ADAM WILLARD

Data Analytics

Portfolio

PROJECTS

GAMECO

Analyze global videogame retail sales using Excel

UNDERSTANDING FLU SEASON

Identify seasonal flu staffing needs throughout the USA, using Excel and Tableau

ROCKBUSTER STEALTH

Movie rental insights and customer analysis using Tableau and SQL

INSTACART BASKET ANALYSIS

Sales data analysis and customer profiling using Python and Jupyter

ESSENTIAL WORKER AFFORDABLE HOUSING

Analysis of affordable housing availability using webscraping in Python, and Tableau operational dashboards

This portfolio contains only excerpts from each project completed. Full reports and analyses are available as links at the bottom of each slide.

GAMECO

OBJECTIVE

Develop a current understanding of the global retail videogame sales market, to inform GameCo's efforts to increase market share.

TOOLS & SKILLS

Excel PowerPoint

- Data quality, integrity, and consistency checks
- Data cleaning
- Pivot tables (data grouping & summarizing)
- Descriptive analysis
- Excel visualizations
- Reporting in PowerPoint

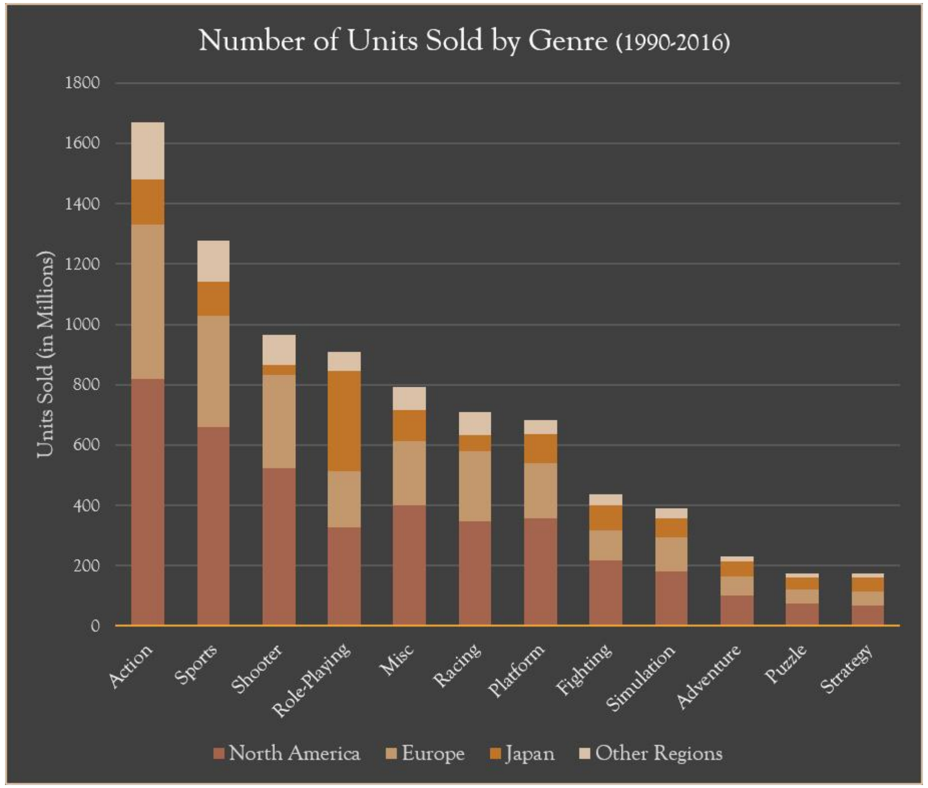
DATA

The data is made publicly available by VGChartz. It covers historical retail sales of videogames for games that sold more than 100,000 copies, until 2016.

CONSTRAINTS

The data available only has figures for numbers of units sold and does not include the price per unit. Additionally, the dataset doesn't include games sold on digital platforms, which accounts for the apparent steep decline in number of games sold after 2009.

INITIAL ANALYSIS

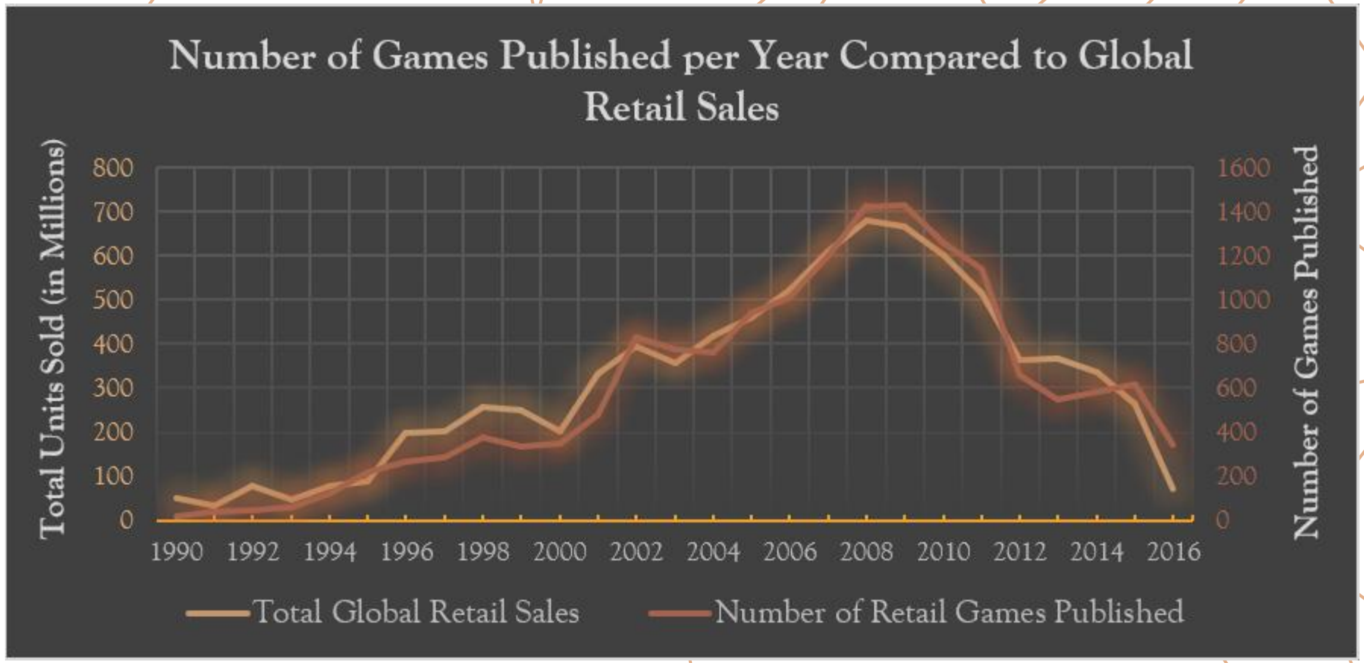


GENRE SALES ANALYSIS BY GEOGRAPHICAL REGION

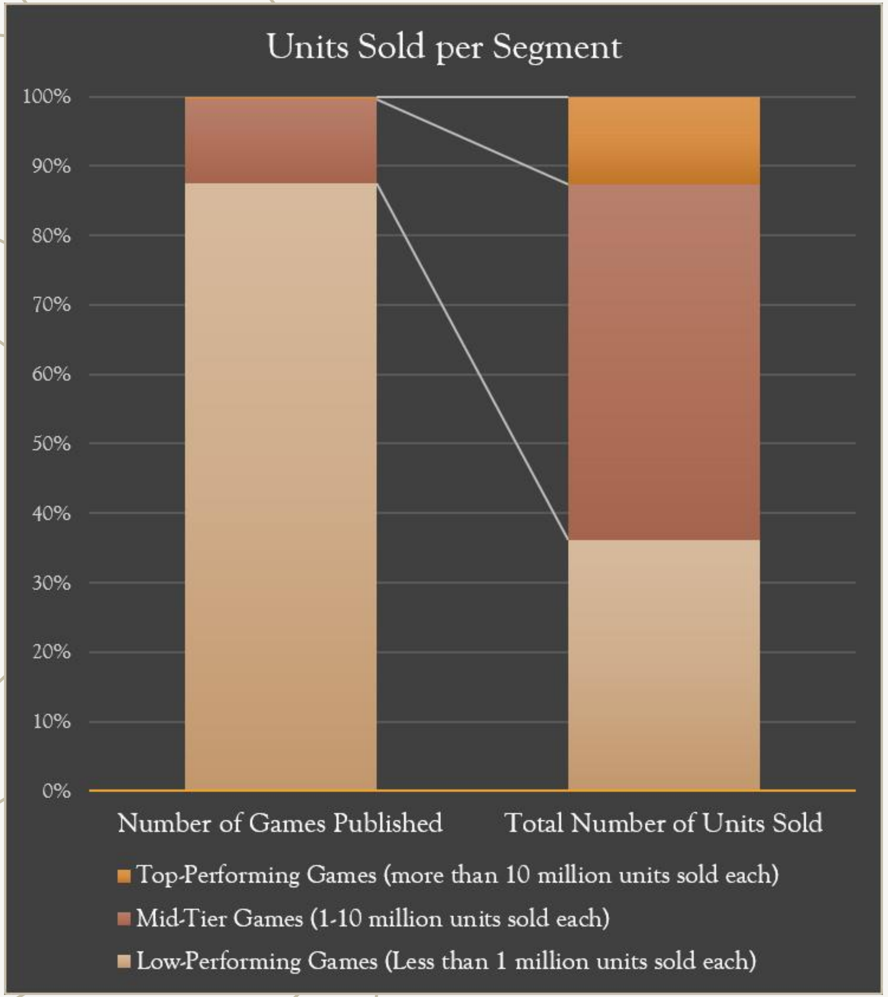
This stacked bar chart shows the top-performing genres as well as their portion of sales per major geographical region.

TIME SERIES ANALYSIS OF GAMES PUBLISHED COMPARED TO GAMES SOLD

This line graph shows that retail games sold correlates very closely with number of games published. Beginning in 2009 there is a simultaneous decline in retail games sold and games published in retail markets.



FINDINGS



NUMBER OF GAMES SOLD PER MARKET SEGMENT

These 100% stacked bar charts show that less than 15% of games published account for over 60% of all units sold. Game studios that can develop AAA games have the potential to demonstrate the biggest growth in market share.

RECOMMENDATIONS

In observation of the overall retail games publishing and sales trends of recent years, any upcoming game studio needs to focus on digital sales platforms.

The consistently best-selling genres are Action, Sports, and Shooter games across the North American and European geographic regions. However, role-playing games have the strongest appeal for the Japanese market.

The full analysis and results report is available as a Powerpoint slideshow.

UNDERSTANDING FLU SEASON

OBJECTIVE

Identify geographic and seasonal trends for annual influenza outbreaks in the USA. Provide tools for a medical staffing agency to identify where and when to allocate additional medical support.

TOOLS & SKILLS

Excel

+ a b | e a u

- Data research project design
- Data profiling and cleaning
- Data integration and transformation
- Statistical hypothesis testing
- Geographic visualizations and time-series forecasting
- Interactive visualizations and storytelling in Tableau

DATA

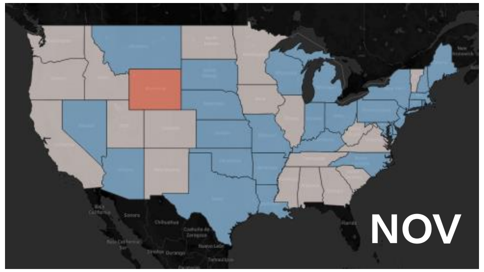
Population data came from the US Census Bureau. Flu death reporting and survey of flu shot rates came from the CDC. Flu lab tests and flu-like illnesses clinical visits data came from the CDC's Fluview site.

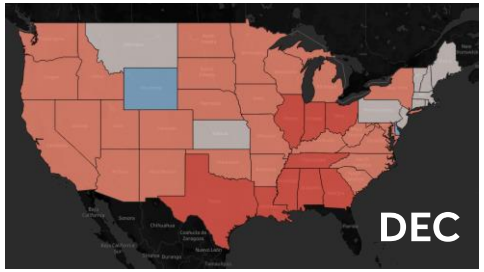
CONSTRAINTS

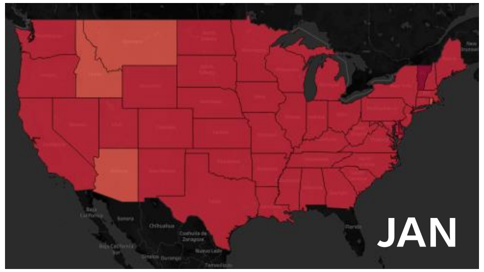
As the data is sourced from government sources, the quality and integrity is high. However, the data used for this project was all gathered before covid-19 and therefore it can be expected that many aspects of flu season were upset during that time and potentially changed since then. Additionally, the vaccination data was sourced from children and may not be highly representative of adult populations.

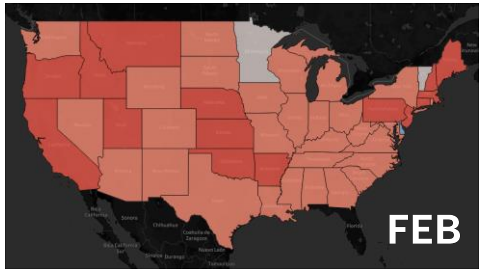
INITIAL ANALYSIS

AVERAGE MONTHLY FLU DEATHS







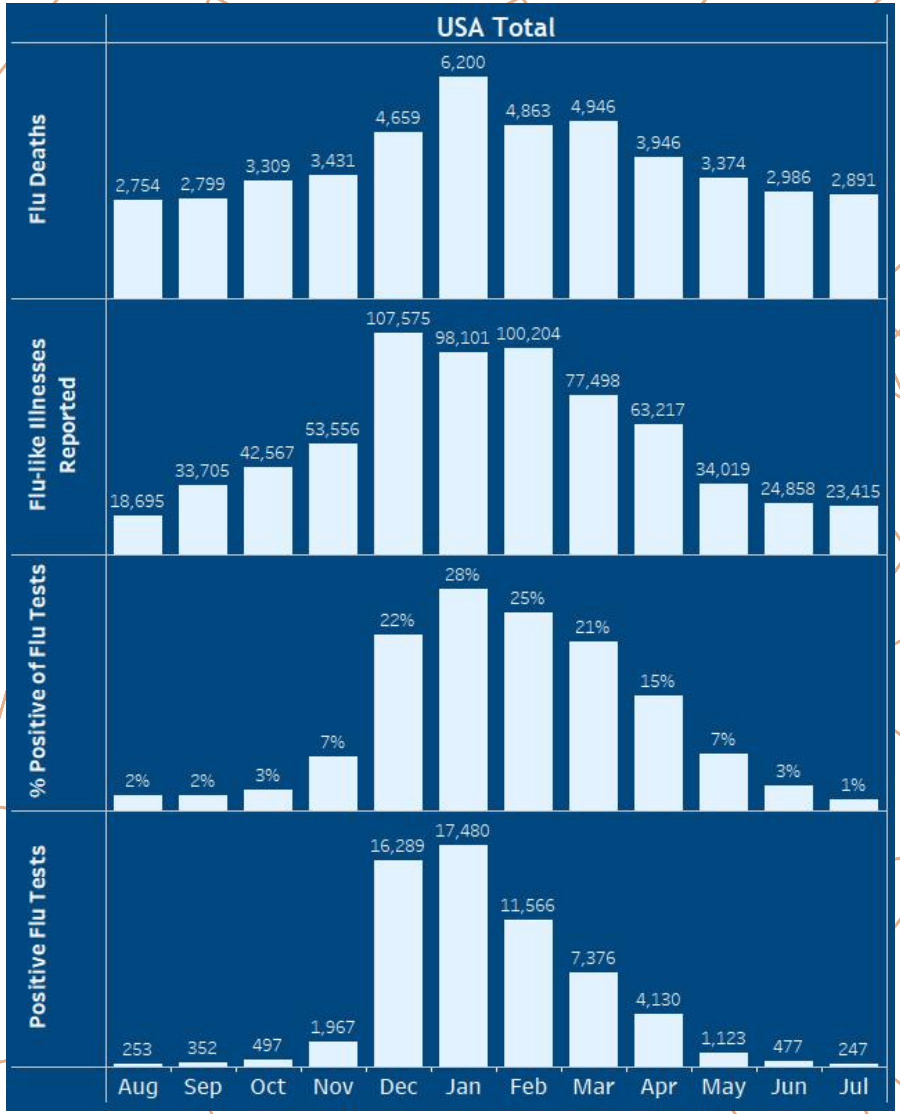


IDENTIFYING FLU SEASON FLUCTUATIONS IN EACH STATE AND REGION

A series of heat maps showing seasonal flu death hotspots was created using the standard deviation for average monthly flu deaths in each state. This helped identify flu trends on both regional and local levels.

ANALYZING VARIOUS FLU SEASON INDICATORS

There are multiple factors that can be used to measure the progression of flu season. By looking at the different measurements together, we can find the earliest indicators that flu season is on the rise and urgent help is needed. This is particularly helpful on a state-by-state basis.



The flu season dashboard can be viewed on Tableau.

STATISTICAL HYPOTHESIS TESTING

	Total Population	population 65 yrs and older	% of population 65 yrs and older	Total flu deaths	65+ Yrs Flu Deaths	Total flu deaths % of 65 yrs and Older	Total % of population	Total Pop % of 65 Yrs
Standard Deviation	6,892,673	897,929	1.671%	1168	1019	5.11%	0.00558%	0.036%
Mean (Average)	6,328,066	859,999	13.951%	975	860	95.44%	0.01296%	0.089%
Outlier Percentage	4.3%	6.3%	6.8%	4.2%	4.1%	3.8%	2.8%	3.1%

Correlation Measuring		
Variables:	% of population 65 yrs and older	Total % of population flu deaths
Correlation Coefficient	0.105	
Strength of Correlation	This would be considered a weak relationship	
Variables:	population 65 yrs and older	Total flu deaths
Correlation Coefficient	0.944	
Strength of Correlation	This is an incredibly strong relationship	
Variables:	Total population	Total flu deaths
Correlation Coefficient	0.953	
Strength of Correlation	This is an incredibly strong relationship	

CORRELATION TESTING

Testing for correlation coefficients indicated that greater numbers of vulnerable populations correlated for greater number of flu deaths, but no more so than simply having greater numbers of overall populations. And greater % of vulnerable populations showed a very weak correlation with total flu deaths.

STATISTICAL ANALYSIS

Descriptive statistics and statistical hypothesis testing affirmed that the vast majority of flu deaths were suffered by those aged 65 and older.

	0-64 Yrs Deaths	*65+ Yrs Deaths*
Mean	85.46808511	896.6099291
Variance	24283.51971	1053020.57
Observations	423	423
Hypothesized Mean Difference	0	
df	441	
t Stat	-16.07303295	
P(T<=t) one-tail	2.17586E-46	
t Critical one-tail	[illegible]	
P(T<=t) two-tail	[illegible]	
t Critical two-tail	[illegible]	

	0-64 years flu deaths % of total population	*65 years and older flu deaths % of total population*
Mean	0.0008%	0.0885%
Variance	9.63844E-11	1.30962E-07
Observations	423	423
Hypothesized Mean Difference	0	
df	423	
t Stat	-49.81094498	
P(T<=t) one-tail	2.3151E-179	
t Critical one-tail	1.648463868	
P(T<=t) two-tail	4.6303E-179	
t Critical two-tail	1.965587999	

Deaths of 65+ as % of population - by Vaccination Status

	High	*Low*
Mean	0.0696%	0.1005%
Variance	1.8E-07	6.6E-08
Observations	90	90
Hypothesized Mear	0	
df	146	
t Stat	-5.905	
P(T<=t) one-tail	1.2E-08	
t Critical one-tail	1.65536	
P(T<=t) two-tail	2.37E-08	
t Critical two-tail	1.97635	

Deaths of 65+ as % of population - by Population Density

	Top third	*Bottom third*
Mean	0.101%	0.062%
Variance	1.4E-07	1.3372E-07
Observations	144	150
Hypothesized Me	0	
df	291	
t Stat	8.963436	
P(T<=t) one-tail	1.95E-17	
t Critical one-tail	1.650107	
P(T<=t) two-tail	3.90E-17	
t Critical two-tail	1.96815	

Deaths of 65+ as % of population by weighted pop density, urbanization, and vaccination status

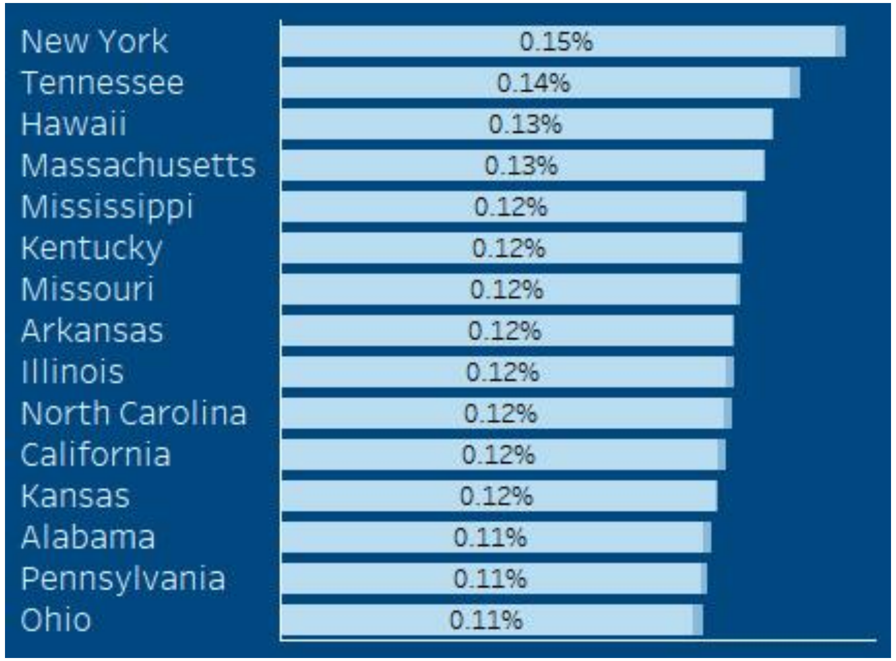
	Bottom third	*Top third*
Mean	0.052%	0.099%
Variance	1.1123E-07	1.3E-07
Observations	150	162
Hypothesized M	0	
df	310	
t Stat	-12.020276	
P(T<=t) one-tail	7.0149E-28	
t Critical one-tail	1.64978382	
P(T<=t) two-tail	1.403E-27	
t Critical two-tai	1.96764593	

STATISTICAL HYPOTHESIS TESTING

Further statistical hypothesis testing on states divided into high and low groups based on factors such as vaccination status, population density, and urbanization showed very strong statistical significances in their average differences. When all three factors were weighted together, it amounted to a 50% difference in vulnerable population flu deaths.

For the full analysis, read the Interim Report. Or visit Tableau to interact with it visually on the flu factors dashboard.

RESULTS

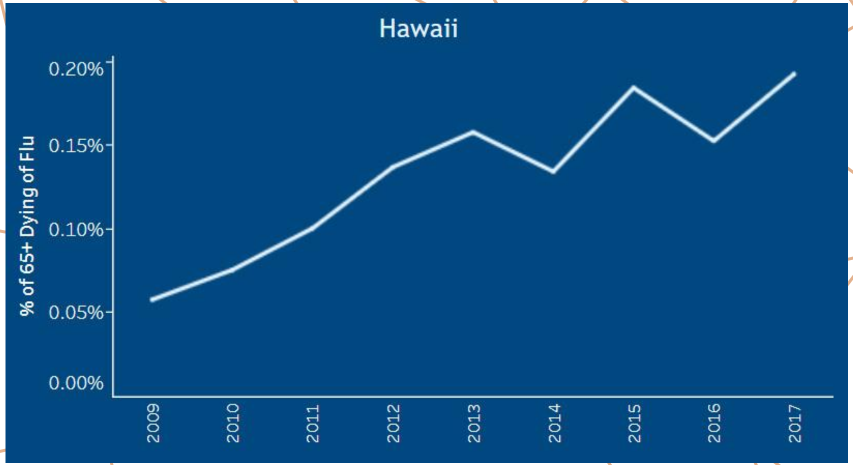


IDENTIFYING STATES MOST VULNERABLE TO FLU

The top states with the greatest percentage of flu deaths in their vulnerable populations have been identified for targeted support. With the flu season dashboard it can be determined when their flu season is entering its severe stages and it's time to send additional support.

TIMELINE OF FLU IMPACT PER STATE

The flu season dashboard contains a timeline for each state's progress or challenges regarding flu deaths over the years. Some states need additional year-round support to combat recent negative trends.



COMPLEX SEASONAL FACTORS

Flu season is a recurring challenge throughout the US, compounded by complex factors that defy simple solutions. However, with the interactive flu season dashboard, contributing factors and timelines specific to each state can be identified and accounted for, to provide targeted medical support when and where it's needed most.

The interactive flu season preparation dashboard and the entire Understanding Flu Season storyboard can both be viewed on Tableau.

ROCKBUSTER STEALTH

OBJECTIVE

Flailing brick-and-mortar video rental giant seeks to launch streaming service to meet customer demand. Their current portfolio and customer trends must be analyzed to inform strategy for the new service launch.

DATA

This dataset is provided by PostgreSQL for usage in tutorials. It contains data about film inventory, customers, payments, and associated details. The dataset can be accessed here.

TOOLS & SKILLS

PostgreSQL DbVisualizer The Universal Database Tool tableau Excel

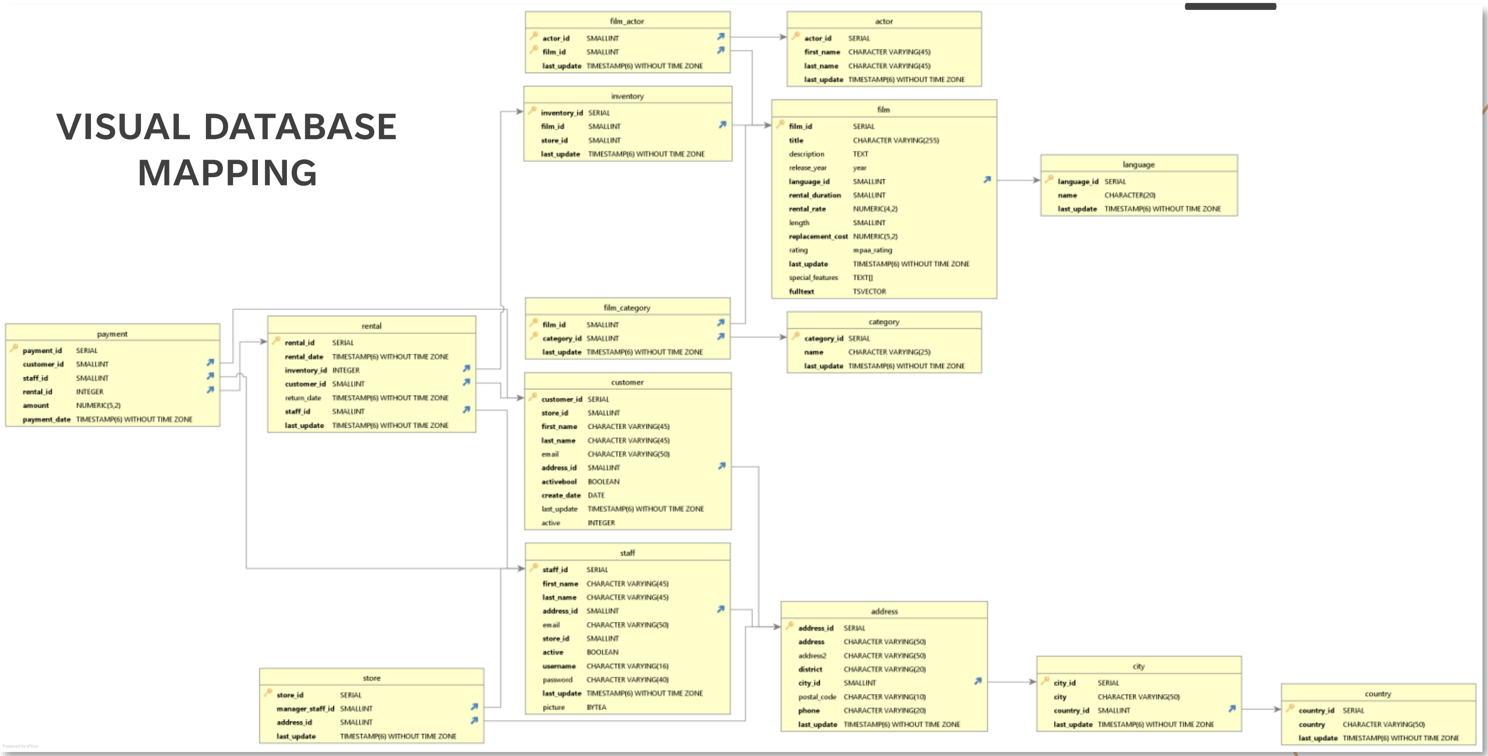
- Relational databases in SQL
- Entity relationship diagram creation and usage
- Data dictionary creation
- Database querying, filtering, and cleaning
- Joining tables in relational database
- Subqueries and common table expressions

CONSTRAINTS

Because this dataset was intended for public tutorial usage, it doesn't contain any realistic data. Instead, all of the information was scrambled, including movie titles, actor names, customer locations, and rental habits. Though an "analysis" was possible for practicing SQL skills, it unfortunately has no real-world reference, which has made gaining legitimate business insights impossible.

DATABASE MANAGEMENT

VISUAL DATABASE MAPPING



By creating an entity relationship diagram, the facts and dimension tables were able to be seen at a glance, with an easy understanding of how the tables relate to one another.

DATA DICTIONARY

2 Legend:

Primary Key
Foreign Key

3 Fact Tables:

3.1 payment

Columns	Data Type	Description	Links to:
payment_id	SERIAL	Primary key, unique serial number for each transaction	
customer_id	SMALLINT	Foreign key linking to unique id numbers for each customer	customer, rental
staff_id	SMALLINT	Foreign key linking to unique id numbers for each employee	staff, rental, store
rental_id	INTEGER	Foreign key linking to unique id number for each rental transaction	rental
amount	NUMERIC(5,2)	Amount of transaction with two decimal places	
payment_date	TIMESTAMP(6) WITHOUT TIME ZONE	Date and time of the transaction	

3.2 rental

Columns	Data Type	Description	Links to:
rental_id	SERIAL	Primary key, unique serial number for each rental transaction	payment
rental_date	TIMESTAMP(6) WITHOUT TIME ZONE	Date and time of the rental	
inventory_id	INTEGER	Foreign key linking to unique id number for each rental movie	inventory
customer_id	SMALLINT	Foreign key linking to unique id numbers for each customer	customer

Creating a data dictionary allows the content of every variable to be quickly identified. It also gives quick reference to each variable's data type and its relation to other tables within the database.

The entire Rockbuster Stealth Data Dictionary can be accessed via GitHub.

ANALYSIS WITH SQL & TABLEAU

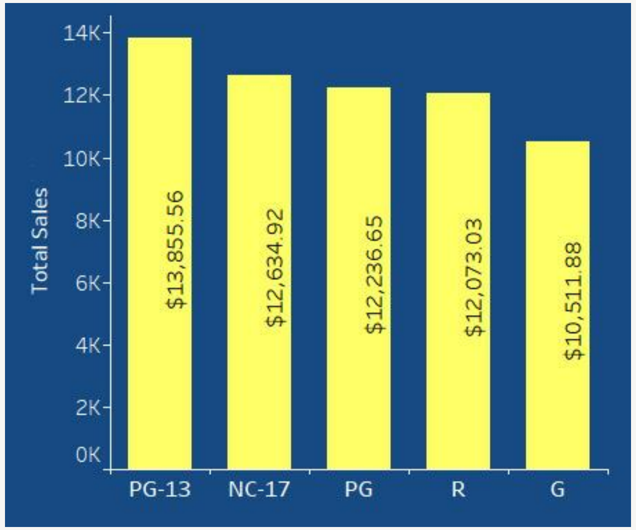
SQL QUERYING & TABLEAU VISUALIZING

Query Query History

```sql
SELECT A.rating,
    SUM(D.amount) AS total_sales,
    AVG(D.amount) AS average_rental_cost,
    COUNT(C.rental_id) AS number_of_rentals
FROM film A
    INNER JOIN inventory B ON A.film_id = B.film_id
    INNER JOIN rental C ON B.inventory_id = C.inventory_id
    INNER JOIN payment D ON C.rental_id = D.rental_id
GROUP BY rating
ORDER BY total_sales DESC
```

To answer the business questions posed, the right table joins and queries had to be written in SQL. Then the resulting table was exported to a csv file and imported into Tableau. At that point, a visualization showing the answers to the business questions could be created.

RENTAL INCOME BY RATING



Some business questions required much more complex common table expressions and sub-queries, as seen on the right. It was used to determine the top 5 paying customers from the top 10 cities within the top 10 countries that have the most Rockbuster customers.

COMMON TABLE EXPRESSIONS

Query Query History

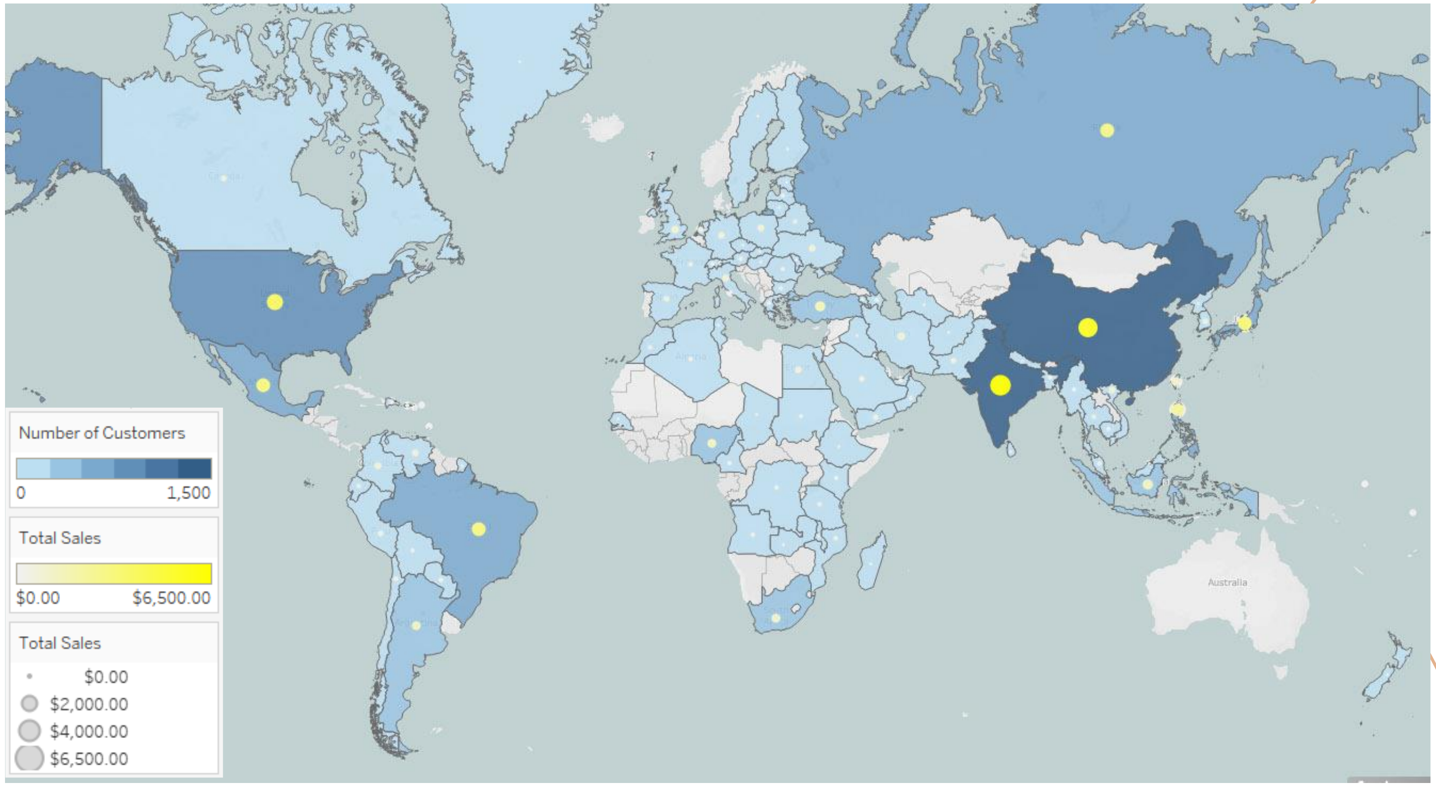
```sql
WITH top_10_cities (city, country, customer_count) AS
  (WITH top_10_countries (country, customer_count) AS
    (SELECT D.country,
        COUNT(customer_id) AS number_of_customers
      FROM customer A
        INNER JOIN address B ON A.address_id = B.address_id
        INNER JOIN city C ON B.city_ID = C.city_id
        INNER JOIN country D on C.country_ID = D.country_ID
      GROUP BY country
      ORDER BY COUNT(customer_id) DESC
      LIMIT 10)

        SELECT C.city,
          D.country,
          COUNT(customer_id) AS number_of_customers
        FROM customer A
          INNER JOIN address B ON A.address_id = B.address_id
          INNER JOIN city C ON B.city_ID = C.city_id
          INNER JOIN country D on C.country_ID = D.country_ID
        WHERE D.country IN (SELECT country FROM top_10_countries)
        GROUP BY c.city, d.country
        ORDER BY COUNT(customer_id) DESC
        LIMIT 10)

SELECT
  A.customer_id,
  A.first_name,
  A.last_name,
  C.city,
  D.country,
  SUM(F.amount) AS total_paid_to_rockbuster
FROM customer A
  INNER JOIN rental E ON A.customer_id = E.customer_id
  INNER JOIN payment F ON E.rental_id = F.rental_id
  INNER JOIN address B ON A.address_id = B.address_id
  INNER JOIN city C ON B.city_ID = C.city_id
  INNER JOIN country D on C.country_ID = D.country_ID
WHERE city IN (SELECT city FROM top_10_cities)
GROUP BY a.customer_id, A.first_name, A.last_name, C.city, D.country
ORDER BY total_paid_to_rockbuster DESC
LIMIT 5
```

The GitHub repository for this project contains several of the more complex CTEs and sub-queries created.

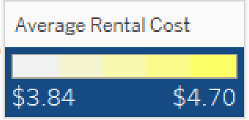
FINAL RESULTS

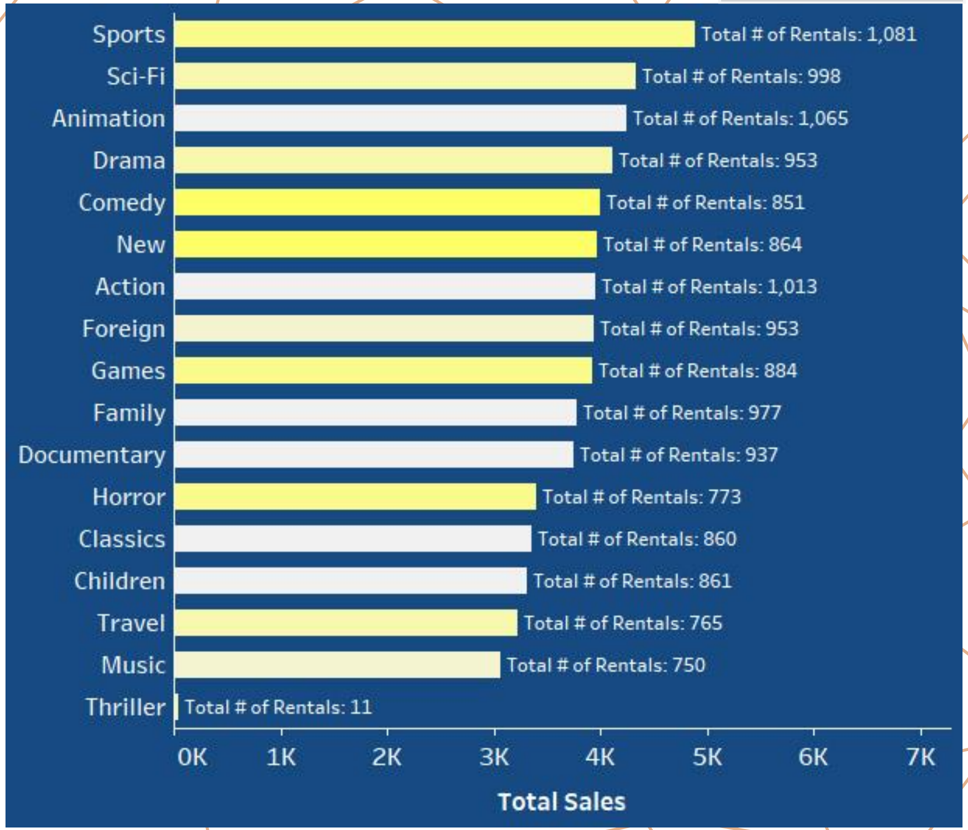
ROCKBUSTER CUSTOMER LOCATIONS

After exporting csv's from SQL queries, the dataset was imported into Tableau to create an interactive map of Rockbuster's customers. Asia has the largest sales and greatest number of customers, with North American and South America being the second and third largest regions.



MOVIE RENTAL SALES PERFORMANCE BY GENRE





Some movie genres make less money even with more overall rentals. This is because the average rental costs for those genres are lower.

The [interactive visualizations]() can be found on Tableau. Click here to view [the full Rockbuster report]().

INSTACART BASKET ANALYSIS

OBJECTIVE

An analysis of Instacart customer purchasing habits must be performed. Results will be used to gain insight and develop various customer profiles with the goal of forming a targeted marketing strategy.

DATA

The orders and product information data is published as open source from Instacart. The customer and demographic data was fabricated for the purpose of this analysis and can be downloaded here.

TOOLS & SKILLS

python ANACONDA jupyter pandas NumPy seaborn matplotlib Excel

- Data wrangling and dataframe merging in Python
- Deriving new variables
- Crosstabs and pivot tables in Python
- Visualizations in multiple Python libraries
- Markup and notebook management in Jupyter

CONSTRAINTS

The open source dataset from Instacart has high quality and integrity. However, it had no demographic data for its customer_id numbers. The fabricated dataset supplies this additional customer data. However, most of the additional variables were evenly distributed among the fabricated dataset in completely unrealistic ways. Because of this the customer profiling was unable to support the goal of targeted marketing.

DATA WRANGLING

DERIVING VARIABLES

5. Determining high and low spenders based on average spending per item across all orders per customer

```
In [12]: # Creating average price column by user id

         df['avg_item_price'] = df.groupby(['user_id'])['prices'].transform(np.mean)

In [14]: # creating flag column for low (<10) and high (>=10) spenders

         df.loc[df['avg_item_price'] < 10, 'spender_type'] = 'Low spender'
         df.loc[df['avg_item_price'] >= 10, 'spender_type'] = 'High spender'

In [15]: df['spender_type'].value_counts(dropna = False)

Out[15]: Low spender     31770614
         High spender      634245
         Name: spender_type, dtype: int64
```

New variables were derived and were then used to flag customers within different categorizations.

ORGANIZING SCRIPTS IN JUPYTER

Table of Contents

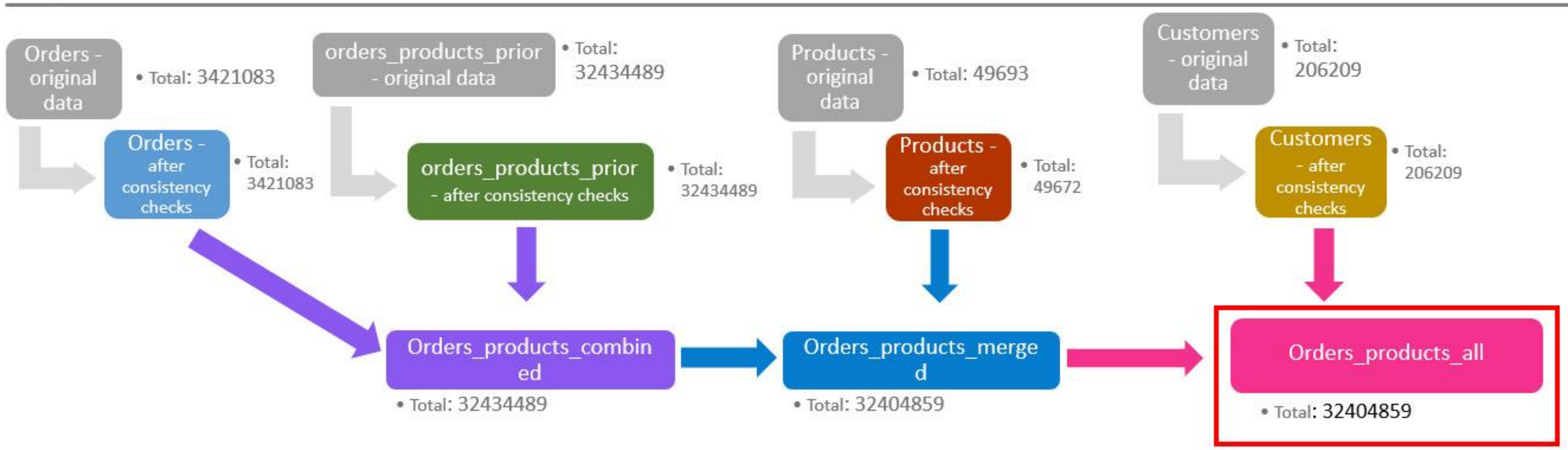
1. Importing libraries
2. Importing dataframes
3. Consistency checks
4. Exporting data
5. Task work

    5b. Consistency check on orders dataframe
    5c. Looking for mixed data types
    5d. Looking for missing data
    5e. Looking for duplicate values

6. Exporting data

DATA CLEANING AND MERGING

Population flow



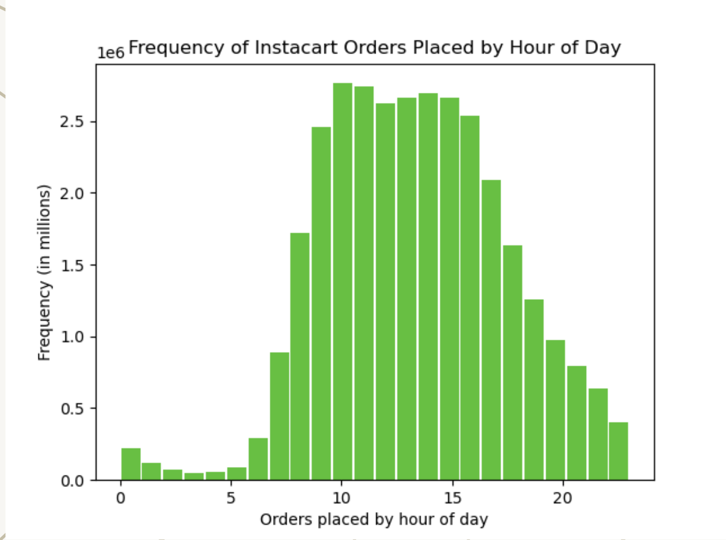
Multiple datasets were checked, cleaned, and merged to create the final dataframe with a total of over 32 million records to be analyzed.

The [Python scripts](#) for this project can be found in the GitHub repository

ANALYSIS IN PYTHON

HISTOGRAM

```
In [87]: # creating histogram

         hist = df['order_hour_of_day'].plot.hist(bins = 24, color = '#68bf43', rwidth=0.9)

         plt.xlabel("Orders placed by hour of day")
         plt.ylabel("Frequency (in millions)")
         plt.title("Frequency of Instacart Orders Placed by Hour of Day")
```



Finding the busiest time of day for Instacart orders

CROSSTAB & PIVOT TABLE CREATION

```
In [62]: # creating a crosstab to compare the two columns' values
         crosstab = pd.crosstab(df['region'], df['spender_type'], dropna = False)

In [15]: crosstab.to_clipboard()

In [16]: crosstab.sort_values(by="High spender", ascending = False)
```

Out[16]:

spender_type region	High spender	Low spender
South	209691	10582194
West	160354	8132559
Midwest	155975	7441350
Northeast	108225	5614511

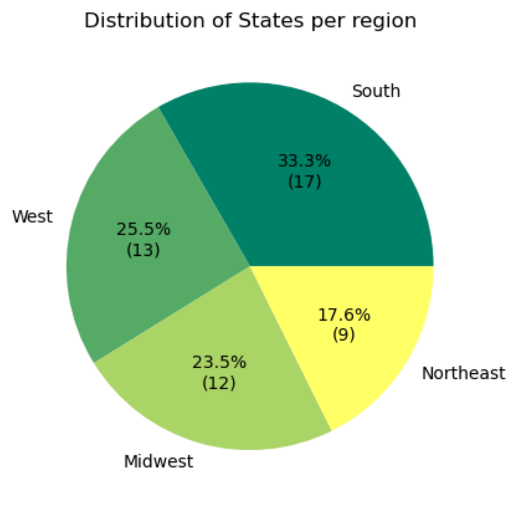
```
In [24]: # creating pivot table to see average per-state high and low spenders by region

         pivot = np.round(pd.pivot_table(crosstab_new, values=['High spender', 'Low spender'],
                                          index=['region'],
                                          aggfunc=np.mean),2)

In [25]: pivot
```

Out[25]:

spender_type region	High spender	Low spender
Midwest	12997.92	620112.50
Northeast	12025.00	623834.56
South	12334.76	622482.00
West	12334.92	625581.46

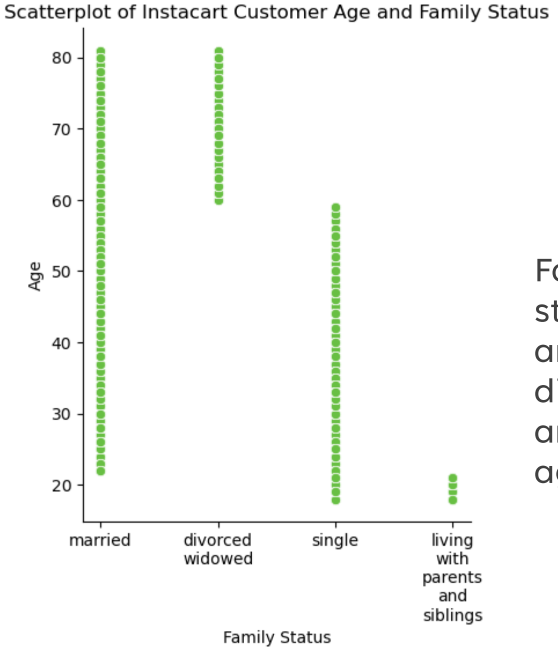
PIE CHART

```
In [77]: # creating pie chart of states per region to compare to several of the profiles to see if there's
         # any difference or if it's all just evenly distributed

         # defining my own autopct call-out so I can include both percentage and actual values for comparing
         def make_autopct(values):
             def my_autopct(pct):
                 total = sum(values)
                 val = int(round(pct*total/100.0))
                 return '{p:.1f}%\n({v:d})'.format(p=pct,v=val)
             return my_autopct

         values = crosstab_new['region'].value_counts()
         region_pie = crosstab_new['region'].value_counts().plot(kind='pie', autopct=make_autopct(values), colormap='summer')
         plt.title ("Distribution of States per region")
         plt.xlabel("")
         plt.ylabel("")
```

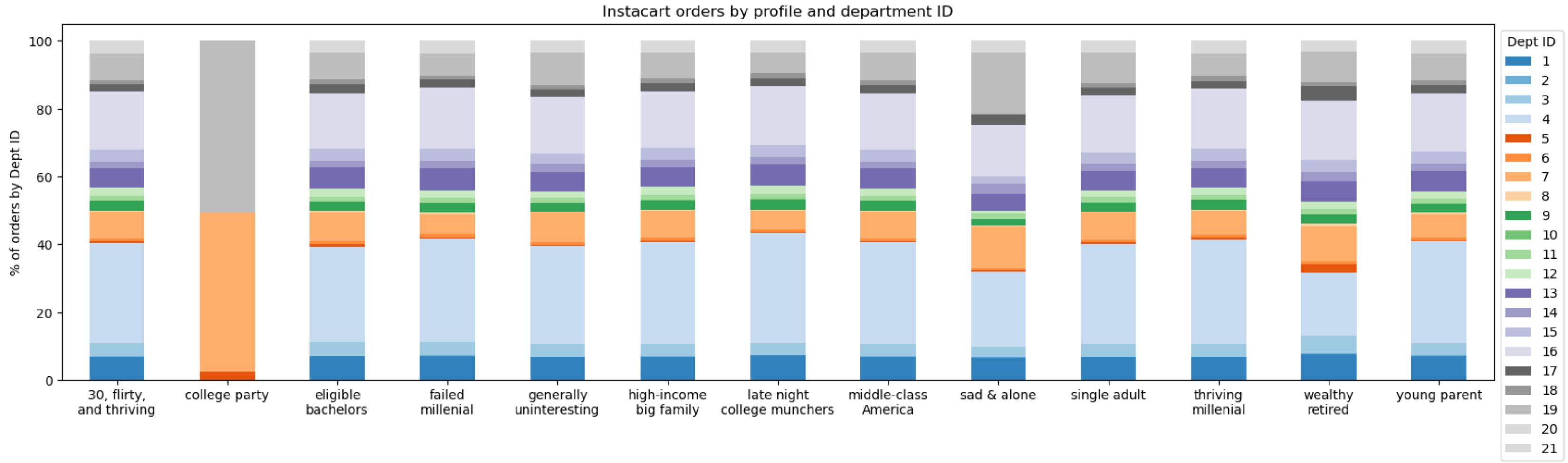


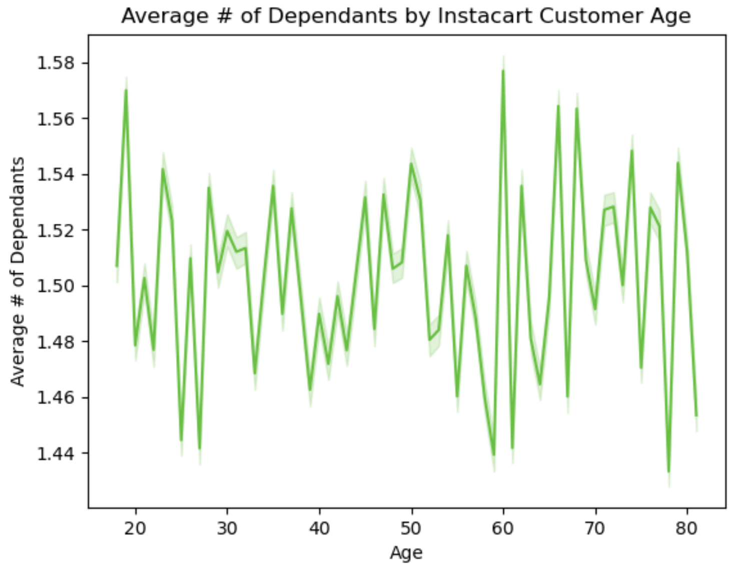
Regional distribution of Instacart customer profiles was considered, but found to be too evenly distributed to provide any business insights.

CUSTOMER PROFILING



Family status was artificially distributed among set age ranges.





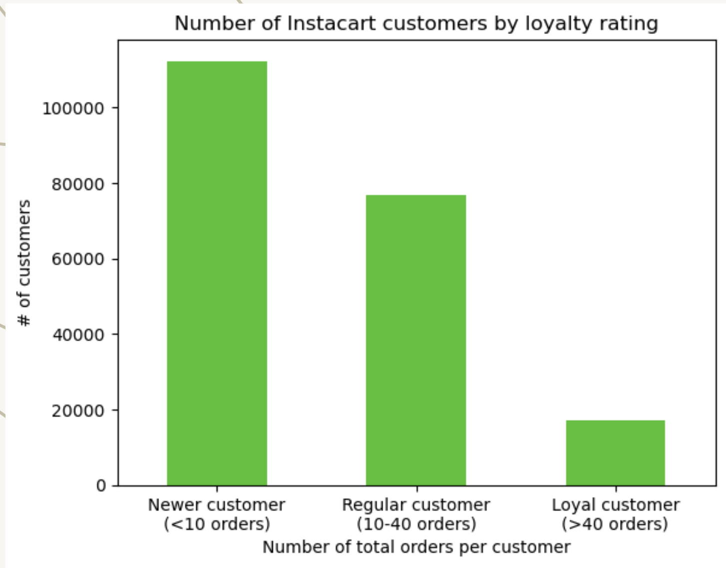
Number of dependents was distributed evenly regardless of age.

No matter how simple or complex the customer profiling, there were very few differences from one group to the other due to the artificially even distribution of customer demographic data.

	prices			avg_days_since_last_order		
	mean	min	max	mean	min	max
profile						
30, flirty, and thriving	7.970154	1.0	25.0	11.168295	0.000000	30.0
college party	5.931935	1.0	15.0	11.598381	0.675261	30.0
eligible bachelors	7.919347	1.0	25.0	11.019103	2.320437	30.0
failed millenial	7.918967	1.0	25.0	10.838530	1.072381	30.0
generally uninteresting	7.709824	1.0	25.0	11.021521	0.000000	30.0
high-income big family	7.963540	1.0	25.0	11.016399	0.726027	30.0
late night college munchers	7.884527	1.0	25.0	11.157885	0.675261	30.0
middle-class America	7.889505	1.0	25.0	10.971854	0.304428	30.0
sad & alone	6.521983	1.0	25.0	11.062230	0.000000	30.0
single adult	7.742998	1.0	25.0	10.973760	0.288591	30.0
thriving millenial	7.984491	1.0	25.0	11.008352	0.000000	30.0
wealthy retired	7.776435	1.0	25.0	9.970635	0.717842	30.0
young parent	7.766060	1.0	25.0	11.022886	0.675261	30.0

All of the visualizations created can be found in the GitHub repository.

FINAL RESULTS

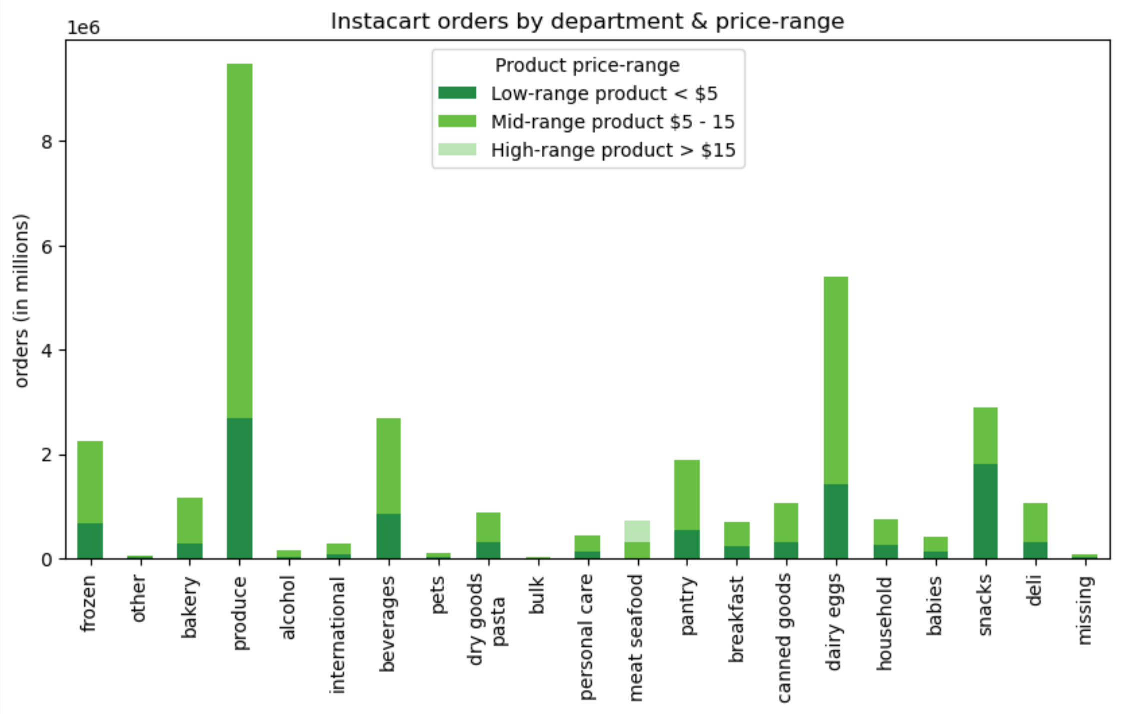


With no meaningful way to profile customers, the final analysis has to rely on more traditional metrics.

"CUSTOMER LOYALTY" COMPARISONS

Most Instacart customers have made less than 10 orders, but they account for the smallest number of overall orders. Finding ways to convert "newer customers" to "regular" and "loyal customers" will significantly increase overall Instacart orders.

ORDERS AND PRICES BY DEPARTMENT



Though produce is the most common department ordered from, it has the lowest range of prices, while meat and seafood has the highest prices. Snacks is a popular category and also has the greatest proportion of mid-range prices of any department. Targeted advertising for higher-priced departments can increase Instacart's overall revenue.

The final report and full analysis can be found here.

ESSENTIAL WORKER AFFORDABLE HOUSING

OBJECTIVE

Find currently available affordable housing in food production facility markets. An operational dashboard is needed to analyze the current situation and identify solutions.

TOOLS & SKILLS

python plotly BeautifulSoup scikit learn ANACONDA jupyter pandas NumPy seaborn matplotlib tableau

- Web-scraping and data cleaning in Python
- Machine learning in Python
- Time-series data analysis in Python
- Operational dashboards in Tableau

DATA

The aggregated historical real estate data is published as open source from realtor.com. The detailed current market data was scraped from realtor.com's publicly facing search engine, most recently on June 16, 2023. The web-scraping script can be found on my GitHub.

CONSTRAINTS

This project required both historical data and non-aggregated current market data. The historical data has very high quality and integrity. However, the up-to-date scraped data from realtor.com had to be cross-checked manually and the web-scraping script had to be updated iteratively together with robust cleaning scripts in order to produce consistently reliable outputs that could be automated for future updates.

WEB SCRAPING

OBTAINING CURRENT DATA

This project required non-aggregated current market data for both “for sale” and “rental” listings. No reliable country-wide data could be found for current listings, so a web scraper had to be created in Python, and the script had to be made to avoid anti-bot detection.

```
# Loop through the list of zip codes.
for zip_code in zip_codes:
    print(f"Processing zip code: {zip_code[0]}")

    # Add a random delay before going to next zip code
    time.sleep(random.randint(30, 90))

    # Loop through the page numbers (1-20).
    for page_num in range(1, 20):
        print(f"Processing page {page_num}")

        # Update the URL in the script.
        url_template = "https://www.realtor.com/apartments/{}/pg-{}"
        url = url_template.format(zip_code[0], page_num)

        # Set user-agent header to avoid bot detection
        headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like G

        # Add a random delay before making the request
        time.sleep(random.randint(8, 16))

        # Send HTTP GET request to the URL and get the HTML response
        response = requests.get(url, headers=headers)
        if response.status_code == 200:
            print("Request successful")
        else:
            print(f"Request failed with status code: {response.status_code}. No more listings pages found.")
            break

        # Parse the HTML response using BeautifulSoup
        soup = BeautifulSoup(response.content, 'html.parser')

        # Find the container for all the properties
        container = soup.find('section', class_=re.compile(r'PropertiesList_propertiesContainer'))

        # Check if the container is empty
        if container is None:
            print("No listings found")
            break
```

CLEANING OUTPUT

The data that was parsed from realtor.com’s website had a lot of mixed formatting and had to go through several rounds of automated cleaning in order to produce regular usable output.

```
# Split the beds, baths, and sqft values at the " - " and create two new rows with the values before

if "-" in price:
    price_split = price.split(" - ")
    if "-" in beds:
        beds_split = beds.split(" - ")
        if "-" in sqft:
            sqft_split = sqft.split(" - ")
            if "-" in baths:
                baths_split = baths.split(" - ")
                data.append([str(zip_code[0]), address1, address2, style, price_split[0], beds_split[
                data.append([str(zip_code[0]), address1, address2, style, price_split[1], beds_split[

            else:
                data.append([str(zip_code[0]), address1, address2, style, price_split[0], beds_split[
                data.append([str(zip_code[0]), address1, address2, style, price_split[1], beds_split[
        else:
            data.append([str(zip_code[0]), address1, address2, style, price_split[0], beds_split[0],
            data.append([str(zip_code[0]), address1, address2, style, price_split[1], beds_split[1],
    else:
        data.append([str(zip_code[0]), address1, address2, style, price_split[0], beds, baths, sqft])
        data.append([str(zip_code[0]), address1, address2, style, price_split[1], beds, baths, sqft])
else:
    data.append([str(zip_code[0]), address1, address2, style, price, beds, baths, sqft])
```

ADDING NEEDED CATEGORICAL INFORMATION

The final scraped data had to be merged with other categorical data so that it could be related to other datasets used for analysis.

Merging county information to the dataframe

```
# Adding the columns and values I want from the zip_cnty dataframe to my df_rr dataframe
df_rr = df_rr.merge(zip_cnty[['zip', 'lat', 'lng', 'county_fips', 'county_name']],
                    left_on='Zip', right_on='zip', how='left')

# Remove the redundant 'zip' column from the merge result
df_rr.drop('zip', axis=1, inplace=True)
```

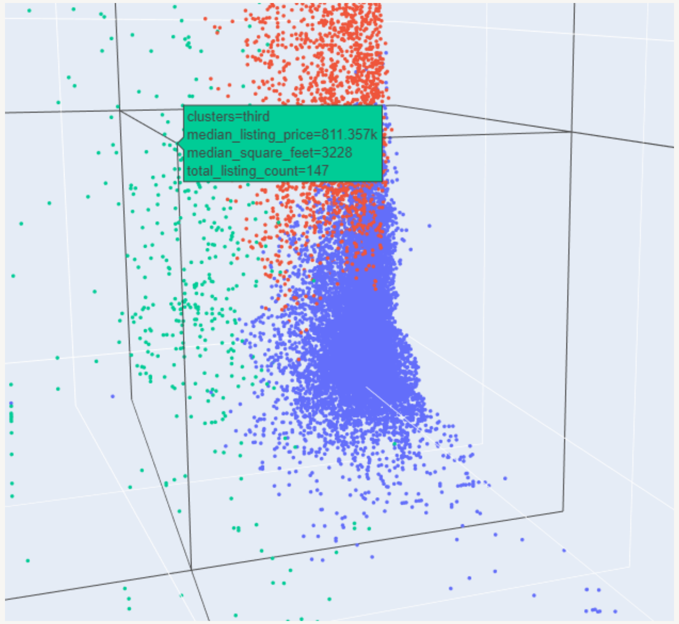
```
# Converting columns with leading zeroes to strings so that I can keep those leading zeroes when exporting

df_rr[['TysonZip', 'Zip', 'county_fips']] = df_rs[['TysonZip', 'Zip', 'county_fips']].astype(str)
```

The full Python scripts for this project can be found in the GitHub repository

MACHINE LEARNING

CLUSTERING & 3D MODELING

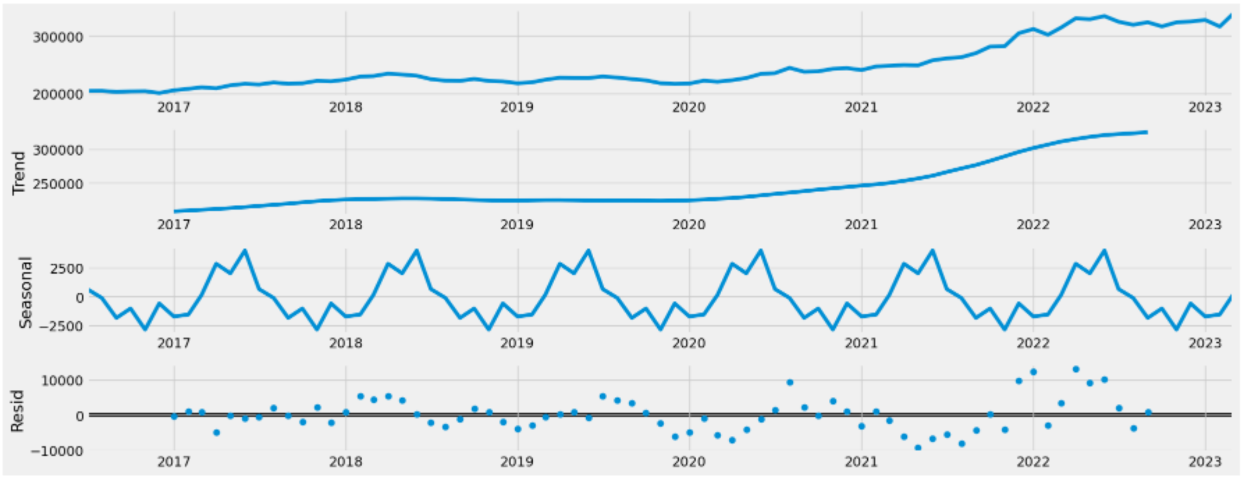
	clusters	median_listing_price		median_square_feet		median_listing_price_per_square_foot		new_listing_count		total_listing_count	
	size	mean	median	mean	median	mean	median	mean	median	mean	median
clusters											
hot market	2986	335780.496651	336955.5	2262.990623	2159.0	150.569324	142.0	62.570663	56.0	197.485599	179.0
typical market	13414	193035.767631	171487.5	1823.466677	1756.0	105.025868	97.0	10.833458	6.0	34.860593	26.0
wealthy market	618	974572.739482	815113.5	3186.943366	3338.5	331.854369	312.5	28.951456	26.0	93.257282	85.0



An unsupervised machine learning cluster analysis discovered three market clusters in the historical home data: “typical” markets, “hot” markets, and “wealthy” markets. The results were displayed in a 3D scatterplot to better visualize their distinctions.

TIME SERIES DECOMPOSITION

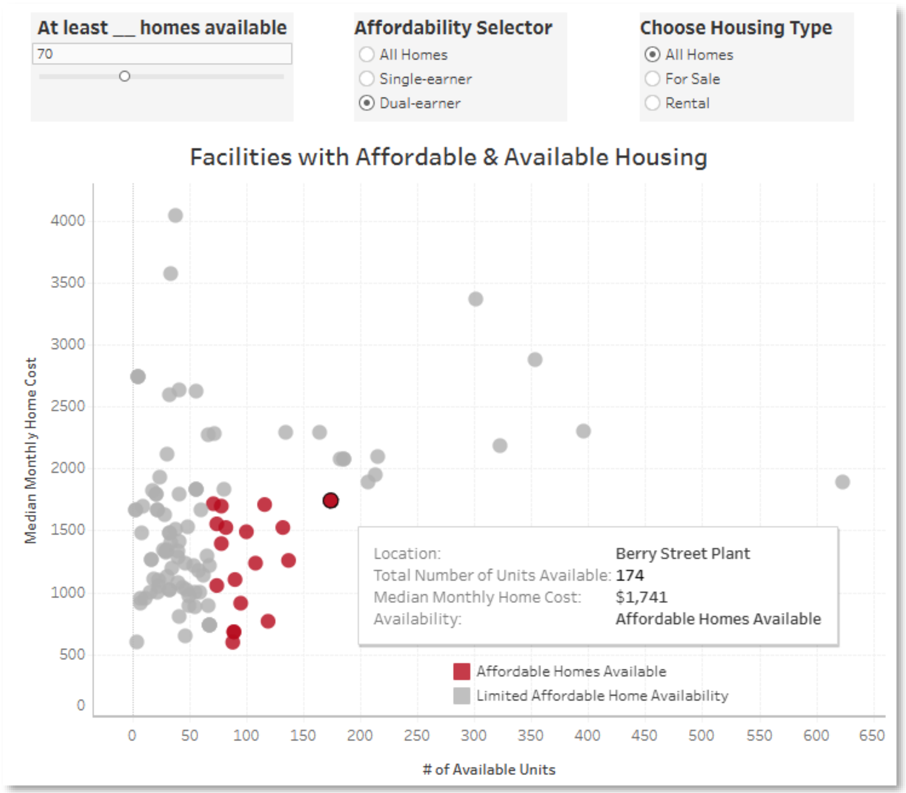
The time series data was decomposed into seasonality, trend, and residual data points. It was then further analyzed for autocorrelation and transformed for stationarity to prepare it for forecasting.



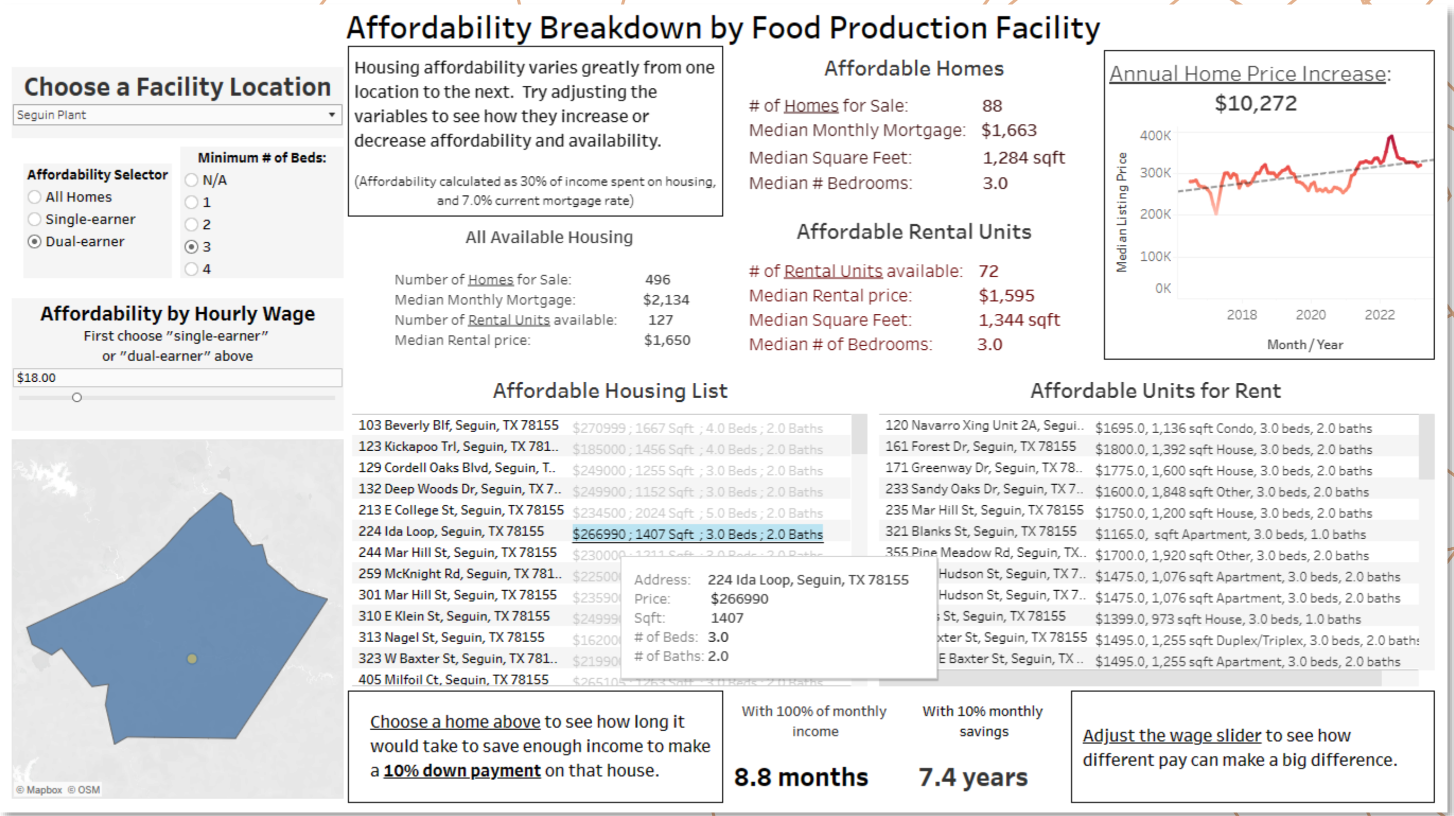
The results of the machine learning analyses helped produce insights into the data sets and the trends within them, which helped give direction to the scope of the project. But the direct results of these analyses would not be useful for the average user that this project is intended for, and thus were not included in the final operational dashboards.

The Python scripts used to conduct the machine learning analyses can be found in the GitHub repository

OPERATIONAL DASHBOARDS



In all of the dashboards, users are allowed to adjust the variables that determine affordability factors. Each location, and all currently available properties at each location, can be examined in-depth, together with the results of multiple factors that affect essential workers' abilities to afford housing. The datasets are intended to be updated monthly.



The goal of this project was to enable users to find up-to-date information on affordable housing in the vicinity of food production facilities. Multiple dashboards were created to allow users to approach the problem from the big picture while also having the opportunity to drill down to the specific details.

All of the dashboards can be found in the [Tableau storyboard]

THANK YOU

tableau GitHub

ADAM WILLARD