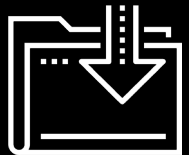




Egad It's Data – Part 1

Data Boot Camp
Lesson 1.2



Welcome to Day 2!

As a reminder:

01

Make certain that you have Microsoft Excel installed.

02

Make certain that you have Slack installed and are actively looking at it.

03

Recall where the GitLab repository for our class is.

04

Note where class videos will be posted.

More on Slack

Because we will use Slack so regularly for communication, let us review:

- Set up Mobile notifications
- Looking for direct messages
- Viewing replies
- Replying to posts
- Reacting to posts
- Formatting messages
- Anything else?

Quick review of the Student Guide!

Class objectives, helpful articles, supplemental resources, and more!

Also, let us look at the Student FAQ.

Example Activity



Example Future Class Activity: Banking Deserts

In this activity, you will use a variety of public demographic data and APIs to explain many real-world social phenomena. Utilize data from sources like the U.S. Census, Google Maps, and more to find insights on poverty, discrimination, and the impact of changing economies.

Sample Activity



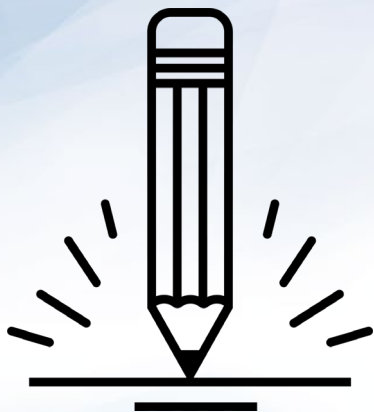


Group Activity:

Form groups of 7 people.
(The people closest to you)

Suggested Time:
1 minute





Group Activity: The Great Debate

Find your group you formed before the break. Together ponder the following question.

Suggested Time:
10 minutes



Group Activity: The Great Debate

Which do Americans prefer:
Italian or Mexican food?



Group Activity: The Great Debate

With your group, develop a strategy for answering this question with as much confidence possible. Specifically, answer questions like:



What data will you attempt to gather?



What relationships will you be looking for?



How will you ensure your answer is most likely “true”?

Assumptions:

You are given 5 hours and a budget of \$10 to accomplish this.

Your answer will be tested by randomly selecting 9 Americans who will each be asked the question—with 0 qualifiers.

You only have your team.

Suggested Time: 10 minutes



The Great Debate (Analyzed)

Step 1: Decompose the “Ask”

Step 1: Decompose the “Ask”

Which do **Americans** prefer:
Italian or Mexican food?



Step 1: Decompose the “Ask”

Which do **Americans** prefer: Italian or Mexican food?



Who exactly is an **American**?



Are **Americans** just homeowners?



Do **Americans** just live in big cities?



Are **Americans** just millennials?



How can we get a
representative sample
of Americans?

Step 1: Decompose the “Ask”

Which do Americans **prefer**:
Italian or Mexican food?



Step 1: Decompose the “Ask”

Which do Americans **prefer**: Italian or Mexican food?



How do we define “preference”?



Do people prefer the foods they eat most frequently?



Do people prefer the foods they wish they could eat if cost was not an issue?



How uniform is the preference? Is it regionalized? Is it different by demographic?



Inherently, preference is **subjective**. We are going to need to make it **objective**.

Step 1: Decompose the “Ask”

Which do Americans prefer:
Italian or Mexican food?



Step 1: Decompose the “Ask”

Which do Americans prefer: **Italian or Mexican food**?

01

How do we
categorize foods?
Is pizza Italian? Is
Taco Bell Mexican?

02

How do we
categorize food?
Does making pasta
at home constitute
Italian? Or are we
just talking about
restaurants?

03

Are we just talking
about “best
experiences”? Or
are we including
poorer renditions of
these foods?

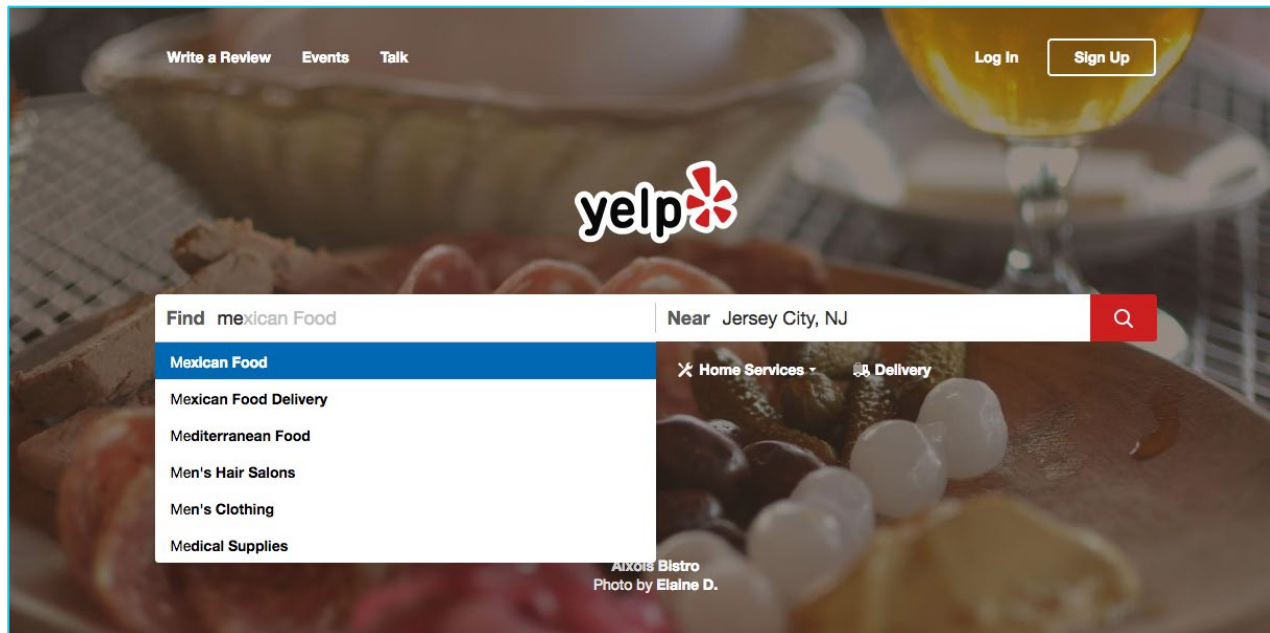


Italian and Mexican are
broad categories we are
pursuing. We will have to
narrow the scope.

Step 2: Identify Data Sources

Step 2: Identify Data Sources

As everyday consumers, we are **regularly** getting a pulse of everyday American food preferences to inform our own decisions. Perhaps we can make use of the same approach.



Step 2: Identify Data Sources

Web services like Yelp provide an almost encyclopedic amount of information about the eating preferences of Americans.

The screenshot shows the Yelp profile for Mi Mariachi Taqueria. At the top, the search bar contains "Find tacos, cheap dinner, Max's" and the location is set to "Near Jersey City, NJ". The page features a red header with the Yelp logo and navigation links for Home Services, Restaurants, Auto Services, and a "Write a Review" button. The business name "Mi Mariachi Taqueria" is prominently displayed, along with a "230 reviews" rating and a "Details" link. A map on the left shows the location at 213 Sip Ave, Jersey City, NJ 07306, with contact information including a phone number and email. The main content area includes a collage of photos: a restaurant interior, a close-up of a chorizo and egg sandwich, and a menu board. A review snippet is visible, mentioning "AI Pastor" and "carnitas tacos". At the bottom, a "Full menu" link and a price range of "\$\$\$\$ Under \$10" are shown.

https://www.yelp.com/biz/mi-mariachi-taqueria-jersey-city-2

Find tacos, cheap dinner, Max's Near Jersey City, NJ

Home Services Restaurants Auto Services More Write a Review

Mi Mariachi Taqueria Unclaimed

★ ★ ★ ★ ★ 230 reviews Details

★ Write a Review Add Photo Share Save

\$ Mexican Edit

213 Sip Ave
Jersey City, NJ 07306
Get Directions
(201) 222-1998
mimariachi.letseat.at
Send to your Phone

Chorizo & egg sandwich no cheese. Simply... by Franco B.

"Love their AI Pastor and **carnitas tacos**, shredded lamb, pork ribs with salsa verde and their tamales!" in 13 reviews

Full menu

\$\$\$ Price range Under \$10

Step 2: Identify Data Sources

Why poll an audience when there already exist enormous databases of information about Americans' food preferences—readily available online?



Step 2: Identify Data Sources

Food Type

Best Italian Food Jersey City, NJ

Review Count

108 reviews

Rating

Zero Otto Uno Cafe

54 reviews



Lots of Data!

Locations

Step 3: Define Strategy and Metrics

Step 3: Define Strategy and Metrics

Here we created a blueprint for what we're targeting:

Americans:

- Ideally, we need thousands of records from Americans in hundreds of different cities. (Large samples)

Preference:

- Number of Yelp Reviews (More = Preference)
- Average Aggregated Ratings (Higher = Preference)

Italian and Mexican Food:

- Top 20 Italian and Mexican restaurants in every city

Step 3: Define Strategy and Metrics

Repeat this analysis for as many cities as possible.

New York, NY	
Italian	Mexican
Restaurant	Restaurant
Restaurant	Restaurant
Restaurant	VS. Restaurant
Restaurant	Restaurant
Restaurant	Restaurant

Tucson, AZ	
Italian	Mexican
Restaurant	Restaurant
Restaurant	Restaurant
Restaurant	VS. Restaurant
Restaurant	Restaurant
Restaurant	Restaurant

Washington, D.C.	
Italian	Mexican
Restaurant	Restaurant
Restaurant	Restaurant
Restaurant	VS. Restaurant
Restaurant	Restaurant
Restaurant	Restaurant

Omaha, NE	
Italian	Mexican
Restaurant	Restaurant
Restaurant	Restaurant
Restaurant	VS. Restaurant
Restaurant	Restaurant
Restaurant	Restaurant

San Diego, CA	
Italian	Mexican
Restaurant	Restaurant
Restaurant	Restaurant
Restaurant	VS. Restaurant
Restaurant	Restaurant
Restaurant	Restaurant

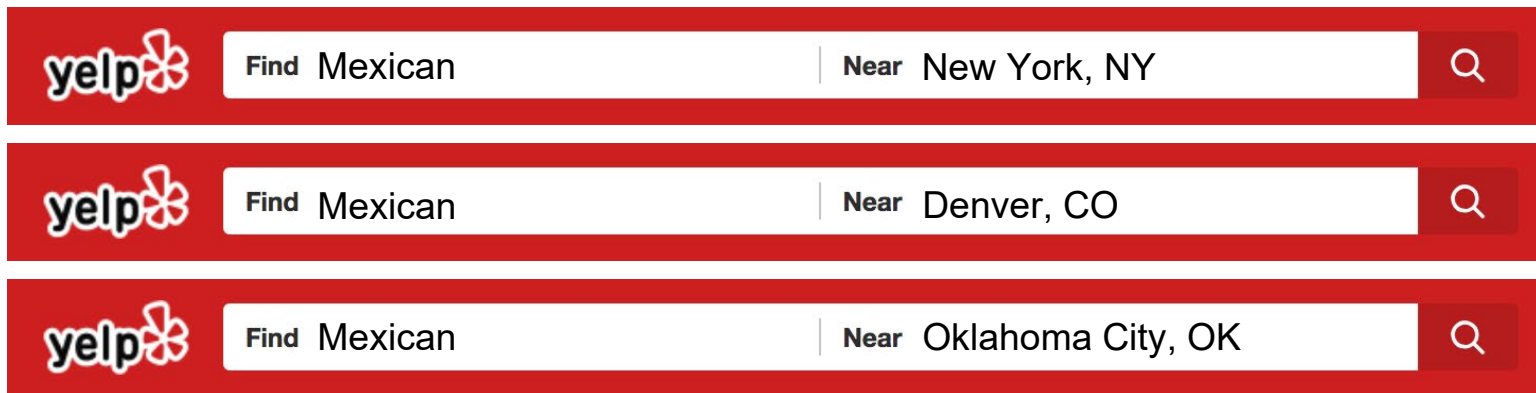
Atlanta, GA	
Italian	Mexican
Restaurant	Restaurant
Restaurant	Restaurant
Restaurant	VS. Restaurant
Restaurant	Restaurant
Restaurant	Restaurant

Step 4: Build Data Retrieval Plan

Step 4: Build Data Retrieval Plan

We could retrieve this data by brute force, but it would be:

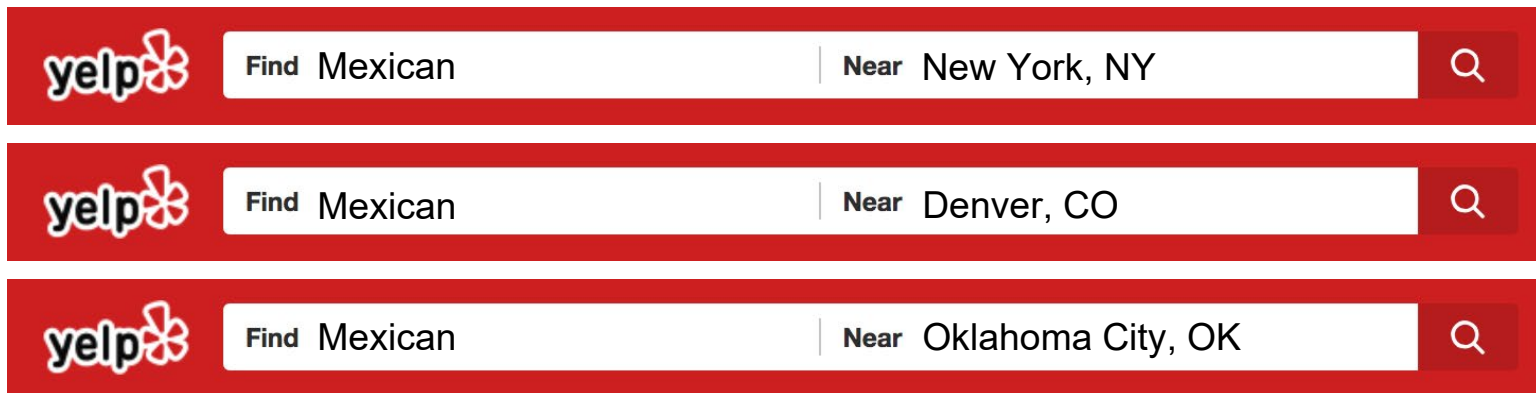
- Extremely time consuming
- Skewed by our city familiarity
- Labor intensive



The image displays three identical Yelp search bars stacked vertically. Each bar has a red background and a white search input area. The search input area is divided into two sections by a vertical line. The left section contains the text 'Find Mexican' and the right section contains the text 'Near New York, NY' (for the top bar), 'Near Denver, CO' (for the middle bar), and 'Near Oklahoma City, OK' (for the bottom bar). To the left of the search input area is the Yelp logo, and to the right is a magnifying glass icon. This visualizes the brute force approach of searching for Mexican restaurants in multiple cities.

Step 4: Build Data Retrieval Plan

Basically, it would be nearly impossible.

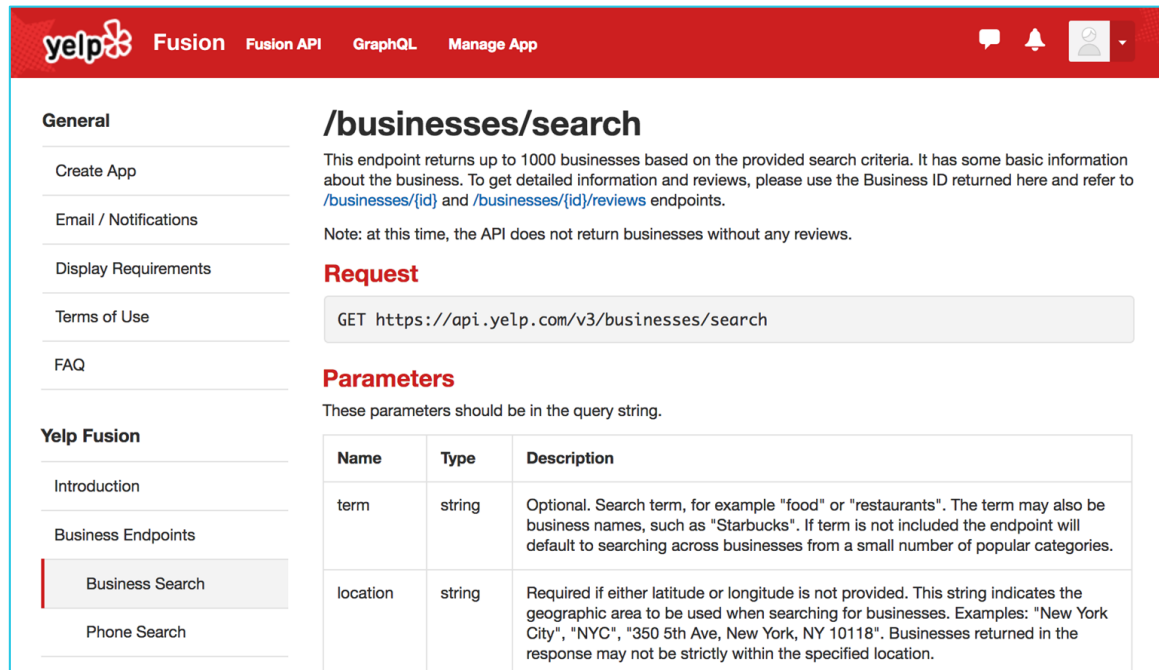


The image displays three identical Yelp search interface elements stacked vertically. Each element consists of a red horizontal bar. On the left of each bar is the Yelp logo. To the right of the logo is a white search input field divided into two sections by a vertical line. The left section of the input field contains the text 'Find Mexican'. The right section contains the text 'Near' followed by a city and state. To the right of the input field is a red square button with a white magnifying glass icon. The three search bars represent the following locations: New York, NY; Denver, CO; and Oklahoma City, OK.

Find	Near
Mexican	New York, NY
Mexican	Denver, CO
Mexican	Oklahoma City, OK

Thank You, Yelp!

Thankfully, we can take advantage of the **Yelp Fusion API** to **programmatically** run our queries. (#ThankGoodnessForProgramming)



The screenshot shows the Yelp Fusion API documentation page. The top navigation bar is red with the Yelp logo and links for Fusion, Fusion API, GraphQL, and Manage App. On the right are icons for chat, notifications, and a user profile. The left sidebar contains a 'General' section with links like 'Create App', 'Email / Notifications', 'Display Requirements', 'Terms of Use', and 'FAQ'. Below that is a 'Yelp Fusion' section with 'Introduction', 'Business Endpoints' (where 'Business Search' is highlighted), and 'Phone Search'. The main content area is titled '/businesses/search' and includes a description of the endpoint, a 'Request' section with the URL 'GET https://api.yelp.com/v3/businesses/search', and a 'Parameters' section with a table of query parameters.

General

- Create App
- Email / Notifications
- Display Requirements
- Terms of Use
- FAQ

Yelp Fusion

- Introduction
- Business Endpoints
 - Business Search
 - Phone Search

/businesses/search

This endpoint returns up to 1000 businesses based on the provided search criteria. It has some basic information about the business. To get detailed information and reviews, please use the Business ID returned here and refer to </businesses/{id}> and </businesses/{id}/reviews> endpoints.

Note: at this time, the API does not return businesses without any reviews.

Request

```
GET https://api.yelp.com/v3/businesses/search
```

Parameters

These parameters should be in the query string.

Name	Type	Description
term	string	Optional. Search term, for example "food" or "restaurants". The term may also be business names, such as "Starbucks". If term is not included the endpoint will default to searching across businesses from a small number of popular categories.
location	string	Required if either latitude or longitude is not provided. This string indicates the geographic area to be used when searching for businesses. Examples: "New York City", "NYC", "350 5th Ave, New York, NY 10118". Businesses returned in the response may not be strictly within the specified location.

Thank You, Yelp!

Response Body

```
{
  "total": 8228,
  "businesses": [
    {
      "rating": 4,
      "price": "$",
      "phone": "+14152520800",
      "id": "four-barrel-coffee-san-francisco",
      "is_closed": false,
      "categories": [
        {
          "alias": "coffee",
          "title": "Coffee & Tea"
        }
      ],
      "review_count": 1738,
      "name": "Four Barrel Coffee",
      "url": "https://www.yelp.com/biz/four-barrel-coffee-san-francisco",
      "coordinates": {
        "latitude": 37.7670169511878,
        "longitude": -122.42184275
      },
      "image_url": "http://s3-media2.fl.yelpcdn.com/bphoto/MmgtASP3l_t4tPCLiAsCg/o.jpg",
      "location": {
        "city": "San Francisco",
        "country": "US",
        "address2": "",
        "address3": "",
        "state": "CA",
        "address1": "375 Valencia St",
        "zip_code": "94103"
      },
      "distance": 1604.23,
      "transactions": ["pickup", "delivery"]
    },
    // ...
  ],
  "region": {
    "center": {
      "latitude": 37.767413217936834,
      "longitude": -122.42820739746094
    }
  }
}
```



Step 4: Build Data Retrieval Plan

We will build a Python script to randomly select over 700 zip codes from the U.S. Census, and then acquire review data from the top 20 Mexican and Italian restaurants for each zip code using the Yelp API.



11101	
Italian	Mexican
Restaurant	Restaurant
Restaurant	Restaurant
Restaurant	Restaurant
Restaurant	Restaurant
Restaurant	Restaurant

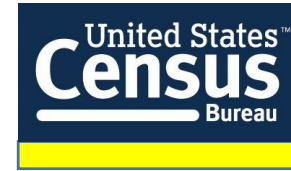
07360	
Italian	Mexican
Restaurant	Restaurant
Restaurant	Restaurant
Restaurant	Restaurant
Restaurant	Restaurant
Restaurant	Restaurant

20001	
Italian	Mexican
Restaurant	Restaurant
Restaurant	Restaurant
Restaurant	Restaurant
Restaurant	Restaurant
Restaurant	Restaurant

68007	
Italian	Mexican
Restaurant	Restaurant
Restaurant	Restaurant
Restaurant	Restaurant
Restaurant	Restaurant
Restaurant	Restaurant

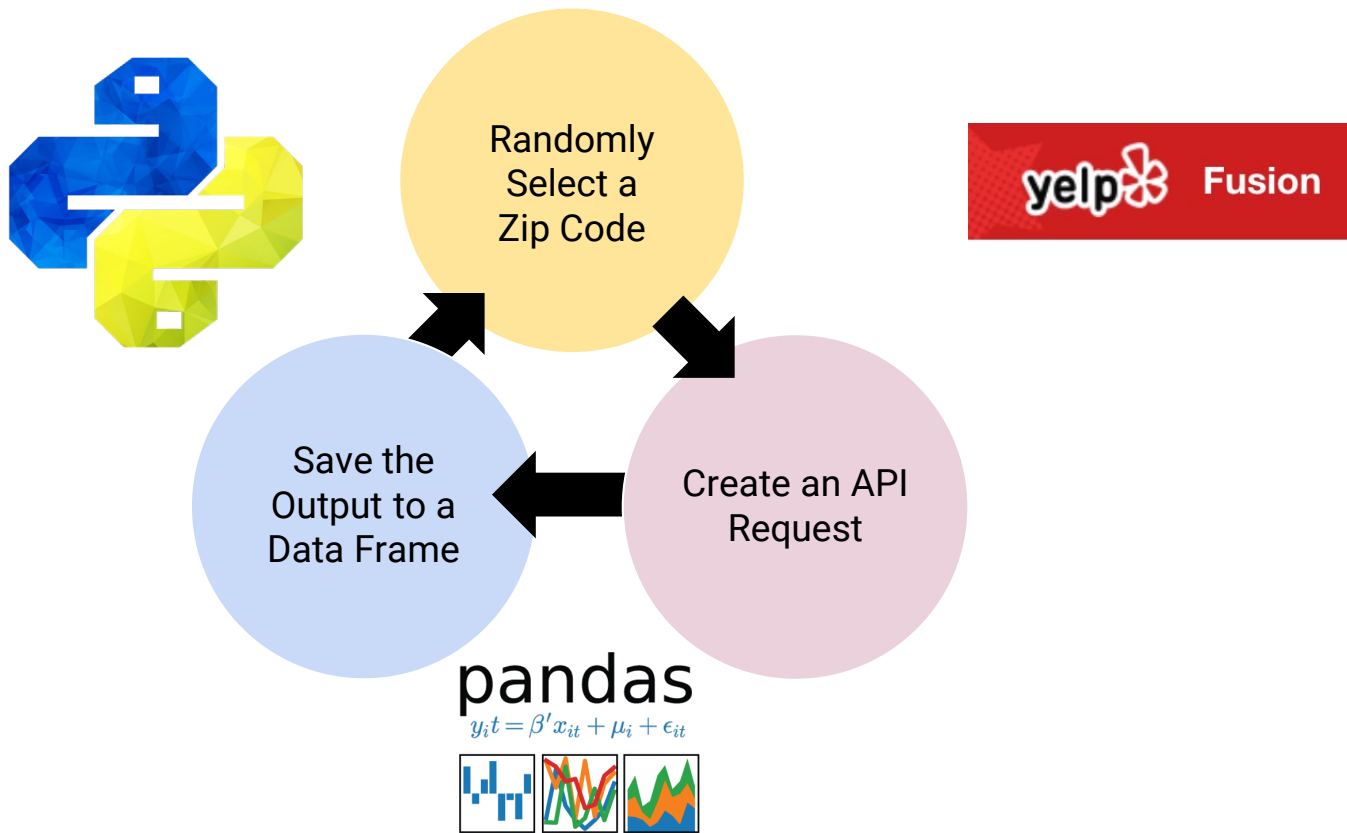
22434	
Italian	Mexican
Restaurant	Restaurant
Restaurant	Restaurant
Restaurant	Restaurant
Restaurant	Restaurant
Restaurant	Restaurant

30301	
Italian	Mexican
Restaurant	Restaurant
Restaurant	Restaurant
Restaurant	Restaurant
Restaurant	Restaurant
Restaurant	Restaurant



Step 5: Retrieve the Data

Pulling with Python



Pulling with Python

```
# Use Try-Except to handle errors
try:

    # Loop through all records to calculate the review count and weighted review value
    for business in yelp_reviews_italian["businesses"]:

        italian_review_count = italian_review_count + business["review_count"]
        italian_weighted_review = italian_weighted_review + business["review_count"] * business["rating"]

    for business in yelp_reviews_mexican["businesses"]:
        mexican_review_count = mexican_review_count + business["review_count"]
        mexican_weighted_review = mexican_weighted_review + business["review_count"] * business["rating"]

    # Append the data to the appropriate column of the data frames
    italian_data.set_value(index, "Zip Code", row["Zipcode"])
    italian_data.set_value(index, "Italian Review Count", italian_review_count)
    italian_data.set_value(index, "Italian Average Rating", italian_weighted_review / italian_review_count)
    italian_data.set_value(index, "Italian Weighted Rating", italian_weighted_review)

    mexican_data.set_value(index, "Zip Code", row["Zipcode"])
    mexican_data.set_value(index, "Mexican Review Count", mexican_review_count)
    mexican_data.set_value(index, "Mexican Average Rating", mexican_weighted_review / mexican_review_count)
    mexican_data.set_value(index, "Mexican Weighted Rating", mexican_weighted_review)

except:
    print("Uh oh")
```



This funky code...

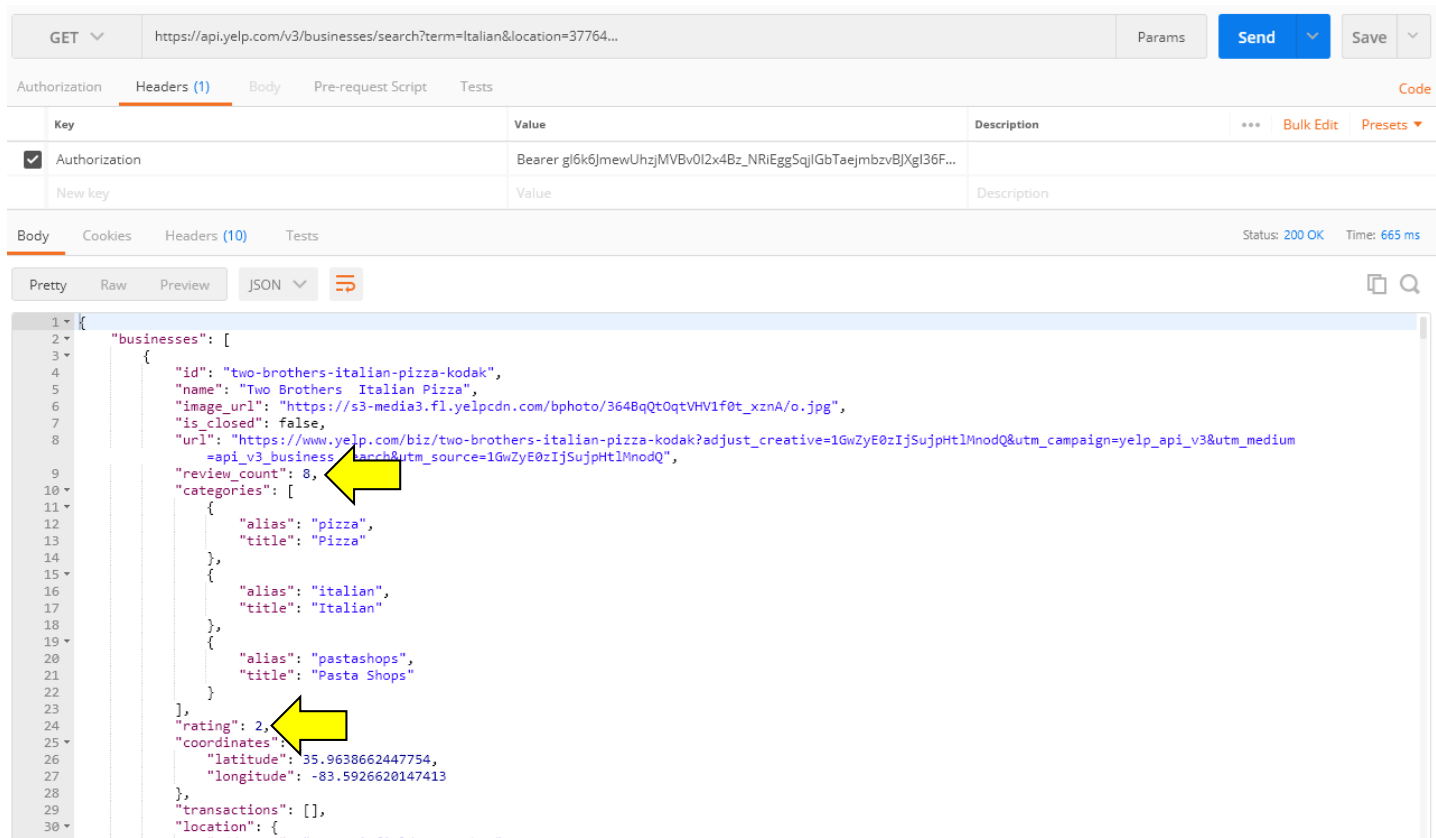
Pulling with Python

```
1
https://api.yelp.com/v3/businesses/search?term=Italian&location=76556
https://api.yelp.com/v3/businesses/search?term=Mexican&location=76556
2
https://api.yelp.com/v3/businesses/search?term=Italian&location=72039
https://api.yelp.com/v3/businesses/search?term=Mexican&location=72039
3
https://api.yelp.com/v3/businesses/search?term=Italian&location=61606
https://api.yelp.com/v3/businesses/search?term=Mexican&location=61606
4
https://api.yelp.com/v3/businesses/search?term=Italian&location=47232
https://api.yelp.com/v3/businesses/search?term=Mexican&location=47232
5
https://api.yelp.com/v3/businesses/search?term=Italian&location=60565
https://api.yelp.com/v3/businesses/search?term=Mexican&location=60565
6
https://api.yelp.com/v3/businesses/search?term=Italian&location=20634
https://api.yelp.com/v3/businesses/search?term=Mexican&location=20634
7
https://api.yelp.com/v3/businesses/search?term=Italian&location=71046
https://api.yelp.com/v3/businesses/search?term=Mexican&location=71046
```



**...will make all of
these URLs.**

Pulling with Python



The screenshot shows a REST client interface with a GET request to `https://api.yelp.com/v3/businesses/search?term=Italian&location=37764...`. The response is a JSON object containing a list of businesses. Two yellow arrows point to specific fields in the JSON: `"review_count": 8` and `"rating": 2`.

```
{
  "businesses": [
    {
      "id": "two-brothers-italian-pizza-kodak",
      "name": "Two Brothers Italian Pizza",
      "image_url": "https://s3-media3.fl.yelpcdn.com/bphoto/3648qQt0qtVHV1f0t_xznA/o.jpg",
      "is_closed": false,
      "url": "https://www.yelp.com/biz/two-brothers-italian-pizza-kodak?adjust_create=1GwZyE0zIjSujpHt1MnodQ&utm_campaign=yelp_api_v3&utm_medium=api_v3_business_search&utm_source=1GwZyE0zIjSujpHt1MnodQ",
      "review_count": 8,
      "categories": [
        {
          "alias": "pizza",
          "title": "Pizza"
        },
        {
          "alias": "italian",
          "title": "Italian"
        },
        {
          "alias": "pastashops",
          "title": "Pasta Shops"
        }
      ],
      "rating": 2,
      "coordinates": {
        "latitude": 35.9638662447754,
        "longitude": -83.5926620147413
      },
      "transactions": [],
      "location": {

```



Each of these URLs holds a piece of our answer.

Step 6: Assemble and Clean the Data

Cleaning with Pandas

No data comes out intrinsically the way you want it to.
In our case, we needed multiple steps to aggregate the data along our channels of interest.

```
# Combine DataFrames into a single DataFrame
combined_data = pd.merge(mexican_data, italian_data, on="Zip Code")
combined_data.head( )
```

	Zip Code	Mexican Review Count	Mexican Average Rating	Mexican Weighted Rating	Italian Review Count	Italian Average Rating	Italian Weighted Rating
0	76556	97	4.1134	399	63	3.78571	238.5
1	72039	256	4.11133	1052.2	266	3.81955	1016
2	61606	378	3.64286	1377	66	3.2197	212.5
3	47232	222	4.16892	925.5	420	3.77857	1587
4	60565	2842	3.94053	11199	2829	3.92824	11113

Step 7: Analyze for Trends

Analyze for Trends (Table)

It's Close:

Display Summary of Results

```
# Model 1: Head-to-Head Review Counts
italian_summary = pd.DataFrame({"Review Counts": italian_data["Italian Review Count"].sum(),
                                "Rating Average": italian_data["Italian Average Rating"].mean(),
                                "Review Count Wins": combined_data["Review Count Wins"].value_counts()["Italian"],
                                "Rating Wins": combined_data["Rating Wins"].value_counts()["Italian"]}, index=["Italian"])

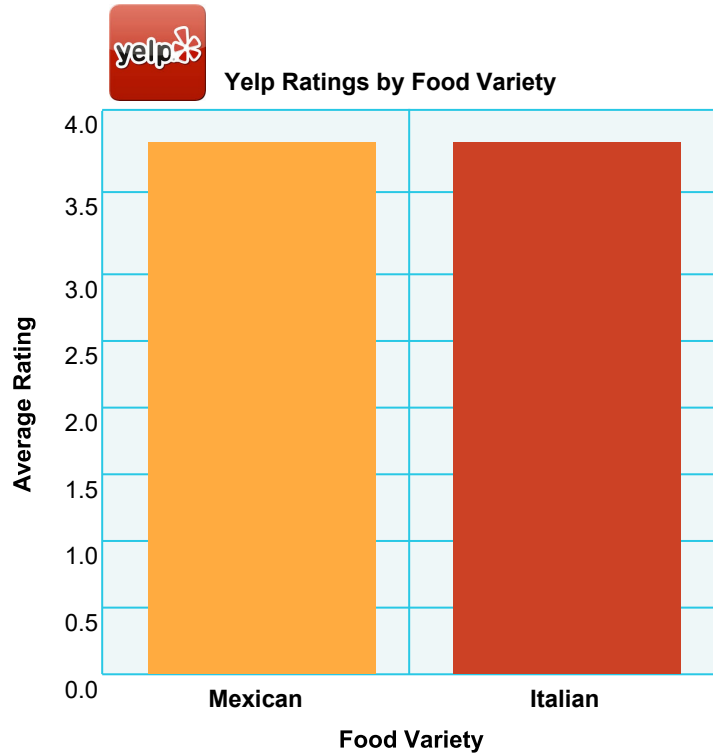
mexican_summary = pd.DataFrame({"Review Counts": mexican_data["Mexican Review Count"].sum(),
                                "Rating Average": mexican_data["Mexican Average Rating"].mean(),
                                "Review Count Wins": combined_data["Review Count Wins"].value_counts()["Mexican"],
                                "Rating Wins": combined_data["Rating Wins"].value_counts()["Mexican"]}, index=["Mexican"])

final_summary = pd.concat([mexican_summary, italian_summary])
final_summary
```

	Rating Average	Rating Wins	Review Count Wins	Review Counts
Mexican	3.826588	273	220	476889
Italian	3.806869	245	298	573733

Analyze for Trends (Ratings)

Yelpers rate Italian and Mexican relatively **equally**.

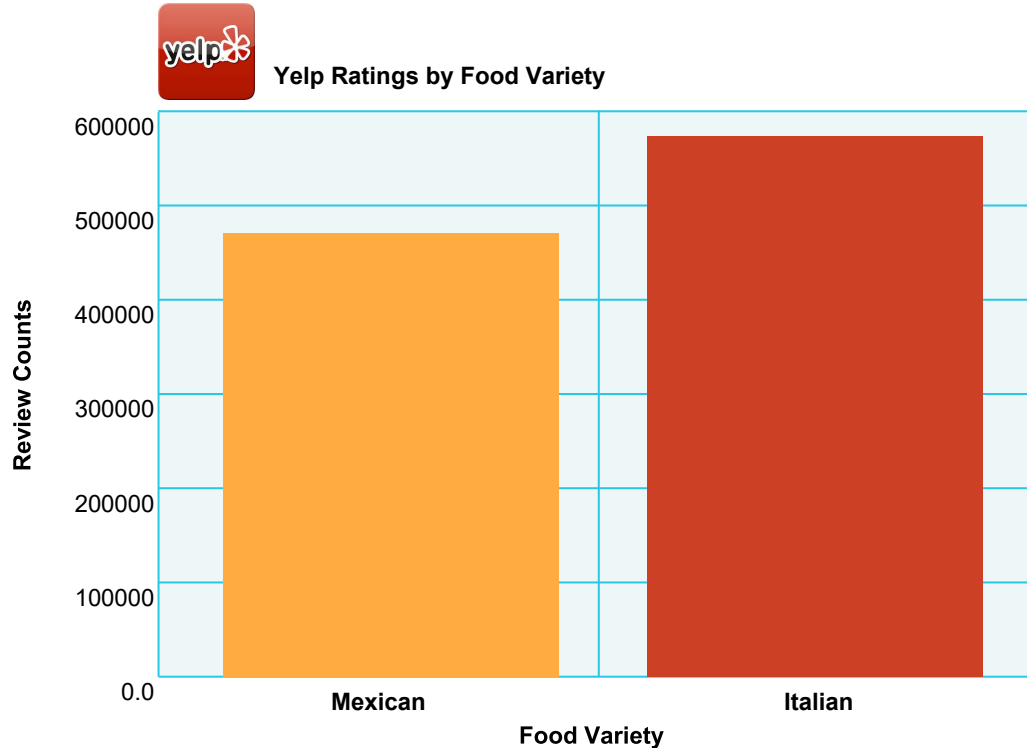


=



Analyze for Trends (Ratings)

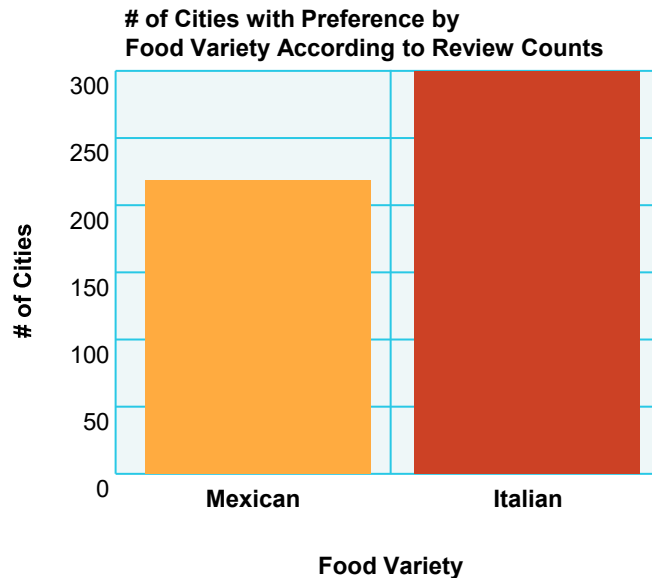
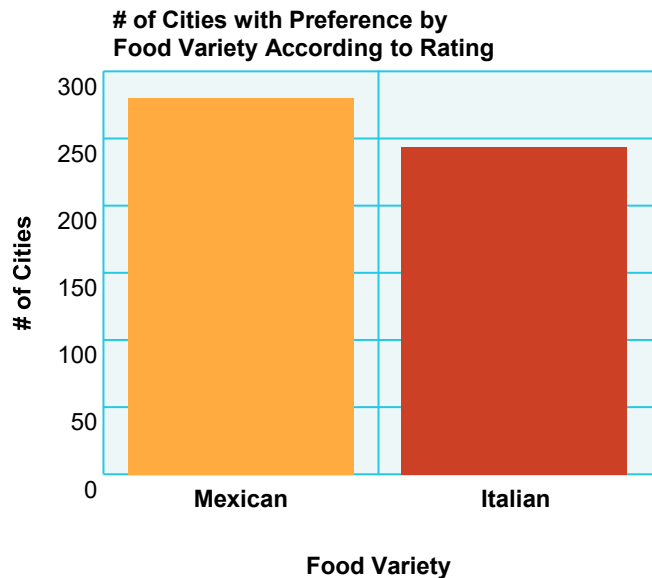
Yelpers seem to significantly **review more Italian** restaurants.



Analyze for Trends (Winner Take All)

Just for kicks, let's throw in an analysis that aggregates the data from all cities using a winner-take-all approach.

It's sort of a wash.



Analyze for Trends (Statistical Analysis)

Because of how close the numbers appear, we utilized a Student's t-test to quickly assess if the perceived differences are not statistically significant but could be considered substantial.

Metric	Italian	Mexican	p-Value (t-test)
Average Rating	3.806	3.826	0.284
Review Counts	573k	476k	0.057



The difference in review count is **not statistically significant**.

Step 8: Acknowledge Limitations

Limitations of Analysis

Yelp demographics may not match the American demographic.



Limitations of Analysis

Restaurant experiences do not equate to home-cooked meals.



Limitations of Analysis

Fine-dining effect?



Step 9: Make the Call

Making the Call

The “Proper” Conclusion:

Based on our analysis, it's clear that Americans' preferences for Italian and Mexican food are similar in nature. As a whole, Americans rate Mexican and Italian restaurants at non-statistically similar scores (avg. score: 3.8, p-value: 0.285). Although there are more reviews for Italian restaurants, we have shown that the difference is statistically significant (+96k, p-value: 0.057).



This may indicate there is an increased interest in visiting Italian restaurants at an experiential level. Or it may merely suggest that Yelp users enjoy writing reviews of Italian restaurants more than Mexican restaurants.

Making the Call

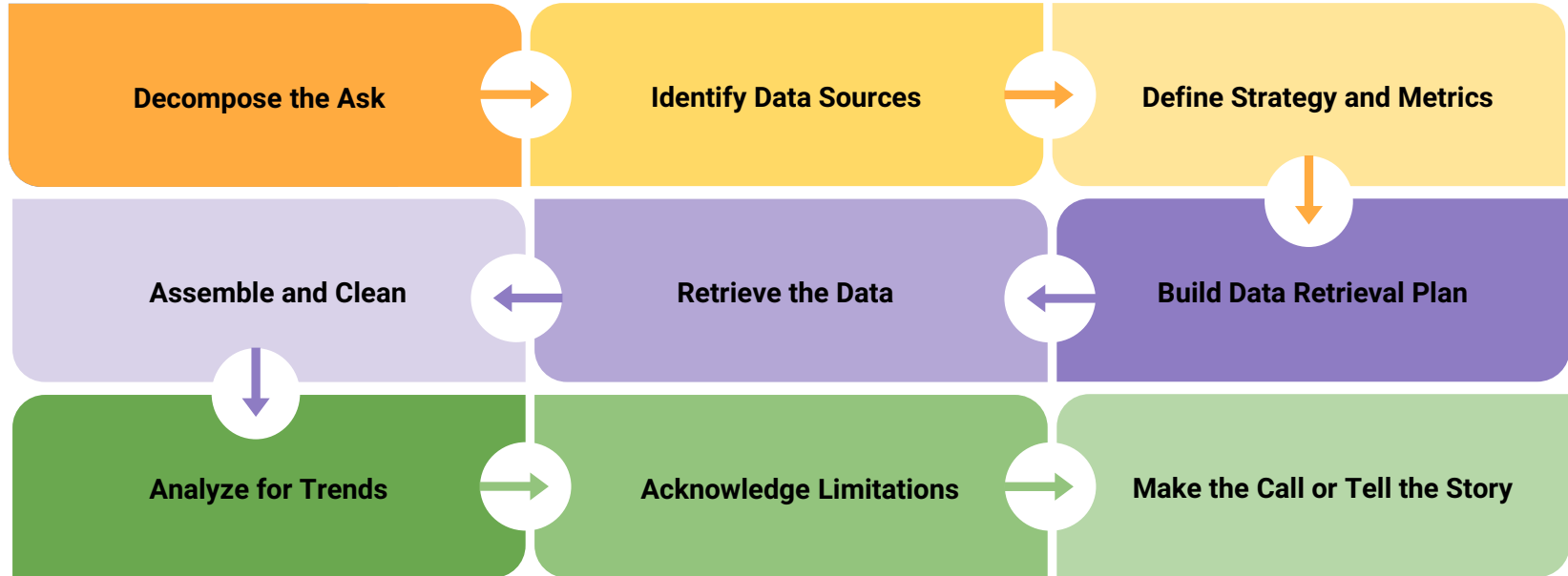
The “Let’s Be Real” Conclusion: Italian (but it’s going to be close).



An Analytics Paradigm

Analytics Paradigm

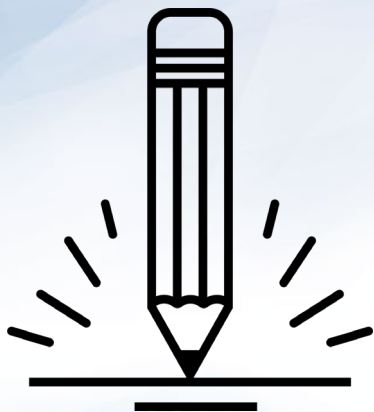
Regardless of type or industry, this paradigm provides a repeatable pathway for effective data problem solving.



Homework:
Kickstart My Chart
Will be posted Saturday



Questions?



If Time Permits...

Optional Group Activity: Predicting Gentrification

Using the Analytics Paradigm as a framework, outline a strategy by which you would identify which neighborhoods in our city are seeing signs of gentrification.

Suggested Time:
10 minutes



Group Activity: Predicting Gentrification

Specifically, how would you answer these questions:



What observable signs can we detect to suggest gentrification is happening?



What means can we use to determine how long the trend has been happening?



What proxies might we use to identify gentrification in non-obvious ways?



How might you create a visualization of this data to best “tell the story”?

Pay special attention to details like:



What data will you use to build your model?



How will you retrieve the data?



What does your final “story” look like?

Suggested Time: 10 minutes





Time's Up! Let's Review.