Analyzing Vaccine Uptake in Texas

true

2021-09-02

Analysis Setup

Before we start building out our reproducible analysis, let's go ahead and make sure any R packages are loaded and installed properly. The code to install necessary packages and load them can be viewed by clicking on the "Show Code" arrow.

```
knitr::opts_chunk$set(warning = FALSE, message = FALSE)
# In case these aren't installed already from the prework, uncomment the next three lines and run them.
# install.packages(c("rmarkdown", "tidyverse", "tidycensus", "distill", "devtools"))
# install.packages(c("janitor", "readxl"))

library(janitor)       # Package with useful + convenient data cleaning functions
library(readxl)        # Package with useful functions for working with Excel Files
library(tidyverse)     # Core Set of R Data Science Tools (dplyr, ggplot2, tidyr, readr, etc.)
```

Analysis

Analysis Goal

The goal for our analysis today is to take data from the state and identify the top 10 counties for vaccine uptake based on the number of fully vaccinated persons in a county as a percent of all eligible residents in a county.

Importing Our Data

This data comes from the Texas Department of State Health Services and contains various reporting metrics related to vaccine uptake across the state of Texas, which can be found on this page. They use it for their own interactive dashboard of vaccine uptake in Texas.[1] For this analysis, we'll focus on the data in the "By County" tab, which we'll download from the state's website and import directly into this analysis. Once downloaded, we'll use the `read_excel()` function from a package called `readxl` to read in the data straight from DSHS so we can do our analysis and make a chart.

The `read_excel()` function comes from the `readxl` package that was loaded when we ran `library(readxl)` in the Analysis Setup section (code chunk at lines 31:40 in the RMarkdown document).

[1] The link for this dashboard is https://tabexternal.dshs.texas.gov/t/THD/views/COVID-19VaccineinTexasDashboard/Summary?:origin=card_share_link&:embed=n

Step 1: Download The Data

We'll download the data using the code below. Once downloaded, we can start working with our data.

```
vax_url <- "https://dshs.texas.gov/immunize/covid19/COVID-19-Vaccine-Data-by-County.xls"
download.file(url=vax_url, destfile = "raw_data/tx_vaccine_uptake.xlsx")
```

Step 2: Import the Downloaded Data

Now that we've downloaded the data, we can import the data and see what we've got. **Notice** two things about the vaccine uptake data that we imported the downloaded from DSHS:

1. We assigned it to a data object called `vaccine_data_raw` and then used that data object again in the `glimpse()` function to preview our imported data. We'll use that `vaccine_data_raw` object in future steps of the analysis to do things like create a subset of our raw data.
2. A lot of our data that should be numeric is classified by as character data (hence, the `<chr>` tag in our previewed data). We'll need to fix this at some point, but not right now.

```
vaccine_data_raw <- readxl::read_excel(path = "raw_data/tx_vaccine_uptake.xlsx", sheet = "By County") |>
  janitor::clean_names() # This function makes column headers machine readable

dplyr::glimpse(vaccine_data_raw) # glimpse() lets you preview a data object
```

```
## Rows: 259
## Columns: 13
## $ county_name                                <chr> "Texas", "Federal Long-T~
## $ public_health_region_phr                   <chr> "--", "--", "--", "4/5N"~
## $ total_doses_allocated                      <dbl> 27513735, 720525, 550970~
## $ vaccine_doses_administered                 <chr> "29181336", "--", "--", ~
## $ people_vaccinated_with_at_least_one_dose   <chr> "16531113", "--", "--", ~
## $ people_fully_vaccinated                    <chr> "13809684", "--", "--", ~
## $ population_12                              <chr> "24068624", "--", "--", ~
## $ population_16                              <chr> "22421178", "--", "--", ~
## $ population_65                              <chr> "3734229", "--", "--", "~
## $ population_phase_1a_healthcare_workers     <chr> "1241763", "--", "--", "~
## $ population_phase_1a_long_term_care_residents <chr> "271813", "--", "--", "8~
## $ population_16_64_any_medical_condition     <chr> "8302974", "--", "--", "~
## $ population_education_and_child_care_personnel <chr> "850859", "--", "--", "1~
```

Working With Data

Step 1: Reclassify Numeric Data & Create a Percent Column

Looking at the preview of our data, it looks like we can use three columns to complete our analysis: `county_name`, `people_fully_vaccinated`, and `population_12` (since only persons aged 12 and older are eligible for the vaccine). So we'll use a function called `mutate()`, which lets add columns or change things about columns that already exist. For our analysis, we'll use `mutate()` in two ways: 1. to reclassify some of that character data referenced above as numeric data. 2. add a column called `pct_uptake` where we divide the number of fully vaccinated residents in a county (`people_fully_vaccinated`) by the number of all eligible residents in a county (`population_12`).

Lastly, we'll want to filter out values from the `county_name` column that are not actual counties. For example, they have an entry in that column for the entire state along with an "Other" column. If we want to rank counties, they can't be in here.

```
vaccine_uptake <- vaccine_data_raw |>
  select(county_name, people_fully_vaccinated, population_12) |> # Pull the three columns we need
  mutate(people_fully_vaccinated = as.numeric(people_fully_vaccinated), # Reclassify Character Data As
         population_12 = as.numeric(population_12)) |>
  mutate(pct_uptake = round(people_fully_vaccinated/population_12, digits = 4)) |>  # Create PCT Column
  filter(county_name!="Texas",                                          # Filter out 'Texas' value
         county_name!="Other",                                          # Filter our 'Other' value
         county_name!="Federal Long-Term Care Vaccination Program",     # Filter our Federal Programs
         county_name!="Federal Pharmacy Retail Vaccination Program")

glimpse(vaccine_uptake) # Preview Our New Data
```

```
## Rows: 254
## Columns: 4
## $ county_name             <chr> "Anderson", "Andrews", "Angelina", "Aransas", ~
## $ people_fully_vaccinated <dbl> 18564, 6377, 30421, 11743, 3354, 579, 19685, 1~
## $ population_12           <dbl> 50661, 14863, 72486, 20834, 7388, 1584, 41738,~
## $ pct_uptake              <dbl> 0.3664, 0.4291, 0.4197, 0.5636, 0.4540, 0.3655~
```

Notice that our columns which previous had `<chr>` tags now have `<dbl>` tags which let us know it's numeric.

Step 2: Rank Counties & Find The Top 10

In the last step, we took a subset of the raw data we downloaded from DSHS and created a column that calculates the percent of uptake. That subset now lives in a data object we called `vaccine_uptake`. Now, we can use the percent of uptake in each county to rank them and determine today's top 10 counties for vaccine uptake. To rank them, we'll use a function called `dense_rank()`, which allows us to rank counties. The only problem with `dense_rank()` is that it automatically ranks by lowest number. In our case, higher is better. So we wrap our `pct_uptake` column in a function called `desc()` which ranks in descending order from the default.

```
ranked_uptake <- vaccine_uptake |>                      # Create New Object
  mutate(uptake_rank = dense_rank(desc(pct_uptake))) |> # Create Rank Column
  filter(uptake_rank <= 10) |>                          # Filter only for Top 10 counties
  arrange(uptake_rank)                                  # Sort by rank

glimpse(ranked_uptake)
```

```
## Rows: 10
## Columns: 5
## $ county_name             <chr> "Presidio", "Webb", "Starr", "Cameron", "El Pa~
## $ people_fully_vaccinated <dbl> 4988, 178642, 40792, 253732, 505558, 499362, 3~
## $ population_12           <dbl> 5392, 215919, 49864, 339698, 688420, 681678, 4~
## $ pct_uptake              <dbl> 0.9251, 0.8274, 0.8181, 0.7469, 0.7344, 0.7325~
## $ uptake_rank             <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
```

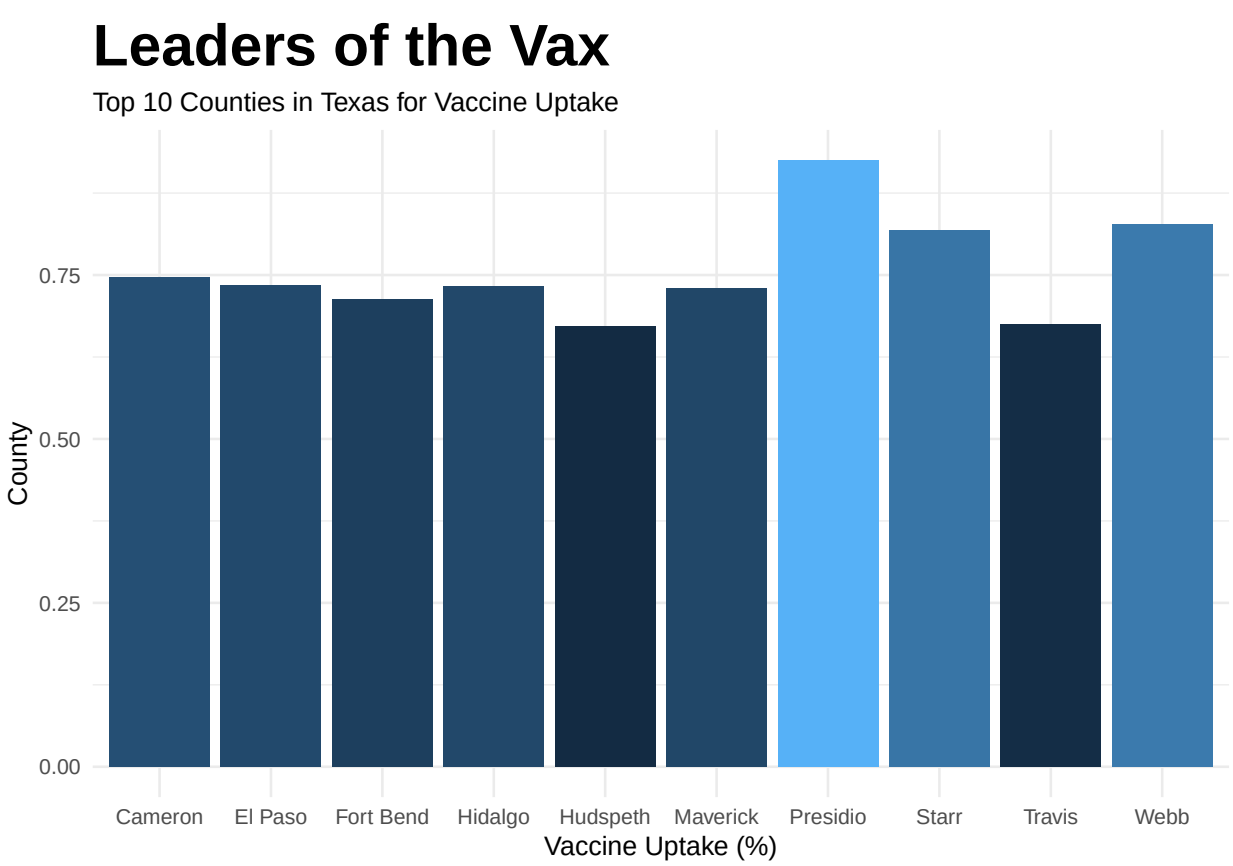
Visualizing Our Data

Lastly, we'll take our ranked data and draw a bar chart of counties, sorted by their uptake percentage.

Step 1: Visualize Our Ranked Data

```
uptake_chart <- ggplot(ranked_uptake) +
                aes(x=county_name, y = pct_uptake, fill=pct_uptake) +
                geom_col() +
                labs(title = "Leaders of the Vax",
                     subtitle = "Top 10 Counties in Texas for Vaccine Uptake",
                     x = "Vaccine Uptake (%)",
                     y = "County") +
                theme_minimal(base_size = 10) +
                theme(plot.title = element_text(size = 22, face = "bold"),
                      legend.position = "none")

uptake_chart
```



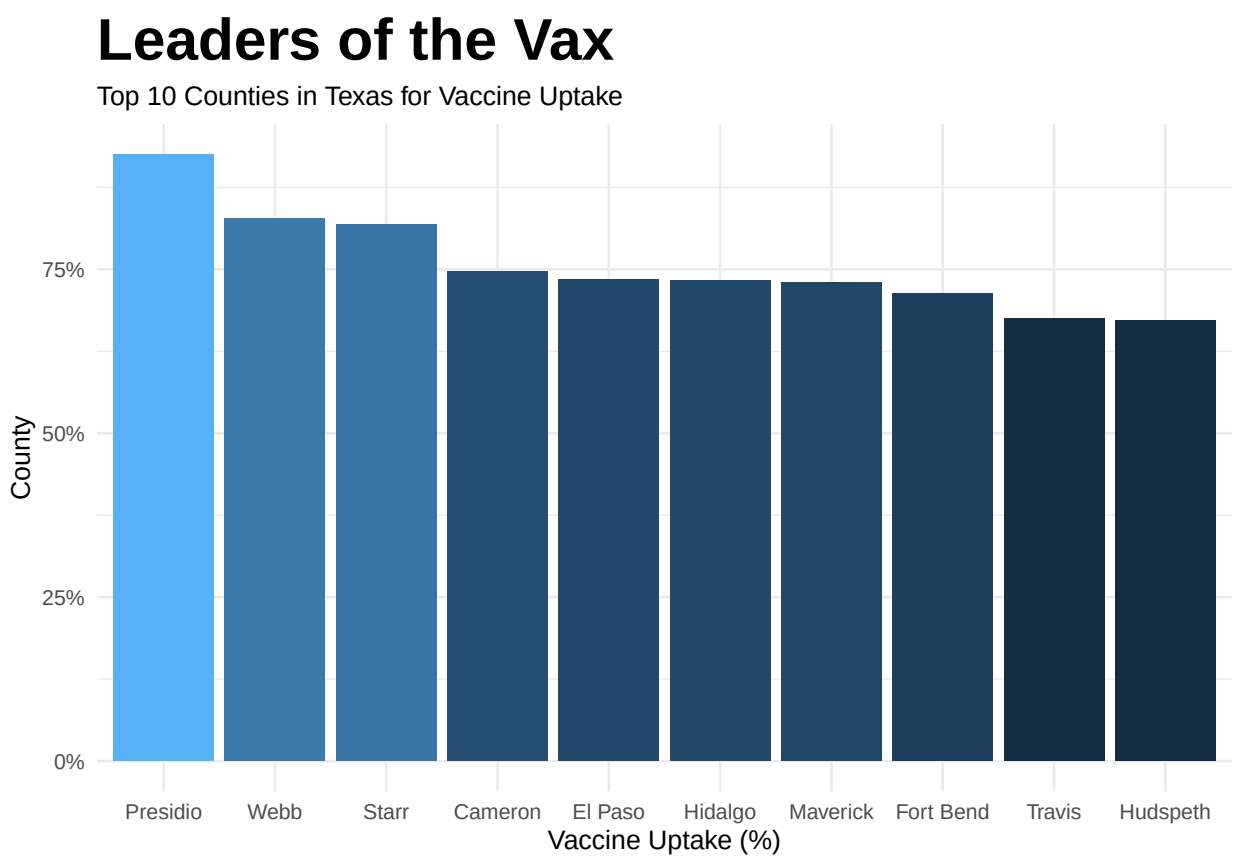
Step 2: Sort our Visualization

This chart looks okay, but it's not quite what we want. To make it easier to read, we'd want to convert the y-axis to a percent instead of the decimal and we'd want to sort the counties by highest uptake. So let's do that real quick using a function called `reorder()` to help with the sorting of bars/counties and a function called `scale_y_continuous()` to help with adjusting the labels on the y-axis.

```
uptake_sorted <- ggplot(ranked_uptake) +
                 aes(x=reorder(county_name, uptake_rank), y = pct_uptake, fill=pct_uptake) +
```

```
                geom_col() +
                scale_y_continuous(labels = scales::percent) +
                labs(title = "Leaders of the Vax",
                     subtitle = "Top 10 Counties in Texas for Vaccine Uptake",
                     x = "Vaccine Uptake (%)",
                     y = "County") +
                theme_minimal(base_size = 10) +
                theme(plot.title = element_text(size = 22, face = "bold"),
                      legend.position = "none")

uptake_sorted
```



Exporting Our Data

Step 1: Export Our Chart

```
ggsave("vaccine_uptake_chart.png", uptake_sorted, dpi = 300, width = 10, height = 6, bg = "white")
```

Step 2: Export Our Data

```
write_csv(ranked_uptake, "vaccine_uptake_top_10.csv")
```