and be functional.

A3. Navigation Instructions

After opening the dashboard via the instructions above, there are a number of ways you can navigate around the dashboard. It should default to the dashboard tab within the workbook titled “Medical Data Dashboard.” If it does not, at the bottom of the workbook, please click the right-most tab titled “Medical Data Dashboard” to bring up the dashboard for this assignment. If desired, one can make the dashboard full screen by clicking the “projector screen” button in the middle of the gray bar just above the dashboard’s visualizations.

The dashboard is interactable and one such way is through tooltips. If you hover over any bar, datapoint, pie slice, and more, a tooltip will appear with information that should be helpful to you regarding that bit of the visualization. For example, the boxplot’s tooltips provide median, the hinges, and the values of the upper and lower whiskers. The age histogram’s tooltip provides the bin name and an exact count of people in that bin.

In addition to tooltips, the dashboard is very filterable. The pie chart visualization can be used as a filter itself— no other visualizations function as a filter on this dashboard. To use the pie chart as a filter, simply click the slice of the pie you’re interested in filtering on. For example, you might click the orange slice of the CDC pie, which will automatically filter the whole dashboard to only show patients who are part of the CDC data and are male. To remove this filter, simply click anywhere outside of the pie charts. You may also want to filter on two pie slices at once. Perhaps you want to filter on all male patients, regardless of data source. To do this, hold Ctrl on your keyboard and select each slice. Removing this filter is exactly the same as before.

There are also a number of easy-to-use filters at the top of the dashboard. The one for gender is a multi-select checkbox filter. To use this filter, simply check or uncheck the genders as you see fit. The condition filters to the right of the gender filter are single selection dropdowns. To use these filters, simply click the down arrow to the right of the dropdown selection and select “Yes” if you would like to see data on patients with that condition, “No,” if you would like to see data on patients without that condition, or “All” if you would like to see everyone regardless of whether or not they have that condition. To clear any of these filters along the top, simply click the “funnel with an x” icon in the top right of the filter you wish to clear. You may use these filters alone or at the same time as each other. Each filter you apply creates an AND condition with previously applied filters. For example, if you set the gender filter to “Male” by checking the box and also select “Yes” in the dropdown for the stroke filter, the dashboard will update to show men from any data source who have had a stroke.

Finally, please note that the “Times Higher than CDC” KPIs have more detail provided below them. The tables with the headers “CDC” on the left and “WGU” on the right directly below the KPIs show the raw percentages of people with the specified condition that the KPI is based on. For example, without any filters selected, the Times Higher than CDC KPI shows the proportion of WGU patients that are overweight is 1.864 times higher than the CDC. Below this KPI, you can see that 38.06% of CDC patients are overweight, while 70.94% of WGU patients are overweight.

A4. SQL Code

Below is all the code I used in PgAdmin4 to create the final table that is a union of the external data and WGU’s data that was already present in the SQL database.

```
/* Before messing with external data, the WGU data needs to be cleaned a bit.
Let's start by fixing the column name typo in "hignblood" */

ALTER TABLE patient
RENAME COLUMN hignblood TO highblood;

/* To recreate D210 dashboard, I need a source column. */

ALTER TABLE patient
ADD COLUMN source text;

UPDATE patient
SET source = 'WGU';

/* The data dictionary indicates "prefer not to answer" in the gender column means nonbinary.
I will update this. */

UPDATE patient
SET gender = REPLACE(gender, 'Prefer not to answer', 'Nonbinary');

/* NHANES top-codes people older than 80 as 80, so I need to
do this in the WGU dataset too. */

UPDATE patient
SET age = 80
WHERE age > 80;

/* The NHANES dataset is made up of different CSV files. I will ignore the CSVs that do not
contain information I'm interested in, as they are very large files with hundreds of columns
in some cases. There is no need to code all of that if I'm not going to use it.
Many of the variables in the demographics CSV are categorical despite being
represented as numbers-- those datatypes will be set to text. */

CREATE TABLE nhanes_demo(
	SEQN text, SDDSRVYR text, RIDSTATR text, RIAGENDR text, RIDAGEYR numeric, RIDAGEMN numeric,
	RIDRETH1 text, RIDRETH3 text, RIDEXMON text, RIDEXAGM numeric, DMQMILIZ text, DMQADFC text,
	DMDBORN4 text, DMDCITZN text, DMDYRSUS text, DMDEDUC3 text, DMDEDUC2 text, DMDMARTL text,
	RIDEXPRG text, SIALANG text, SIAPROXY text, SIAINTRP text, FIALANG text, FIAPROXY text, FIAINTRP text,
	MIALANG text, MIAPROXY text, MIAINTRP text, AIALANGA text,
	DMDHHSIZ text, DMDFMSIZ text, DMDHHSZA numeric,
	DMDHHSZB numeric, DMDHHSZE numeric, DMDHRGND text,
	DMDHRAGZ text, DMDHREDZ text, DMDHRMAZ text,
	DMDHSEDZ text, WTINT2YR numeric, WTMEC2YR numeric, SDMVPSU
	text, SDMVSTRA text, INDHHIN2 text,
	INDFMIN2 text, INDFMPIR numeric,

	CONSTRAINT sequence_no PRIMARY KEY (SEQN)
);

/* import from demo file and clean it up a little */

COPY nhanes_demo
FROM 'C:\Users\Public\Documents\demographics.csv'
DELIMITER ','
CSV HEADER;

/* fix seqn to not have that .0 it has for some reason */

UPDATE nhanes_demo
SET seqn = regexp_replace(seqn, '\.0$', '', 'g')
WHERE seqn LIKE '%.0';

/* Round columns I care about */

UPDATE public.nhanes_demo
SET ridageyr = ROUND(ridageyr);

UPDATE nhanes_demo
SET dmdhhsza = ROUND(dmdhhsza);

UPDATE nhanes_demo
SET dmdhhszb = ROUND(dmdhhszb);

/* Rename columns that I'm going to use in my Tableau dashboard to be
human-readable. */

ALTER TABLE nhanes_demo
RENAME COLUMN riagendr TO gender;

ALTER TABLE nhanes_demo
RENAME COLUMN ridageyr TO age;

ALTER TABLE nhanes_demo
RENAME COLUMN dmdhhsza TO young_children;
```

```
ALTER TABLE nhanes_demo
RENAME COLUMN dmdhhszb TO old_children;

/*  create and import diabetes table. Because there are codes for "refused" and "unknown"
and such in otherwise
numeric columns, I will import all the columns as text to begin with. I don't need a
lot of these columns anyway, and will probably drop them from the final, joined table. */

CREATE TABLE public.nhanes_diabetes (
   SEQN text, DIQ010 text, DID040 text, DIQ160 text, DIQ170 text, DIQ172 text,
   DIQ175A text, DIQ175B text, DIQ175C text, DIQ175D text, DIQ175E text, DIQ175F
text,
   DIQ175G text, DIQ175H text, DIQ175I text, DIQ175J text, DIQ175K text, DIQ175L
text,
   DIQ175M text, DIQ175N text, DIQ175O text, DIQ175P text, DIQ175Q text,
DIQ175R text,
   DIQ175S text, DIQ175T text, DIQ175U text, DIQ175V text, DIQ175W text,
DIQ175X text,
   DIQ180 text, DIQ050 text, DID060 text, DIQ060U text, DIQ070 text, DIQ230 text,
   DIQ240 text, DID250 text, DID260 text, DIQ260U text, DIQ275 text, DIQ280 text,
   DIQ291 text, DIQ300S text, DIQ300D text, DID310S text, DID310D text, DID320
text,
   DID330 text, DID341 text, DID350 text, DIQ350U text, DIQ360 text, DIQ080 text,

         CONSTRAINT seqn_no PRIMARY KEY (SEQN)
);

/* import from diabetes file and clean it up a little */

COPY nhanes_diabetes
FROM 'C:\Users\Public\Documents\diabetes.csv'
DELIMITER ','
CSV HEADER;

/* fix seqn to not have that .0 it has for some reason */

UPDATE nhanes_diabetes
SET seqn = regexp_replace(seqn, '\.0$', '', 'g')
WHERE seqn LIKE '%.0';

/* Rename columns that I'm going to use in my Tableau dashboard to be human-readable.
*/

ALTER TABLE nhanes_diabetes
RENAME COLUMN diq010 TO diabetes;
```

```
/*  create and import cardio table. Because there are codes for "refused" and "unknown"
and such in otherwise numeric columns, I will import all the columns as text to begin
with. I don't need a lot of these columns anyway, and will probably drop them from the
final, joined table. */

CREATE TABLE public.nhanes_cardio (
   SEQN text, BPQ020 text, BPQ030 text, BPD035 text,
   BPQ040A text, BPQ050A text, BPQ080 text, BPQ060 text,
   BPQ070 text, BPQ090D text, BPQ100D text,
         CONSTRAINT seq_no PRIMARY KEY (SEQN)
);

/* import from cardio file and clean it up a little */

COPY nhanes_cardio
FROM 'C:\Users\Public\Documents\cardio.csv'
DELIMITER ','
CSV HEADER;

/* fix seqn to not have that .0 it has for some reason */

UPDATE nhanes_cardio
SET seqn = regexp_replace(seqn, '\.0$', '', 'g')
WHERE seqn LIKE '%.0';

/* Rename columns that I'm going to use in my Tableau dashboard to be human-readable.
*/

ALTER TABLE nhanes_cardio
RENAME COLUMN bpq080 TO hyperlipidemia;

ALTER TABLE nhanes_cardio
RENAME COLUMN bpq020 TO highblood;

/*  create and import conditions table. Because there are codes for "refused" and
"unknown" and such in otherwise
numeric columns, I will import all the columns as text to begin with. I don't need a
lot of these columns anyway, and will probably drop them from the final, joined table. */

CREATE TABLE nhanes_condits (
   SEQN TEXT, MCQ010 TEXT, MCQ025 TEXT, MCQ035 TEXT, MCQ040 TEXT,
   MCQ050 TEXT, AGQ030 TEXT, MCQ053 TEXT, MCQ080 TEXT, MCQ092
TEXT,
   MCD093 TEXT, MCQ149 TEXT, MCQ151 TEXT, RHD018 TEXT, MCQ160A
TEXT,
   MCD180A TEXT, MCQ195 TEXT, MCQ160N TEXT, MCD180N TEXT, MCQ160B
TEXT,
   MCD180B TEXT, MCQ160C TEXT, MCD180C TEXT, MCQ160D TEXT,
MCD180D TEXT,
   MCQ160E TEXT, MCD180E TEXT, MCQ160F TEXT, MCD180F TEXT,
MCQ160M TEXT,
   MCQ170M TEXT, MCD180M TEXT, MCQ160G TEXT, MCD180G TEXT,
MCQ160K TEXT,
   MCQ170K TEXT, MCD180K TEXT, MCQ160O TEXT, MCQ160L TEXT,
MCQ170L TEXT,
   MCD180L TEXT, MCQ500 TEXT, MCQ510A TEXT, MCQ510B TEXT, MCQ510C
TEXT,
   MCQ510D TEXT, MCQ510E TEXT, MCQ510F TEXT, MCQ520 TEXT, MCQ530
TEXT,
   MCQ540 TEXT, MCQ550 TEXT, MCQ560 TEXT, MCQ570 TEXT, MCQ203
TEXT,
   MCQ206 TEXT, MCQ220 TEXT, MCQ230A TEXT, MCD240A TEXT, MCQ230B
TEXT,
   MCD240B TEXT, MCQ230C TEXT, MCD240C TEXT, MCQ230D TEXT,
MCQ300B TEXT,
   MCQ300C TEXT, MCQ300A TEXT, MCQ366A TEXT, MCQ366B TEXT,
MCQ366C TEXT,
   MCQ366D TEXT, MCQ371A TEXT, MCQ371B TEXT, MCQ371C TEXT,
MCQ371D TEXT,
   OSQ230 TEXT,
         CONSTRAINT seqn_num PRIMARY KEY (SEQN)
);

/* import from condits file and clean it up a little */

COPY nhanes_condits
FROM 'C:\Users\Public\Documents\conditions.csv'
DELIMITER ','
CSV HEADER;

/* fix seqn to not have that .0 it has for some reason */

UPDATE nhanes_condits
SET seqn = regexp_replace(seqn, '\.0$', ", 'g')
WHERE seqn LIKE '%.0';

/* Rename columns that I'm going to use in my Tableau dashboard
to be human-readable. */

ALTER TABLE nhanes_condits
RENAME COLUMN MCQ010 TO asthma;
```

```
ALTER TABLE nhanes_condits
RENAME COLUMN MCQ080 TO overweight;

ALTER TABLE nhanes_condits
RENAME COLUMN MCQ160A TO arthritis;

ALTER TABLE nhanes_condits
RENAME COLUMN MCQ160F TO stroke;

/* Clean up coded values so that they are human readable for columns
that I care about */

UPDATE nhanes_demo
SET gender = CASE WHEN gender = '1.0' THEN 'Male' WHEN gender = '2.0' THEN
'Female' END;

UPDATE nhanes_cardio
SET highblood = CASE WHEN highblood = '1.0' THEN 'Yes' WHEN highblood = '2.0'
THEN 'No'
WHEN highblood = '7.0' THEN NULL WHEN highblood = '9.0' THEN 'No' END;

UPDATE nhanes_cardio
SET hyperlipidemia = CASE WHEN hyperlipidemia = '1.0' THEN 'Yes' WHEN
hyperlipidemia = '2.0' THEN 'No'
WHEN hyperlipidemia = '7.0' THEN NULL WHEN hyperlipidemia = '9.0' THEN 'No'
END;

UPDATE nhanes_diabetes
SET diabetes = CASE WHEN diabetes = '1.0' THEN 'Yes' WHEN diabetes = '2.0' THEN
'No'
WHEN diabetes = '3.0' THEN 'No' WHEN diabetes = '7.0' THEN NULL
WHEN diabetes = '9.0' THEN 'No' END;

UPDATE nhanes_condits
SET stroke = CASE WHEN stroke = '1.0' THEN 'Yes' WHEN stroke = '2.0' THEN 'No'
WHEN stroke = '7.0' THEN NULL WHEN stroke = '9.0' THEN 'No' END;

UPDATE nhanes_condits
SET overweight = CASE WHEN overweight = '1.0' THEN 'Yes' WHEN overweight =
'2.0' THEN 'No'
WHEN overweight = '7.0' THEN NULL WHEN overweight = '9.0' THEN 'No' END;

UPDATE nhanes_condits
SET arthritis = CASE WHEN arthritis = '1.0' THEN 'Yes' WHEN arthritis = '2.0' THEN
'No'
```

```
WHEN arthritis = '7.0' THEN NULL WHEN arthritis = '9.0' THEN 'No' END;

UPDATE nhanes_condits
SET asthma = CASE WHEN asthma = '1.0' THEN 'Yes' WHEN asthma = '2.0' THEN
'No'
WHEN asthma = '7.0' THEN NULL WHEN asthma = '9.0' THEN 'No' END;

/* Create master NHANES table that includes important columns from all the smaller
tables. */

CREATE TABLE nhanes_master AS(
	SELECT demo.age, demo.gender, demo.young_children + demo.old_children AS
children,
	condits.arthritis, condits.asthma, diabetes.diabetes, cardio.highblood,
cardio.hyperlipidemia,
	condits.overweight, condits.stroke
	FROM nhanes_demo AS demo
	INNER JOIN nhanes_condits AS condits
	ON demo.seqn = condits.seqn
	INNER JOIN nhanes_cardio AS cardio
	ON condits.seqn = cardio.seqn
	INNER JOIN nhanes_diabetes AS diabetes
	ON cardio.seqn = diabetes.seqn
);

/* To recreate D210 dashboard, I need a source column. */

ALTER TABLE nhanes_master
ADD COLUMN source text;

UPDATE nhanes_master
SET source = 'CDC';

/* Set all NULLs = to No, since non-response is likely to just be a "No." */

UPDATE nhanes_master
SET hyperlipidemia = 'No'
WHERE hyperlipidemia IS NULL;

UPDATE nhanes_master
SET stroke = 'No'
WHERE stroke IS NULL;

UPDATE nhanes_master
SET arthritis = 'No'
WHERE arthritis IS NULL;
```

```
/* WGU dataset doesn't include minors, so we need to remove them from NHANES. */

DELETE FROM nhanes_master
WHERE age < 18;

/* Create foreign keys for WGU database since they're oddly missing to enforce
referential integrity. */
ALTER TABLE servicesaddon
ADD FOREIGN KEY (patient_id) REFERENCES patient(patient_id);

ALTER TABLE survey_responses_addon
ADD FOREIGN KEY (patient_id) REFERENCES patient(patient_id);

/* Create foreign keys for NHANES data */

ALTER TABLE nhanes_cardio
ADD FOREIGN KEY (seqn) REFERENCES nhanes_demo(seqn);

ALTER TABLE nhanes_diabetes
ADD FOREIGN KEY (seqn) REFERENCES nhanes_demo(seqn);

ALTER TABLE nhanes_condits
ADD FOREIGN KEY (seqn) REFERENCES nhanes_demo(seqn);

/* Time to union! We could do this in Tableau, but I don't want to. */

CREATE TABLE wgu_temp AS(
        SELECT patient.age, patient.gender, patient.children, condits.arthritis,
        condits.asthma, condits.diabetes, patient.highblood, condits.hyperlipidemia,
        condits.overweight, patient.stroke, patient.source
        FROM patient
        INNER JOIN servicesaddon AS condits
        ON patient.patient_id = condits.patient_id
);

CREATE TABLE tableau_table AS(
        SELECT *
        FROM wgu_temp
        UNION ALL
        SELECT *
        FROM nhanes_master
);
```

B. Panopto Presentation

https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=e16894bc-653b-4d0c-bb36-b1940006dc90

C1. Alignment with Organizational and Stakeholder Needs

WGU's medical data dictionary signifies a need to address readmission rates, since excessive readmission can result in monetary penalties for the hospital system. Because I struggled to find a dataset that could be used to compare the WGU hospital system to the country's hospital readmission rates, I was forced to take a broader approach than might be anticipated by the audience.

Instead of trying to benchmark the WGU hospital system's readmission rate against national averages, I instead analyzed patient demographics and health conditions as compared to CDC data that represents the country as a whole. Intuitively speaking, being overweight and having high blood pressure are both complicating factors during hospital stays and could lead to longer stays or readmissions. In addition, older patients are inherently more at risk. Ideally, prior analysis using a machine learning technique would back up this statement of intuition. Thus, analyzing how WGU's patients compare to the nation as a whole in terms of these two condition factors and patient age would be beneficial. With this analysis, the WGU hospital system can be proactive about how it works up patients because it has knowledge about the averages of its patient base compared to the national averages.

The dashboard allows for exploration of patient age and gender using a histogram and a boxplot, respectively. Gender is primarily present for slicing purposes, in case there are trends in high blood pressure or overweight status that are different based on gender. Using the histogram to view the WGU hospital system's patient age distribution may help inform WGU so it can make effective decisions regarding where to allocate resources, such as geriatric or pediatric units, that may prevent readmissions.

The two KPIs, which both signify how many times higher than the national average WGU's patient base is for each of two conditions, provide in the form of a single number how much WGU's patient base is (in most cases,) at higher risk than the rest of the country. Having one or both of the conditions highlighted, as aforementioned, is a complicating factor in a hospital stay and could lead to readmission, which the hospital must avoid. Executives of the WGU hospital system can use these KPIs to make informed decisions regarding what programs may need to be started in order to be proactive and prevent these two preventable conditions, which, in turn, would allow the lowering of readmission rates (Nelson, 2024).

C2. Justification of Business Intelligence Tool

The recommended tool for the analysis this assignment requires was Tableau Desktop. Tableau is a powerful tool for data visualization as well as basic data exploration and analysis. It allows for the creation of highly interactive dashboards containing visualizations of many types that are replete with informative tooltips that appear as one simply hovers over any datapoint. In Tableau, everything is customizable—making it easier for the analyst to uncover and communicate patterns that might otherwise go unnoticed in the SQL database, since SQL databases do not have robust visualization tools. Slicers are easily added by the data analyst, making a well-designed Tableau dashboard a one-stop shop for any stakeholder, be they a data analyst or a C-suite executive. At the risk of sounding like an ad for Tableau, a dashboard makes data analysis quick and easy—anyone can explore the data—no need to understand complicated SQL or Python. One might even say it *democratizes* data.

In addition, Tableau also has an easy connector for PostgreSQL, the type of SQL database in which this hospital system's data was stored. No need to convert between formats—the data analyst can just set up the connector and get creating. Tableau can also handle a variety of other formats including .csv files and Excel sheets, should new data ever need to be added in these formats or a large number of other formats which I will not enumerate here.

C3. Data Preparation

The steps I took to prepare the data are as follows:

1. Fix errors in column names (specifically, hignblood, which should have been "highblood.")
2. Add a source column to the WGU medical data so it can be distinguished from the NHANES data.
3. Replace "Prefer not to answer" in the gender column with "Nonbinary," as indicated in the data dictionary for the medical dataset.
4. Top-code WGU's patients who are older than 80 as 80 years old in order to stay consistent with the NHANES data, which top-codes their respondents over 80 as 80 years old.
5. Create tables to house the various .csv files from the NHANES dataset. There are four such tables created:
    a. Nhanes_demo, into which I loaded the demographics.csv file.
    b. Nhanes_diabetes, into which I loaded the diabetes.csv file.
    c. Nhanes_cardio, into which I loaded the cardio.csv file.
    d. Nhanes_condits, into which I loaded the conditions.csv file.
6. Use COPY to upload the appropriate file into the appropriate table (detailed above).

7. Use a regular expression to clean up the seqn column in each file so that it does not have a “.0” at the end of the sequence number—these are unique identifiers, and the appending of “.0” by the database makes for dirty data.
8. Round the columns ridageyr, dmdhhsza, and dmdhhszb columns in nhanes_demo, since these are the age of the patient and number of children (younger and older) that the patient has, and they should not be decimals.
9. Rename the columns I will be using from each NHANES file to a human-readable column name.
10. Recast the coded condition and gender values to the appropriate human-readable categories (ex. 1.0 = male, 2.0 = female, etc. or 1.0 = Yes, 2.0 = No, etc.)
11. Create a master NHANES dataset table called nhanes_master to combine all of the smaller tables that resulted from having multiple .csv files using an INNER JOIN.
    a. During the creation of this table, add the young_children and old_children columns together to create a children column, since the WGU data does not delineate between older and younger children.
12. Create and fill a source column for the nhanes_master table that makes it easily identifiable as CDC data.
13. Set all null values equal to “No”—making the assumption that non-response means the patient probably does not have the condition.
14. Drop rows from nhanes_master where the patient’s age is less than 18, since the WGU data does not include minors. This is for an apples-to-apples comparison.

15. Create a temporary WGU data table that gathers only the columns I will be using into one table called wgu_temp. This requires performing an INNER JOIN on the tables patient and servicesaddon.
16. UNION ALL wgu_temp and nhanes_master into one table called tableau_table, which is my final table. It will be used to make the dashboard without having to perform the union in Tableau.

C4. Dashboard Creation

Below are the steps used to create the dashboard, including data connection, visualization creation, and dashboard creation.

1. Loading Data
    a. On the blue, left pane, click "More.." under the "To a Server" subsection.
    b. Click “PostgreSQL.”
    c. In the box that pops up, enter the following in the fields indicated:
        i. Server: localhost
        ii. Port: 5432
        iii. Database: medical_data
        iv. Authentication: set to “Username and Password”
        v. Username: postgres
        vi. Password: Passw0rd!
    d. Click “Sign In.”
    e. Click "Sheet 1" at the bottom of the screen. Double-click "Sheet 1" and change the name to "Age Boxplot." This is the first visualization we will make after we do some preprocessing.

f. In the gray, left panel of the Data Source tab, double-click the “tableau_table” table to connect to it.

2. Preprocessing Dataset

   a. Tableau automatically tries to assign variables appropriate datatypes and place them correctly either as dimensions (above the gray line on the Data pane on the left) or measures (below the gray line.) Sometimes, we need to make modifications, as Tableau does make mistakes from time to time. However, all the Yes/No condition variables appear to be correctly assigned the string datatype and are dimensions, while Age and Children have been assigned a numerical datatype and are appropriately measures. Thus, no preprocessing is needed in Tableau before making our visualizations.

3. Age Boxplot Visualization

   a. Drag "Age" from the Data pane to the Rows shelf.

   b. Click "Analysis" on the white ribbon at the top and uncheck "aggregate measures."

   c. Click "Show Me" in the top right and select the boxplot icon to create a box plot.

   d. Drag "Source" from the Data pane to the Columns shelf to separate the data by source (i.e. WGU vs CDC data.) This gives you two box plots.

   e. Click on "Analytics" on the left to switch from the Data pane to the Analytics pane. Drag "Average Line" to the visualization and drop it on the tile called "Cell." This adds average lines that are specific to each boxplot.

   f. Click the gray part of the boxplot and click "Format" on the box that pops up.

   g. Change the "Fill" dropdown on the left to "Orange" because it stands out more than the gray.
   h. Use the sliding switch to change the opacity to 50%.
   i. Click the "x" on the Format section so you can see the Data pane again.
   j. Right-click "Gender" in the Data pane and click "Show Filter."
   k. Right-click "Source" at the top of the visualization (a label) and click "Hide Field Labels for Columns."

4. Age Histogram Visualization
   a. Click the "add new worksheet" icon at the bottom and name it "Age Histogram."
   b. Select "Age" from the Data pane and then click "Show Me."
   c. In the Show Me menu, select the histogram icon.
   d. Now, drag "Source" from the Data pane to the columns shelf.
   e. In the Data pane, right-click the newly-made "Age (bin)" data field.
   f. Click "Edit" in the menu, then change "Size of bins" to 3. Click "OK."
   g. Right-click the y-axis and click "Edit Axis."
   h. Change the axis title to "Number of People" and exit.
   i. Right-click "Source" at the top of the visualization (a label) and click "Hide Field Labels for Columns."
   j. Right-click "Arthritis" in the Data pane and click "Show Filter." Repeat for "Asthma," "Diabetes," "Highblood," "Hyperlipidemia," "Overweight," and "Stroke."
   k. For each filter on the right, click the small black down-arrow in the top right of the filter and select "Single Value (dropdown)." If they default to being filtered on

"No," clear the filter with the funnel-with-an-x icon in the top right of the filter in question.

l. Double-click the tooltip marks card and edit the text in the text box. Replace "Count of Age" with "# of People". Delete "Source: < Source >." Click "OK."

5. Average # of Children Bar Chart Visualization

    a. Click the "add new worksheet" icon at the bottom and name it "Avg # Children Bar Graph."

    b. Drag "Source" from the Data pane to the columns shelf.

    c. Drag "Age (bin)" from the Data pane to the columns shelf.

    d. Drag "Children" from the Data pane to the rows shelf, then right-click it, hover over "Measure (Sum)" and select "Average" from the menu that appears.

    e. Press and hold Ctrl on the keyboard while dragging "AVG(Children)" from the rows shelf to the label card. This adds labels over the bars.

    f. Right-click the y-axis and click "Edit Axis."

    g. Change the axis title to "Avg. # of Children" and exit.

    h. Right-click "Source" at the top of the visualization (a label) and click "Hide Field Labels for Columns."

    i. Double-click the tooltip marks card and edit the text in the text box. Delete "Source: < Source >." Change "Avg. Children" to "Avg. # of Children." Click "OK."

6. Gender Pie Chart Visualization

    a. Click the "add new worksheet" icon at the bottom and name it as "Gender Pie Chart."

b. Use the dropdown on the marks card to change the visualization type to "Pie."

c. Drag "Gender" from the Data pane to the color marks card.

d. Drag "Gender" from the Data pane again, but this time drag it to the angle marks card. Both of these will show up underneath the different marks cards.

e. Right-click the "Gender" bar with the angle icon and hover over "Measure," then select "Count."

f. Right-click the same "Gender" bar again (which now says "CNT(Gender)" and hover over "Quick Table Calculation."

g. From the menu that appears, click "Percent of Total." Right-click "CNT(Gender)" once more and click "Edit Table Calculation."

h. In the pop-up, under the "Compute Using" section, select "Table (down)" and exit. Hold Ctrl on your keyboard and, at the same time, drag the "CNT(Gender)" bar up to the label marks card.

i. Drag "Source" from the Data pane to the columns shelf.

j. Right-click "Source" at the top of the visualization (a label) and click "Hide Field Labels for Columns."

k. Click the "Color" marks card and select "Edit Colors" from the pop-up that appears.

l. In the menu that appears, use the "Select Color Palette" drop down to change the color palette to "Color Blind." Click "Assign Palette" and then click "OK."

m. Double-click the tooltip marks card and edit the text in the text box. Delete "Source: < Source >." Change "% of Total Count of Gender along Table (Down)" to "% of Total." Click "OK.

7. KPI – Overweight

    a. Click the "add new worksheet" icon at the bottom and name it as "KPI Overweight."
    b. Drag "Source" from the Data pane to the columns shelf.
    c. Drag "Overweight" from the Data pane to the rows shelf.
    d. Drag "Overweight" from the Data pane again, but this time to the Text card.
    e. Right-click the "Overweight" bar that appears on the marks card, hover over "Measure," and select "Count."
    f. Right-click this bar again, hover over "Quick Table Calculation," and from the menu that appears, click "Percent of Total."
    g. Right-click that same bar once more and click "Edit Table Calculation." In the pop-up, under the "Compute Using" section, select "Table (down)" and exit.
    h. On the visualization, right-click the "No" row-label and select "Hide."
    i. Right-click the title "Overweight" on the visualization and click "Hide Field Labels for Rows."
    j. Right-click "Source" at the top of the visualization (a label) and click "Hide Field Labels for Columns."
    k. Right-click the "Yes" row-label and click "Show Header" to uncheck it.
    l. Right-click the title and select "Edit Title." Center it using the three-lines icon that looks centered. Click "OK."
    m. On the gray ribbon across the top of the program, select the down-arrow next to a dropdown that currently says "Standard." Change this to "Fit Width."
    n. Right-click the datapoint in the visualization and click "Format."

  o. Click the alignment icon (which looks like a stack of lines) on the formatting pane, which covers the data pane for now. Click the first dropdown under the "Default" section. A small pop-up appears. In the "Horizontal" section, select the centered-lines icon.

  p. Double-click the tooltip marks card and edit the text in the text box. Delete "Source: < Source >." Change "% of Total Count of Overweight along Table (Down)" to "% of Total."

8. KPI - Overweight Times Higher

   a. Click the "add new worksheet" icon at the bottom and name it as "KPI Overweight Times Higher."

   b. Right-click "Overweight" on the Data pane, hover over "Create," then select "Calculated Field."

   c. In the box that pops up, name the calculation "# CDC Overweight." In the text box on this pop-up, type "SUM(IF[Overweight] = "Yes" AND [Source] = "CDC" THEN 1 ELSE 0 END)" and click "OK."

   d. Right-click "Overweight" on the Data pane again, hover over "Create," then select "Calculated Field."

   e. In the box that pops up, name the calculation "# WGU Overweight." In the text box on this pop-up, type "SUM(IF[Overweight] = "Yes" AND [Source] = "WGU" THEN 1 ELSE 0 END)" and click "OK."

   f. Right-click "# CDC Overweight" and click "duplicate." Then, right-click the newly created field and click "Edit."

g. Replace the formula in the box with "SUM(IF[Source] = "CDC" THEN 1 ELSE 0 END)". Rename it to "# CDC Patients."

h. Right-click "# WGU Overweight" and click "duplicate." Then, right-click the newly created field and click "Edit."

i. Replace the formula in the box with "SUM(IF[Source] = "WGU" THEN 1 ELSE 0 END)". Rename it to "# WGU Patients."

j. Right-click "Overweight" on the Data pane again, hover over "Create," then select "Calculated Field."

k. In the box that pops up, name the calculation "WGU Overweight Times Higher." In the text box on this pop-up, type "([# WGU Overweight]/[# WGU Patients])/([# CDC Overweight]/[# CDC Patients])" and click "OK."

l. From the Data pane, drag "WGU Overweight Times Higher" to the text marks card.

m. Right-click the title and select "Edit Title." Type "Overweight" then a carriage return, and "Times Higher than CDC." Center it using the three-lines icon that looks centered. Click "OK."

n. On the gray ribbon across the top of the program, select the down-arrow next to a dropdown that currently says "Standard." Change this to "Fit Width."

o. Right-click the datapoint in the visualization and click "Format." Click the alignment icon (which looks like a stack of lines) on the formatting pane, which covers the data pane for now. Click the first dropdown under the "Default" section. A small pop-up appears. In the "Horizontal" section, select the centered-lines icon.

p. Click on the small "A" to go back to the Format Font screen. Under the "Default" section, click the first dropdown, which is labeled "Worksheet." Use the dropdown to change the font to Tableau Bold, and the font size dropdown to change the font size to 14. Click one of the blue-colored squares to change the font color to a nice blue.

9. KPI - High Blood Pressure

a. Click the "add new worksheet" icon at the bottom and name it as "KPI High Blood Pressure."

b. Drag "Source" from the Data pane to the columns shelf.

c. Drag "Highblood" from the Data pane to the rows shelf.

d. Drag "Highblood" from the Data pane again, but this time to the Text card.

e. Right-click the "Highblood" bar that appears on the marks card, hover over "Measure," and select "Count."

f. Right-click this bar again, hover over "Quick Table Calculation," and from the menu that appears, click "Percent of Total."

g. Right-click that same bar once more and click "Edit Table Calculation." In the pop-up, under the "Compute Using" section, select "Table (down)" and exit.

h. On the visualization, right-click the "No" row-label and select "Hide."

i. Right-click the title "Highblood" on the visualization and click "Hide Field Labels for Rows."

j. Right-click "Source" at the top of the visualization (a label) and click "Hide Field Labels for Columns."

k. Right-click the "Yes" row-label and click "Show Header" to uncheck it.

l. Right-click the title and select "Edit Title." Center it using the three-lines icon that looks centered. Click "OK."

m. On the gray ribbon across the top of the program, select the down-arrow next to a dropdown that currently says "Standard." Change this to "Fit Width."

n. Right-click the datapoint in the visualization and click "Format." Click the alignment icon (which looks like a stack of lines) on the formatting pane, which covers the data pane for now. Click the first dropdown under the "Default" section. A small pop-up appears. In the "Horizontal" section, select the centered-lines icon.

o. Double-click the tooltip marks card and edit the text in the text box. Delete "Source: < Source >." Change "% of Total Count of High Blood along Table (Down)" to "% of Total."

10. KPI - High BP Times Higher

a. Click the "add new worksheet" icon at the bottom and name it as "KPI High BP Times Higher."

b. Right-click "High Blood" on the Data pane, hover over "Create," then select "Calculated Field."

c. In the box that pops up, name the calculation "# CDC High Blood." In the text box on this pop-up, type "SUM(IF[Highblood] = "Yes" AND [Source] = "CDC" THEN 1 ELSE 0 END)" and click "OK."

d. Right-click "High Blood" on the Data pane again, hover over "Create," then select "Calculated Field."

e. In the box that pops up, name the calculation "# WGU High Blood." In the text box on this pop-up, type "SUM(IF[Highblood] = "Yes" AND [Source] = "WGU" THEN 1 ELSE 0 END)" and click "OK."

f. Right-click "Highblood" on the Data pane again, hover over "Create," then select "Calculated Field."

g. In the box that pops up, name the calculation "WGU High BP Times Higher." In the text box on this pop-up, type "(([# WGU High Blood]/[# WGU Patients])/([# CDC High Blood]/[# CDC Patients]))" and click "OK."

h. From the Data pane, drag "WGU High BP Times Higher" to the text marks card.

i. Right-click the title and select "Edit Title." Type "High BP" then a carriage return, and "Times Higher than CDC." Center it using the three-lines icon that looks centered. Click "OK."

j. On the gray ribbon across the top of the program, select the down-arrow next to a dropdown that currently says "Standard." Change this to "Fit Width."

k. Right-click the datapoint in the visualization and click "Format." Click the alignment icon (which looks like a stack of lines) on the formatting pane, which covers the data pane for now. Click the first dropdown under the "Default" section. A small pop-up appears. In the "Horizontal" section, select the centered-lines icon.

l. Click on the small "A" to go back to the Format Font screen. Under the "Default" section, click the first dropdown, which is labeled "Worksheet." Use the dropdown to change the font to Tableau Bold, and the font size dropdown to change the font size to 14. Click one of the blue-colored squares to change the font color to a nice blue.

11. Making the Dashboard

   a. Click the "add new dashboard" icon at the bottom and name it as "Medical Data Dashboard."
   b. On the Dashboard pane on the left, change the dropdown under the section called "Size" to "Automatic."
   c. From the Dashboard pane, drag every sheet from the "Sheets" section to the dashboard and arrange them in a way that makes sense and is visually pleasing.
   d. From the Dashboard pane, at the bottom, find and drag "Horizontal" in the Objects section to the top of the dashboard. You'll know when to drop it when the top half of the dashboard is gray. Resize it as appropriate.
   e. From the Dashboard pane, drag "Text" to this new container. This adds a text box we can use for a title.
   f. A box will pop up. Type "Our Patients: How Can We Help?" into the text field, then change the font to "Tableau Bold" using the leftmost dropdown. In the dropdown right next to this one, change the font size to 26 and use the color dropdown to change the color to blue to match the rest of the dashboard. Click "OK."
   g. Right-click the "Age Histogram" title and click "Hide Title."
   h. Right-click the "Avg # Children Bar Graph" title and click "Hide Title."
   i. Right-click the "Gender Pie Chart" title and click "Hide Title."
   j. Right-click the "Age Boxplot" title and click "Hide Title."

k. Right-click the "KPI Overweight" title and click "Hide Title."

l. Right-click the "KPI High Blood Pressure" title and click "Hide Title."

m. Click the Gender legend on the right side where the filters are and click the little down-arrow. This opens a menu. From the menu, click "Floating" and drag the resulting legend near the pie charts.

n. Click the bar containing the rest of the filters so that it's selected. Click the small down-arrow and select "Floating."

o. Select the now floating container and click the small down-arrow again, but this time select "Remove Container."

p. From the Dashboard pane, at the bottom, find and drag "Horizontal" in the Objects section to the top of the dashboard. You'll know when to drop it when the top-right half of container we placed earlier (which contains the title) is gray. Resize it as appropriate.

q. Drag the now separate filters to the new horizonal container at the top of the dashboard and arrange them in a visually pleasing way.

r. For each filter, select it and click the small down-arrow. Click “Floating” to uncheck it. The filter should attach itself to the horizontal container. If it doesn’t, drag it up there.

s. For each filter, select it and click the small down-arrow, hover over "Apply to Worksheets," then click "All Using This Data Source."

t. Select the pie chart visualization and click the little "funnel" icon to make the visualization able to be used as a filter for the whole dashboard.

C5. Data Analysis Results

The dashboard I created illuminated three findings I found significant. First, that the WGU patient base distribution is very uniform, no matter the subset of patients you look at. This is entirely unlike the NHANES dataset, whose patients follow expected patterns for certain conditions, such as arthritis. For example, where NHANES patients show more older patients having arthritis (a left skewed distribution,) the WGU patients who have arthritis are much more uniform in age, where almost as many younger patients have arthritis as older patients. As a result, some groups of WGU patients, such as younger patients with arthritis, may be neglected because they are afflicted by diseases not common to their age group nationwide, which could lead to avoidable readmissions.

Secondly, WGU patients are diagnosed with high blood pressure and obesity at substantially higher rates than the NHANES national dataset. These conditions are often preventable while also being risk factors for readmission since they complicate care. Thus, it may be helpful to allocate resources toward preventative care programs that could reduce the number of patients with these two conditions in the WGU patient base.

Lastly, the WGU patient base contains substantially more geriatric patients than the NHANES national dataset. With this in mind, readmissions might be reduced if more resources are directed toward geriatric care. These findings directly relate to the goal of the dashboard: to find trends that could lead to readmission and find ways to reduce readmission rates for the WGU hospital system so as to avoid monetary penalties.

C6. Limitations of Data Analysis

One limitation of this analysis is that while the webpage for the NHANES dataset states that "the survey examines a nationally representative sample of about 5,000 persons each year," whether or not this is actually representative of the United States as a whole remains questionable because of non-response bias (NHANES - About the National Health and Nutrition Examination Survey, n.d.). The NHANES survey is conducted within a patient's home, which may be uncomfortable for those concerned with privacy. In fact, even taking a government survey—especially a very long, labor-intensive one such as NHANES—can be unpalatable for some.

Another limitation is that because the WGU data represents people who have already been hospitalized, our data analysis may not be making an apples-to-apples comparison despite our attempts to make it so. People who have already been hospitalized were hospitalized for a reason, after all—usually serious reasons. The patient base that the NHANES survey covers may or may not have been hospitalized at any point in their lives, which may make the NHANES survey respondents look healthier than WGU patients on average. It may also make serious issues appear more common among WGU patients.

Lastly, there are limitations due to how both the WGU dataset and the NHANES dataset handle age. Because the NHANES dataset top-codes people older than 80 as 80 years old, we lose information about people older than 80. There may be patterns amongst the oldest in the patient population that we simply cannot see due to the top-coding. WGU's dataset is not perfect either, however, since it omits minors. Minors made up a significant portion of the NHANES dataset, and their omission negatively impacts analysis since again, we are now blind to patterns and trends that might emerge in WGU data on minors. Without this data, WGU will have no way of knowing if minors significantly contribute to the readmission rate. Implementing plans and goals based on analysis on only adults to pediatric units may actually result in negative results.

D. Web Sources

Centers for Disease Control and Prevention. (n.d.). *Nhanes questionnaires, datasets, and related documentation*. Centers for Disease Control and Prevention. https://wwwn.cdc.gov/nchs/nhanes/continuousnhanes/default.aspx?BeginYear=2017

NHANES - about the National Health and Nutrition Examination Survey. (n.d.). https://www.cdc.gov/nchs/nhanes/about_nhanes.htm

E. Sources

Nelson, M. (2024, April 11). D210: Representation and Reporting. Unpublished manuscript, Western Governors University.