**Texas COIVD-19 Vaccine Allocation and Distribution Analysis:**

Weekly and Daily reports of COVID-19 vaccine administration in Texas

Tyler Mann

4/1/2021

Macro & SQL Assignment

University of Dallas

Table of Contents

Contents

[Introduction 2](#_Toc68151282)

[Data 3](#_Toc68151283)

[Data Dictionaries 3](#_Toc68151284)

[Analysis and Results 4](#_Toc68151285)

[Conclusion 9](#_Toc68151286)

# Introduction

The Coronavirus, also known as COIVD-19, has changed the way we do many things in our lives. The ways that we shop, work, go to church, attend college, etc. has changed dramatically since early last year. COVID-19 is caused by infection with a new coronavirus and flu is caused by infection with influenza viruses. Because of the dangers of this virus, we were glued to our televisions, our radios, and social media to figure out the impact that COIVD-19 would have in our society. There’s data everywhere giving statistics and projections for where we are now and where we are headed. Using SAS Macros and different PROC SQL procedures can break down that data further to see where we have been and where we are headed.

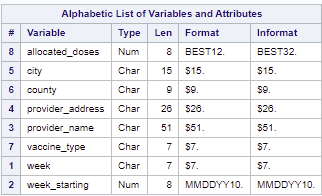
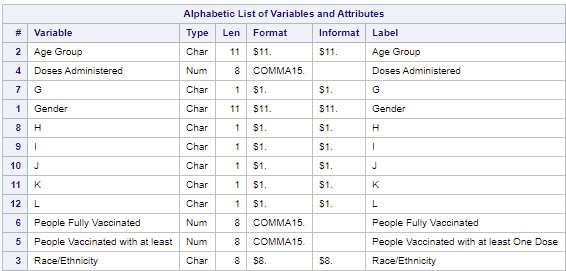
The main purpose of the dataset is to provide weekly data on the allocation and administration data of COIVD-19 vaccines across the state of Texas. There is another table that breaks down vaccinations based on race/ethnicity and the county they reside in. These datasets are a glimpse into recovery from COVID-19, showing the progress that is made each week to vaccinate Texas one week at a time.

Data is a part of our lives each day, even when it is not noticed by others. It is all around us and being broadcasted to us 24/7. Companies are learning the value of data and are looking for more efficient ways to compile that data. That is why the duties of a Data Analyst will continue to change overtime. The SAS basics of understanding SAS Macros and SQL are used widely and will assist in filtering the data to get the needed results. This report will go over an analysis of different variables to provide compelling results to better understand vaccine distribution in the state of Texas.

# Data

The datasets that I have named WORK.IMPORT and WORK.ALLOCATIONS come from the Kaggle website. They are based on the weekly and daily reports of COVID-19 vaccine administration in Texas. The datasets are maintained and updated by Rajkumar Sengottuvel is a Senior Web Technologist at Oracle. The COVID-19 Vaccine Data by County.xlsx is provided by Texas Department of State Health Services daily that contains information about total doses administered in each county. It contains 12 columns with 102 rows, further information is in the chart to the left. The weekly\_allocation.csv is scrapped and cleaned from the weekly doses allocation report provided by the Texas Department of State Health Services. It contains 8 columns and 8,449 rows; further information is in the chart to the right. The data was collected because of its relevance on today and to see the impact and how the state of Texas is distributing the vaccine in the counties.

# Data Dictionaries



# Analysis and Results

Appendix A:

1. These files were first downloaded, and then uploaded into SAS Studio.
2. The datasets were studied, and observations were pulled from both datasets.
3. PROC CONTENTS was run to notice variables and columns from the datasets.
4. Columns were now selected that provided information that was needed.
5. Reports were created and the data was analyzed. Results above.

Appendix B:

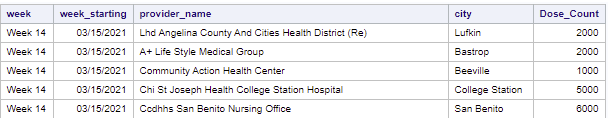
Using the SELECT statement and creating a table query result from the code.

Appendix C:

Created a new table by defining columns.

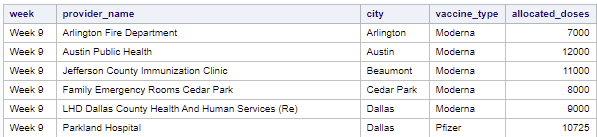
Appendix D:

Created a new column of Dose\_Count for week 14.



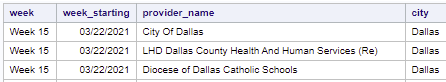
Appendix E:

Sorted the data using a WHERE clause for allocated\_doses > 5000 and ORDERED BY one and 3 descending.



Appendix F:

Retrieved the rows that satisfied a condition.

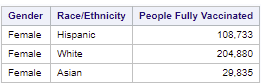


Appendix G:

Using a FULL JOIN to put together two different tables by combining them. It shows all rows and columns.

Appendix H:

Combine tables using the set operator running multiple tables before producing the dataset with a UNION ALL statement.



Appendix I:

Using the summary function to get the MAX, MIN, and AVERAGE.

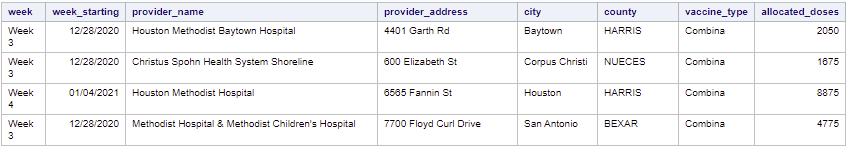


Appendix J:

Changing the names of the columns and grouping the data.

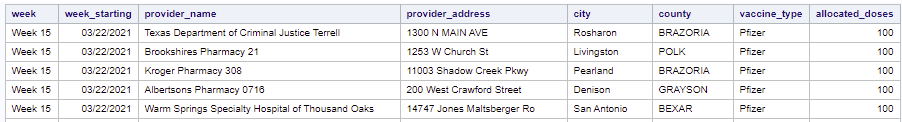
Appendix K:

Filtering grouped data.



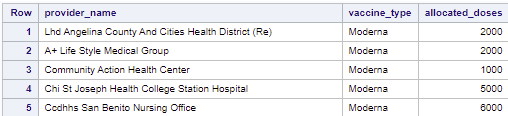
Appendix L:

Subset data by using non-correlates subqueries (HAVING clause) for those that are < 5000.



Appendix M:

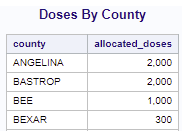
Subset data by using correlated subqueries.



Appendix N:

Appendix O:

Insert rows into tables which I looked at allocated doses based on County.



Appendix P:

Updated the values from the previous table to simulate if the doses were increased x3 for a projection.

Appendix Q:

Delete Rows.

Appendix R:

Alter column attributes which I created a table and created a new label called future\_allocation\_projection.

Appendix S:

Create an index.

Appendix T:

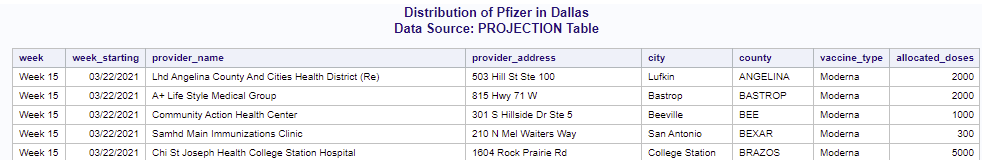
Delete a table.

Appendix U:

Use the DESCRIBE TABLE statement.

Appendix V:

User-defined and automatic MACRO variables, also entered comments into MACRO and used the symbolgen function.

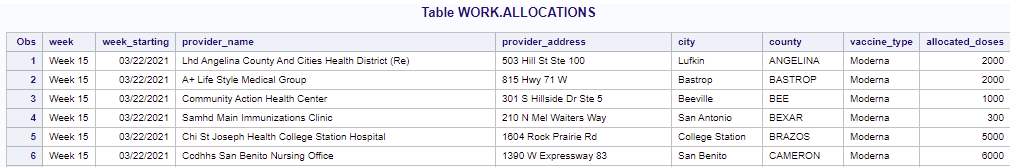


Appendix W:

Definition

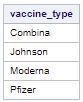
Appendix X:

Use of the %GLOBAL statement and the MACRO statement within a printable. Also using the MPRINT option



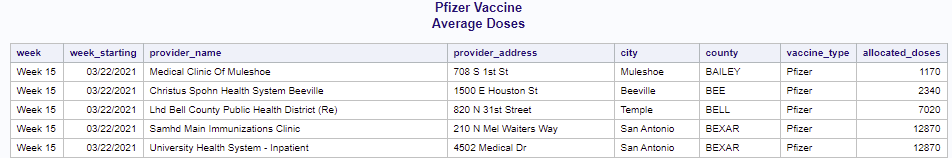
Appendix Y:

Use the INTO clause of the SELECT statement in SQL and pass information into a MACRO using parameters.



Appendix Z:

Using the %LOCAL statement.



Appendix AA:

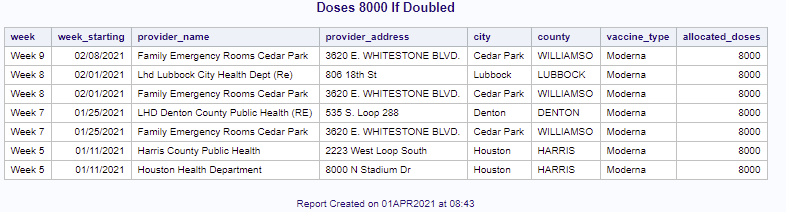
The SYMPUTX function since the SYMGET function is no longer available, also using MACRO quoting functions.

Appendix BB:

Definition

Appendix CC:

Use MACRO evaluation functions and %sysevalf to create the allocated doses if they were to be doubled.



Appendix DD:

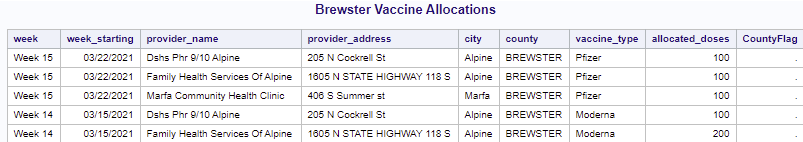
Deleting a column out of the global symbol table.

Appendix EE:

Use the %PUT statement to tack problems within the statement.

Appendix FF:

Using the %IF-%THEN-%ELSE and %DO loop.



Appendix GG:

Indirect reference to MACRO variables.

Appendix HH:

Graphs for Powerpoint Presentation.

# Conclusion

SAS and the MACRO language provide great insights to the COVID-19 datasets to see the distribution of the vaccine by county in the state of Texas. The reports can be used and built up over time for better distribution of the vaccine if it will be needed each year. The use of the MACRO language can assist for more rapid changes to the data. These reports provide insight about those counties that may need more of the vaccine, or areas that are not getting the vaccine at all. The data also shows those with the 1st shot only, so those numbers can be monitored to ensure they come back for the 2nd shot for full vaccination.

/\*Appendix A: Download Dataset\*/

%web\_drop\_table(WORK.ALLOCATIONS);

FILENAME REFFILE '/home/u49574186/EMC1V2/weekly\_allocations.csv';

PROC IMPORT DATAFILE=REFFILE

DBMS=CSV

OUT=WORK.ALLOCATIONS;

GETNAMES=YES;

RUN;

PROC CONTENTS DATA=WORK.ALLOCATIONS; RUN;

%web\_open\_table(WORK.ALLOCATIONS);

/\*Download Dataset\*/

%web\_drop\_table(WORK.IMPORT);

FILENAME REFFILE '/home/u49574186/EMC1V2/COVID-19 Vaccine Data by County.xlsx';

PROC IMPORT DATAFILE=REFFILE

DBMS=XLSX

OUT=WORK.IMPORT;

GETNAMES=YES;

SHEET="By Age, Gender, Race";

RUN;

PROC CONTENTS DATA=WORK.IMPORT; RUN;

%web\_open\_table(WORK.IMPORT);

/\*Appendix B: 1. Use the SELECT statement / 6. Creating a table from a query result\*/

PROC SQL;

CREATE TABLE WORK.QUERY AS

SELECT \*

FROM WORK.ALLOCATIONS;

RUN;

QUIT;

PROC DATASETS NOLIST NODETAILS;

CONTENTS DATA=WORK.QUERY OUT=WORK.details;

RUN;

PROC PRINT DATA=WORK.details;

RUN;

/\*Appendix C: 2. Create a table by defining columns\*/

PROC SQL;

CREATE TABLE WORK.QUERY AS

SELECT week, week\_starting, provider\_name

FROM WORK.ALLOCATIONS;

RUN;

QUIT;

/\*Appendix D: 3. Creating a new column\*/

PROC SQL;

SELECT week, week\_starting, provider\_name, city, allocated\_doses AS Dose\_Count

FROM WORK.ALLOCATIONS

WHERE week='Week 14';

QUIT;

/\*Appendix E: 4. Sort data Subsetting calculated values\*/

PROC SQL;

SELECT week, provider\_name, city, vaccine\_type, allocated\_doses

FROM WORK.ALLOCATIONS

WHERE allocated\_doses > 5000

ORDER BY 1 desc, 3;

QUIT;

/\*Appendix F: 5. Retrieve rows that satisfy a condition\*/

PROC SQL;

SELECT week, week\_starting, provider\_name, city

FROM WORK.ALLOCATIONS

WHERE city='Dallas';

QUIT;

/\*Appendix G: 7. Join tables\*/

PROC SQL;

CREATE TABLE WORK.FEMALE AS

SELECT \*

FROM WORK.IMPORT

WHERE Gender="Female";

QUIT;

PROC SQL;

CREATE TABLE WORK.MALE AS

SELECT \*

FROM WORK.IMPORT

WHERE Gender="Male";

QUIT;

PROC SQL;

SELECT \*

FROM WORK.MALE FULL JOIN WORK.FEMALE

ON MALE.Gender = FEMALE.Gender;

QUIT;

/\*Appendix H: 8. Combine tables using set operators - (ex: union, outer join, except, or intercept)\*/

PROC SQL;

SELECT \*

FROM WORK.IMPORT;

QUIT;

PROC SQL;

CREATE TABLE WORK.FEMALE2 AS

SELECT \*

FROM WORK.IMPORT

WHERE Gender="Female";

QUIT;

PROC SQL;

CREATE TABLE WORK.MALE2 AS

SELECT \*

FROM WORK.IMPORT

WHERE Gender="Male";

QUIT;

PROC SQL;

SELECT Gender, "Race/Ethnicity"n, "People Fully Vaccinated"n

FROM WORK.MALE2

UNION ALL

SELECT Gender, "Race/Ethnicity"n, "People Fully Vaccinated"n

FROM WORK.FEMALE2

ORDER BY Gender;

QUIT;

/\*Appendix I: 9. Summary Functions (Max), (Min), (Avg)\*/

PROC SQL;

SELECT MAX(allocated\_doses) AS MaxEst\_dosage,

MIN(allocated\_doses) AS MinEst\_dosage,

AVG(allocated\_doses) AS AvgEst\_dosage

FROM WORK.ALLOCATIONS;

quit;

/\*Appendix J: 10. Grouping the data and Changing the names of the columns\*/

PROC SQL;

SELECT provider\_name "Local Facility",

city label="Location", county,

allocated\_doses format=comma8.0

FROM WORK.ALLOCATIONS;

WHERE week='week 1';

quit;

/\*Appendix K: 11. Filter grouped data\*/

PROC SQL;

SELECT \*

FROM WORK.ALLOCATIONS

GROUP BY county

ORDER BY vaccine\_type;

quit;

/\*Appendix L: 12. Subset data by using non-correlates subqueries (HAVING clause)\*/

PROC SQL;

SELECT \*

FROM WORK.ALLOCATIONS

WHERE vaccine\_type='Pfizer'

GROUP BY week

HAVING allocated\_doses < 5000

ORDER BY allocated\_doses;

quit;

/\*Appendix M: 13. Subset data by using correlated subqueries\*/

PROC SQL;

SELECT AVG(allocated\_doses) AS Average\_Doses

FROM WORK.ALLOCATIONS;

QUIT;

PROC SQL;

SELECT \*

FROM WORK.ALLOCATIONS

WHERE allocated\_doses > 10000

ORDER BY city;

QUIT;

PROC SQL NUMBER;

SELECT provider\_name, vaccine\_type, allocated\_doses

FROM WORK.ALLOCATIONS

WHERE allocated\_doses > (SELECT AVG(allocated\_doses) AS Average\_Doses

FROM WORK.ALLOCATIONS DESC);

QUIT;

/\*Appendix N: 14. Reference an in-line view with other views or tables (multiple tables)\*/

PROC SQL NUMBER;

SELECT \*

FROM (SELECT MAX(allocated\_doses) AS MAX\_Doses

FROM WORK.ALLOCATIONS DESC);

QUIT;

/\*Appendix O: 15. Insert rows into tables\*/

PROC SQL;

CREATE TABLE WORK.COVID LIKE WORK.ALLOCATIONS;

PROC SQL;

TITLE "Doses By County";

INSERT INTO WORK.COVID (county, allocated\_doses)

SELECT county, allocated\_doses

FROM WORK.ALLOCATIONS

WHERE allocated\_doses < 5000;

QUIT;

PROC SQL;

TITLE "Doses By County";

SELECT county, allocated\_doses format=comma8.0

FROM WORK.COVID;

QUIT;

/\*Appendix P: 16. Update data values in a table\*/

PROC SQL;

UPDATE WORK.COVID

SET allocated\_doses=allocated\_doses\*3;

TITLE "Doses By County";

SELECT county, allocated\_doses format=comma10.0

FROM WORK.COVID;

QUIT;

/\*Appendix Q: 17. Delete rows\*/

PROC SQL;

DELETE FROM WORK.COVID

WHERE county LIKE 'HARRIS%';

QUIT;

/\*Appendix R: 18. Alter columns attributes\*/

PROC SQL;

CREATE TABLE WORK.PROJECTION LIKE WORK.COVID;

INSERT INTO WORK.PROJECTION

SELECT \*

FROM WORK.COVID;

QUIT;

PROC SQL;

CREATE TABLE WORK.PROJECTION AS

SELECT \*, allocated\_doses\*3 AS future\_projection

label='future\_allocation\_projection'

format=comma10.0

FROM WORK.PROJECTION;

QUIT;

/\*Appendix S: 19. Create an index\*/

PROC SQL;

CREATE INDEX provider\_name

ON WORK.COVID(provider\_name);

QUIT;

/\*Appendix T: 20. Delete a table\*/

PROC SQL;

DROP TABLE WORK.PROJECTION

QUIT;

/\*Appendix U: 21. Use the DESCRIBE TABLE statement\*/

PROC SQL;

describe table WORK.ALLOCATIONS;

quit;

/\*Appendix V: 22. User-defined and automatic MACRO variables / 30. Insert comments into MACRO / 33. Use MACRO functions / 38. Use SYMBOLGEN\*/

OPTIONS SYMBOLGEN;

%LET city=Dallas;

%LET vaccine\_type=Pfizer;

PROC SQL;

TITLE "Distribution of Pfizer in %sysfunc(propcase(&city))";

TITLE2 "Data Source: %substr(&syslast, 6) Table";

FOOTNOTE "Created on &sysday, &sysdate9";

%\*Showing the number of allocated doses of Moderna.;

SELECT \*

FROM WORK.ALLOCATIONS;

WHERE city=&city and vaccine\_type=&vaccine\_type;

RUN;

TITLE;FOOTNOTE;

OPTIONS NOSYMBOLGEN;

%put \_automatic\_;

/\*Appendix W: 23. Define MACRO variables: Tools that enable you to dynamically modify the text in a SAS program through symbolic substitution.\*/

/\*Appendix X: 24. Use %GLOBAL statement / 29. Use the MACRO statement\*/

%MACRO printtable(city=WORK.ALLOCATIONS);

TITLE "Table &city";

FOOTNOTE "Report Created on %sysfunc(today(), date9.) at %sysfunc(time(), timeampm.)";

PROC PRINT DATA=&city;

RUN;

%MEND printtable;

QUIT;

OPTIONS MPRINT;

%printtable(city=WORK.ALLOCATIONS);

OPTIONS NOMPRINT;

/\*Appendix Y: 25. Use INTO clause of the SELECT statement in SQL / 31. Pass information into a MACRO using parameters\*/

PROC SQL;

SELECT DISTINCT vaccine\_type

INTO :vaccinelist separated by ", "

FROM WORK.IMPORT;

QUIT;

%put &=vaccinelist;

%put &=sqlobs;

%PUT \_ALL\_;

/\*Appendix Z: 26. %LOCAL statement\*/

%MACRO create;

DATA \_NULL\_;

SET WORK.ALLOCATIONS;

CALL symputx('Type', 'vaccine\_type', 'L');

RUN;

%MEND;

%create

%PUT &=AvgDoses;

TITLE "&vaccine\_type Vaccine";

TITLE2 "Average Doses";

PROC PRINT DATA=WORK.ALLOCATIONS noobs;

WHERE vaccine\_type="&vaccine\_type";

RUN;

TITLE;

/\*Appendix AA: 27. The SYMPUTX function since SYMGET doesn't work / 34. Use MACRO quoting functions\*/

OPTIONS MPRINT MLOGIC;

DATA \_NULL\_;

SET WORK.ALLOCATIONS;

CALL symputx("AvgDoses", 'doses\_mean');

CALL symputx("MaxDoses", 'doses\_max;');

RUN;

%LET AllocateAvgDoses=%SUPERQ(AvgDoses);

%LET AllocateMaxDoses=%SUPERQ(MaxDoses);

%PUT MACRO variable AvgDoses is &AllocateAvgDoses;

%PUT MACRO variable MaxDoses is &AllocateMaxDoses;

OPTIONS NOMPRINT NOMLOGIC;

/\*Appendix BB: 28. Define a MACRO: A macro name must be a SAS name, which you supply; you cannot use a text expression to generate a macro name in a %MACRO statement. In addition, do not use macro reserved words as a macro name.\*/

/\*Appendix CC: 35. Use MACRO evaluation functions\*/

%LET allocated\_doses=4000;

TITLE "Doses %sysevalf(&allocated\_doses\*2) If Doubled";

FOOTNOTE "Report Created on &sysdate9 at &systime";

PROC PRINT DATA=WORK.ALLOCATIONS NOOBS;

WHERE allocated\_doses=&allocated\_doses\*2;

RUN;

TITLE;FOOTNOTE;

/\*Appendix EE: 39. Use the %PUT statement to track problems\*/

%PUT \_USER\_; \*lists all macro variables created by the user or application\*

%put NOTE: &=path;

%put ERROR- Course files are in &path;

%PUT \_ALL\_;

/\*Appendix FF: 32. Using the %IF-%THEN-%ELSE MACRO statements / 42. %DO loop\*/

%LET foot=&county;

%LET county=Brewster;

data &county;

set WORK.ALLOCATIONS end=lastrow;

where upcase(County)="%upcase(&county)";

retain CountyFlag;

if county="Brewster" then CountyFlag=1;

if lastrow then do;

if CountyFlag=1 then do;

call symputx("foot", "&County Largest County in Texas");

end;

else do;

call symputx("foot", "&County Not the largest County in Texas");

end;

end;

run;

TITLE "&County Vaccine Allocations";

FOOTNOTE "&foot";

PROC PRINT DATA=&County NOOBS;

RUN;

TITLE; FOOTNOTE;

/\*Appendix GG: 41. Indirect reference to MACRO variables\*/

%PUT \_global\_;

%LET county="Dallas";

%LET n=1;

%PUT &&county&n;

/\*Appendix DD: To delete columns out of the global symbol table\*/

%SYMDEL allocated\_doses

# Work Cited

“COVID-19 Vaccine Information.” *Texas Department of State Heealth Services*, [www.dshs.texas.gov/covidvaccine. Accessed 1 Apr. 2021](http://www.dshs.texas.gov/covidvaccine.%20Accessed%201%20Apr.%202021).

Sengottuvel, Rajkumar. “TEXAS COVID-19 Vaccine Allocation and Distribution.” *Kaggle*, 31 Mar. 2021, www.kaggle.com/rajsengo/texas-covid19-vaccine-allocation.