Extracting Insights from Amazon.com Customer Reviews

Ryan Burakowski

NYC Data Science Academy

May 24th, 2021

Project Statement

- Analyze the distribution characteristics of online reviews, at the product/service level.
- Investigate how many reviews a product/service needs to be meaningful.
- Build a tool to help businesses extract meaning from the customer reviews of their products/services
- Narrowing the scope to focus on the world's largest online retail marketplace, Amazon.com.

```
    output: 90% confidence interval as a float
    '''
    sample_df = pd.read_csv(r'rough_data\amazon_dataset\amzn_confidence_interval_dataset.csv')

    output_list = []
    for ind in range(5000):
        review_subset = np.random.choice(sample_df['rating'], num_reviews, replace=False)
        avg = review_subset.mean()
        output_list.append(avg)

    series_out = pd.Series(output_list)
    # Standard dev of the samples
    std = np.std(series_out)
    # z-score for 90% confidence interval
    z_score = 1.645
    #90% confidence interval
    c_i_range = z_score * std
    return round(c_i_range,3)


#Complete
def analysis_results(c_i_range, average_rating):
    '''
    Inputs: user list anaysis mtrics, sample analysis metrics
    Output: %-tile for bottom of 90% confidence interval, for average
    rating, and for top of 90% confidence interval.
    '''
    cum_freq_df = pd.read_csv(r'rough_data\amazon_dataset\amzn300_cum_freq_table.csv')
    # Confidence interval endpoints, and average, rounded to align with the
        # values in the frequency table and bounded by the table limits.
    low_c_i = max(round((average_rating - c_i_range)/2,2)*2,1.02)
    high_c_i = min(round((average_rating + c_i_range)/2,2)*2,5)
    average_rating = round(average_rating/2,2)*2

    # Retrieve the corresponding cumulative frequency
    low_ci_rating = cum_freq_df[cum_freq_df['star_rating'] \
        == low_c_i].iloc[0,1]
    average_rating_num = cum_freq_df[cum_freq_df['star_rating'] \
        == average_rating].iloc[0,1]
    high_ci_rating = cum_freq_df[cum_freq_df['star_rating'] \
        == high_c_i].iloc[0,1]
    # Turn them into percents
    low_ci_percent = round(low_ci_rating*100)
    average_rating_percent = round(average_rating_num*100)
    high_ci_percent = round(high_ci_rating*100)
    return low_ci_percent, average_rating_percent, high_ci_percent
```

Agenda

1. The Problem

How to find meaning in online customer reviews?

2. Data & Analysis

Source of data.
Analysis to conduct.

3. Project Results

Results of analysis and why they are important.

4. Tool Demonstration

Bringing it all together for a purpose.

5. Future Work / Q&A

Extensibility.
Additional analysis.

The Problem

What do Online Customer Ratings Really Tell Us?

- 3 / 5 Stars is not ‘average’
- Amazon
- Airbnb
- Yelp
- Seamless

How can Companies Use Online Review Information?

- Known selection bias issues with self-reporting data.
- What to do with this biased / skewed data?

How to Gain Actual Insight?

Calculating a number is easy:

- *Does it have meaning?*
- *Is it robust?*
- *Is it statistically significant?*

Background: Online Customer Reviews

Biases

- Polarity self-selection
- Purchase self-selection

Distribution

- High level of polarity
- Large positive imbalance
- On user-level, depend on number of reviews written
- On platform level, affected by scale ratings, social networking aspects, fee vs. informational platform.

Prior Research

- Entire universe of online reviews
- Reviews by platform
- Platform to platform variation
- Scale effects
- Scarce analysis of reviews grouped at the product level

Great research paper about online customer reviews, for those interested:
The Polarity of Online Reviews: Prevalence, Drivers and Implications (Schoenmuller, Netzer, Stahl)

Project Statement

- Analyze the distribution characteristics of online reviews, at the product/service level.
- Investigate how many reviews a product/service needs to be meaningful.
- Build a tool to help businesses extract meaning from the customer reviews of their products/services
- Narrowing the scope to focus on the world's largest online retail marketplace, Amazon.com.

```
    output: 90% confidence interval as a float
    '''
    sample_df = pd.read_csv(r'rough_data\amazon_dataset\amzn_confidence_interval_dataset.csv')

    output_list = []
    for ind in range(5000):
        review_subset = np.random.choice(sample_df['rating'], num_reviews, replace=False)
        avg = review_subset.mean()
        output_list.append(avg)

    series_out = pd.Series(output_list)
    # Standard dev of the samples
    std = np.std(series_out)
    # z-score for 90% confidence interval
    z_score = 1.645
    #90% confidence interval
    c_i_range = z_score * std
    return round(c_i_range,3)


#Complete
def analysis_results(c_i_range, average_rating):
    '''
    Inputs: user list anaysis mtrics, sample analysis metrics
    Output: %-tile for bottom of 90% confidence interval, for average
    rating, and for top of 90% confidence interval.
    '''
    cum_freq_df = pd.read_csv(r'rough_data\amazon_dataset\amzn300_cum_freq_table.csv')
    # Confidence interval endpoints, and average, rounded to align with the
        # values in the frequency table and bounded by the table limits.
    low_c_i = max(round((average_rating - c_i_range)/2,2)*2,1.02)
    high_c_i = min(round((average_rating + c_i_range)/2,2)*2,5)
    average_rating = round(average_rating/2,2)*2

    # Retrieve the corresponding cumulative frequency
    low_ci_rating = cum_freq_df[cum_freq_df['star_rating'] \
        == low_c_i].iloc[0,1]
    average_rating_num = cum_freq_df[cum_freq_df['star_rating'] \
        == average_rating].iloc[0,1]
    high_ci_rating = cum_freq_df[cum_freq_df['star_rating'] \
        == high_c_i].iloc[0,1]
    # Turn them into percents
    low_ci_percent = round(low_ci_rating*100)
    average_rating_percent = round(average_rating_num*100)
    high_ci_percent = round(high_ci_rating*100)
    return low_ci_percent, average_rating_percent, high_ci_percent
```

The Data

Data Gathering

- 82M reviews of 9.8M products on Amazon.com between 1996-2014.
- Collected by Julian McAuley, Professor at UCSD.
- There is a newer data, tens of millions more reviews that runs through 2018. I could not gain access to the newer data in time for this project.

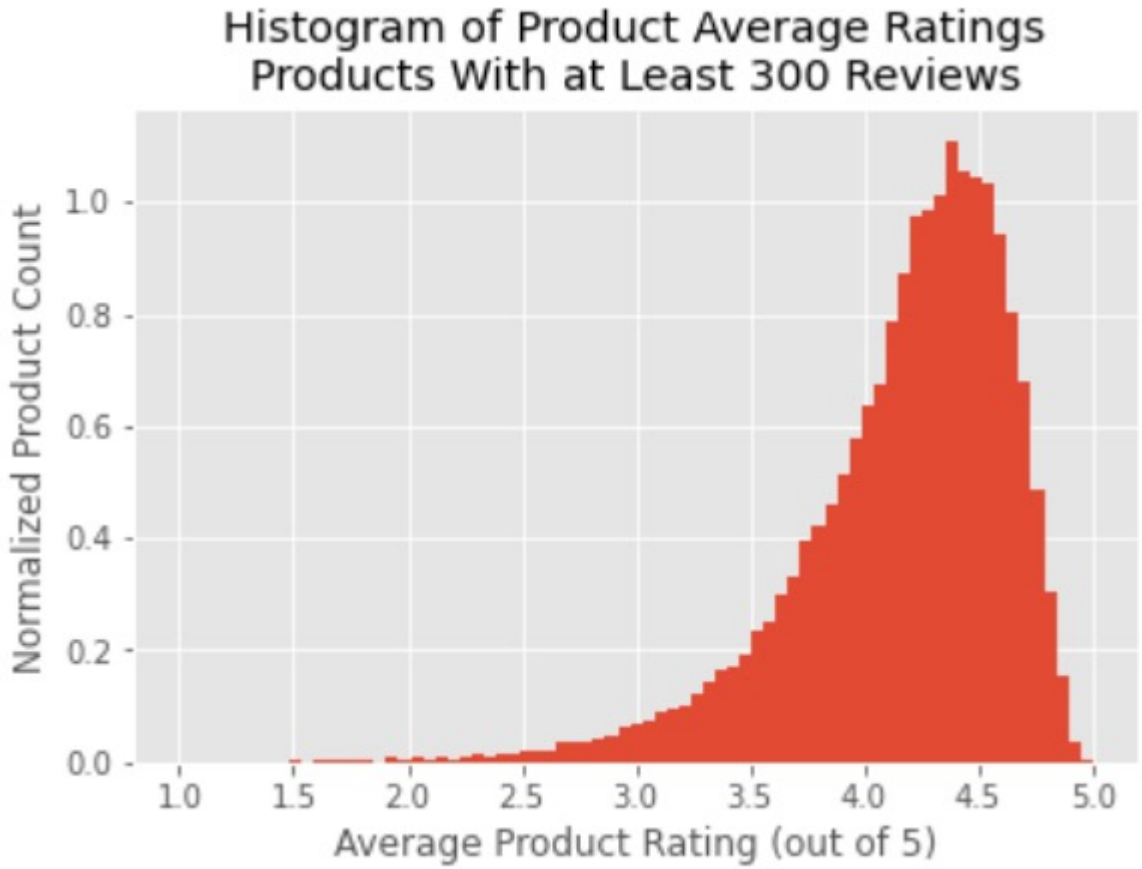
Data Transformation

- Grouped individual reviews by their unique item key.
- Create columns for the five star ratings, average rating, and total count.

	item	average_rating	total_ratings	5_star_ratings	4_star_ratings	3_star_ratings	2_star_ratings	1_star_ratings
0	B0054JZC6E	4.329313	25368.0	0.523810	0.330377	0.114436	0.014073	0.017305
1	B00FAPF5U0	4.369381	24024.0	0.637696	0.209832	0.079878	0.029346	0.043248
2	B009UX2YAC	4.674445	23956.0	0.788529	0.144056	0.038654	0.010853	0.017908
3	0439023483	4.644406	21398.0	0.770726	0.155529	0.039957	0.015001	0.018787
4	030758836X	3.794433	19867.0	0.388081	0.278351	0.160266	0.086525	0.086777

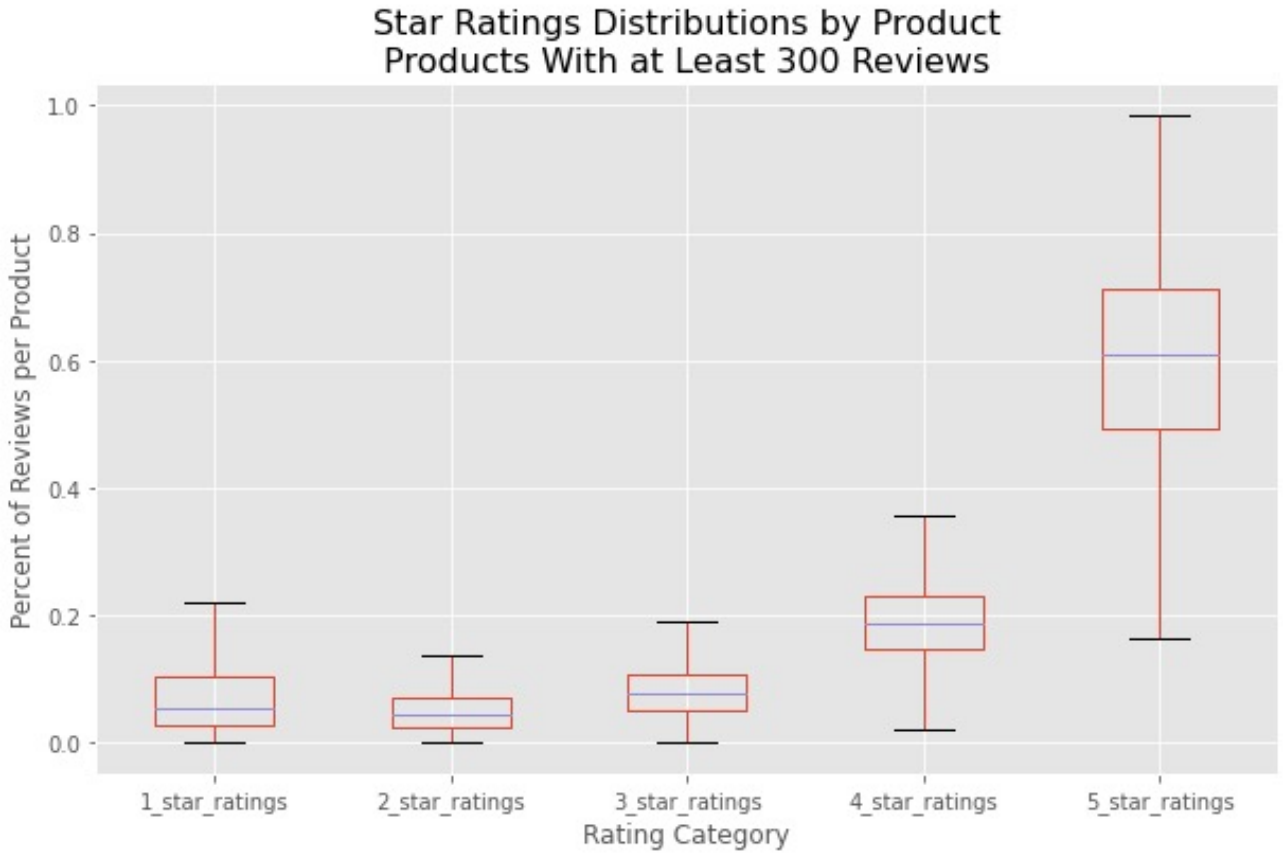
Product Distribution Analysis

- Average ratings for products create a well-ordered distribution.
- Distribution hardly changes at all for different minimum reviews/product level. Very robust.
- Approximates a normal distribution with a super-fat lower tail.



Product Distribution Analysis

- Average ratings for products create a well-ordered distribution.
- Distribution hardly changes at all for different minimum reviews/product level. Very robust.
- Approximates a normal distribution with a super-fat lower tail.



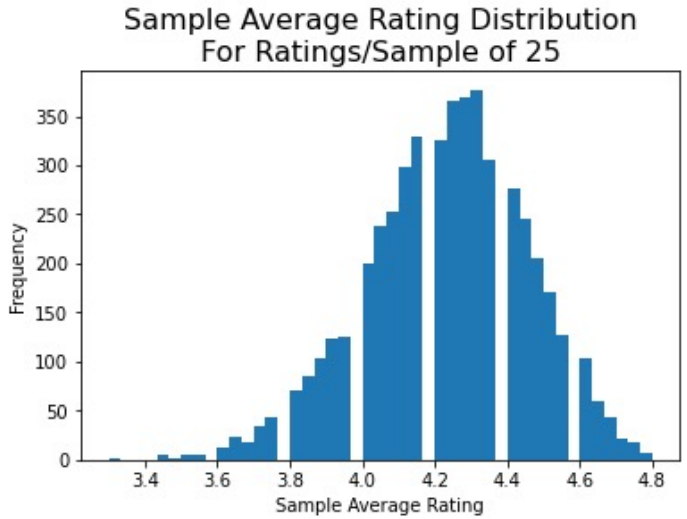
Product Distribution Analysis

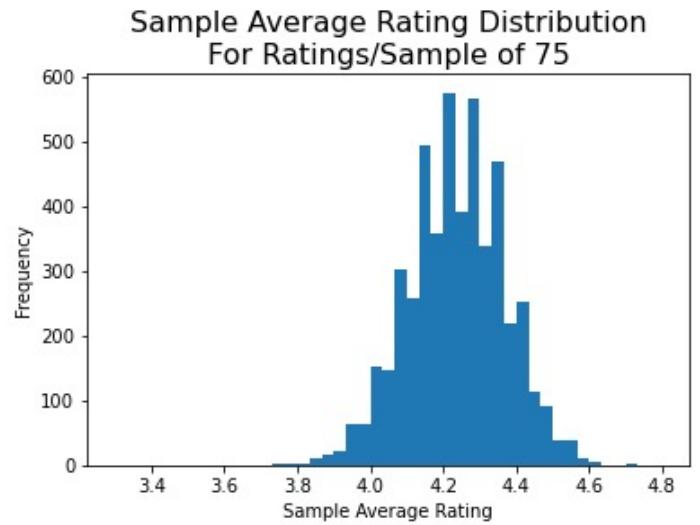
- Average ratings for products create a well-ordered distribution.
- Distribution hardly changes at all for different minimum reviews/product level. Very robust.
- Approximates a normal distribution with a super-fat lower tail.

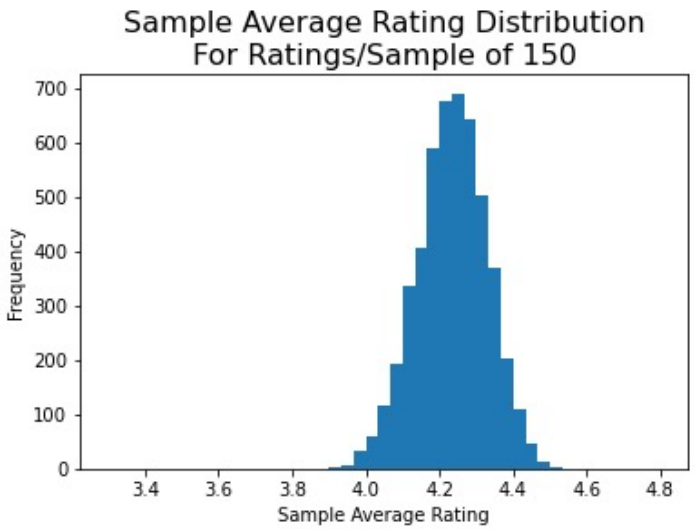
star_rating	cumulative_frequency
4.9	1.00
4.7	0.93
4.5	0.74
4.3	0.53
4.1	0.35
3.9	0.23
3.7	0.14
3.5	0.09
3.3	0.05
3.1	0.03
2.9	0.02
2.7	0.01
2.5	0.01
2.3	0.00
2.1	0.00
1.9	0.00
1.7	0.00
1.5	0.00
1.3	0.00
1.1	0.00

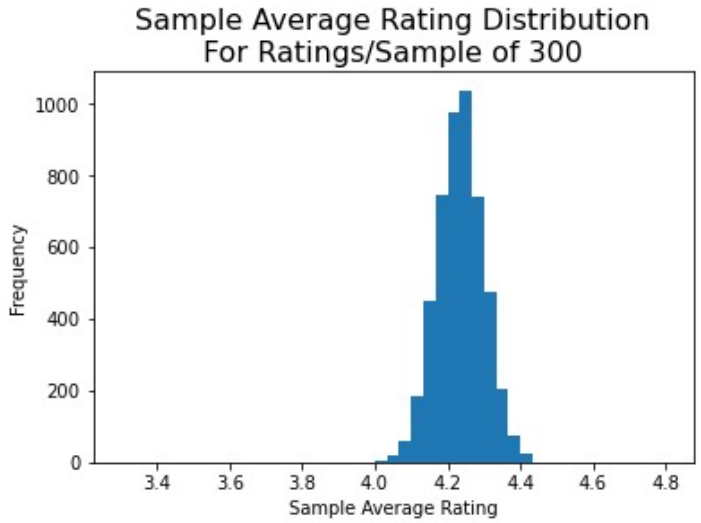
Sample Size Analysis

- How many reviews does it take to get a meaningful average for a single product?
- Conducted random sampling on a ‘typically’-distributed product with 14,114 reviews with different sample sizes.
- Computed 90% confidence intervals for these normal distributions of sample averages.
- For useable results, products should have 300+ reviews. Equates to 90% confidence interval of approximately +-0.10 stars or smaller from observed average.
- Ended idea of creating a custom scoring system using outlier frequency data, would need an even larger min review count.



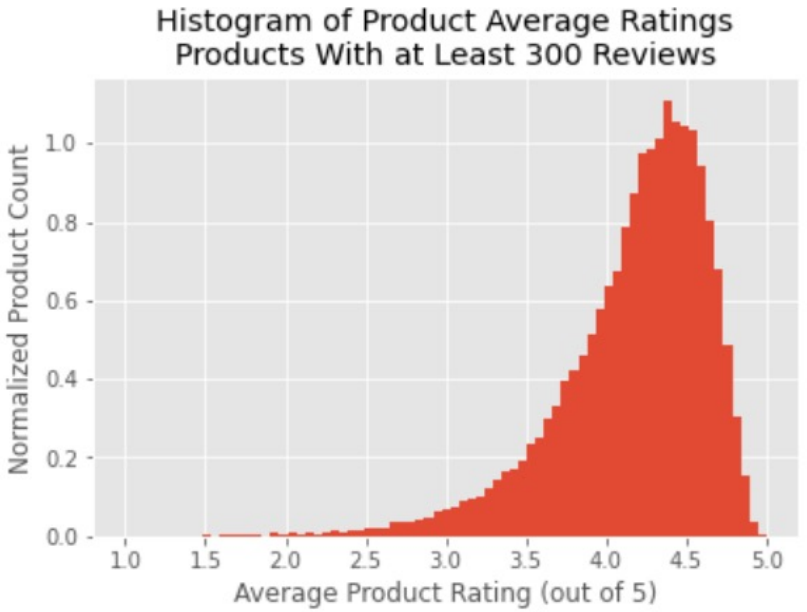


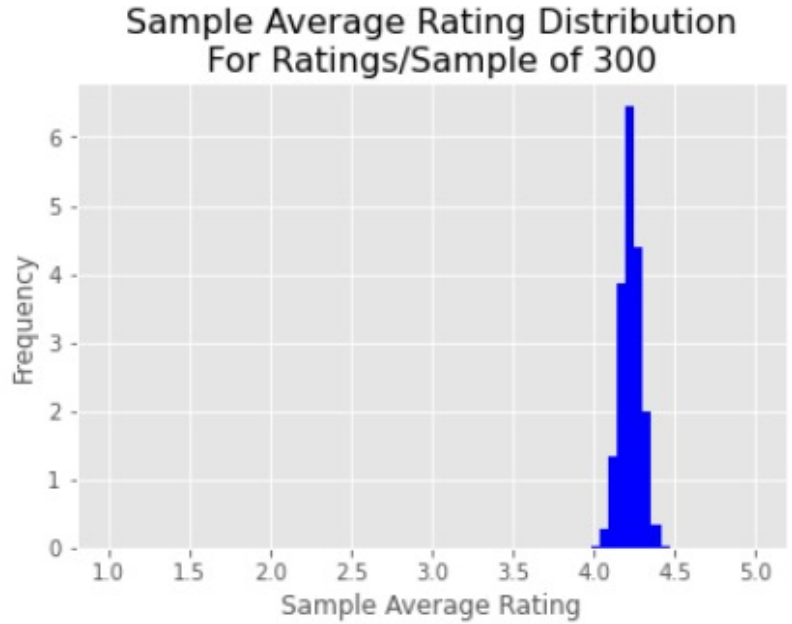




Sample Size Analysis

- How many reviews does it take to get a meaningful average for a single product?
- Conducted random sampling on a ‘typically’-distributed product with 14,114 reviews with different sample sizes.
- Computed 90% confidence intervals for these normal distributions of sample averages.
- For useable results, products should have 300+ reviews. Equates to 90% confidence interval of approximately +-0.10 stars or smaller from observed average.
- Ended idea of creating a custom scoring system using outlier frequency data, would need an even larger min review count.





Summary of Results

Average score distribution

For products with more than 300 reviews:

Mean product mean rating: 4.17
Median product mean rating: 4.27
Median 5-Star ratings: 61%
Median 1-Star ratings: 5%

Calculated cumulative frequency of product average ratings.

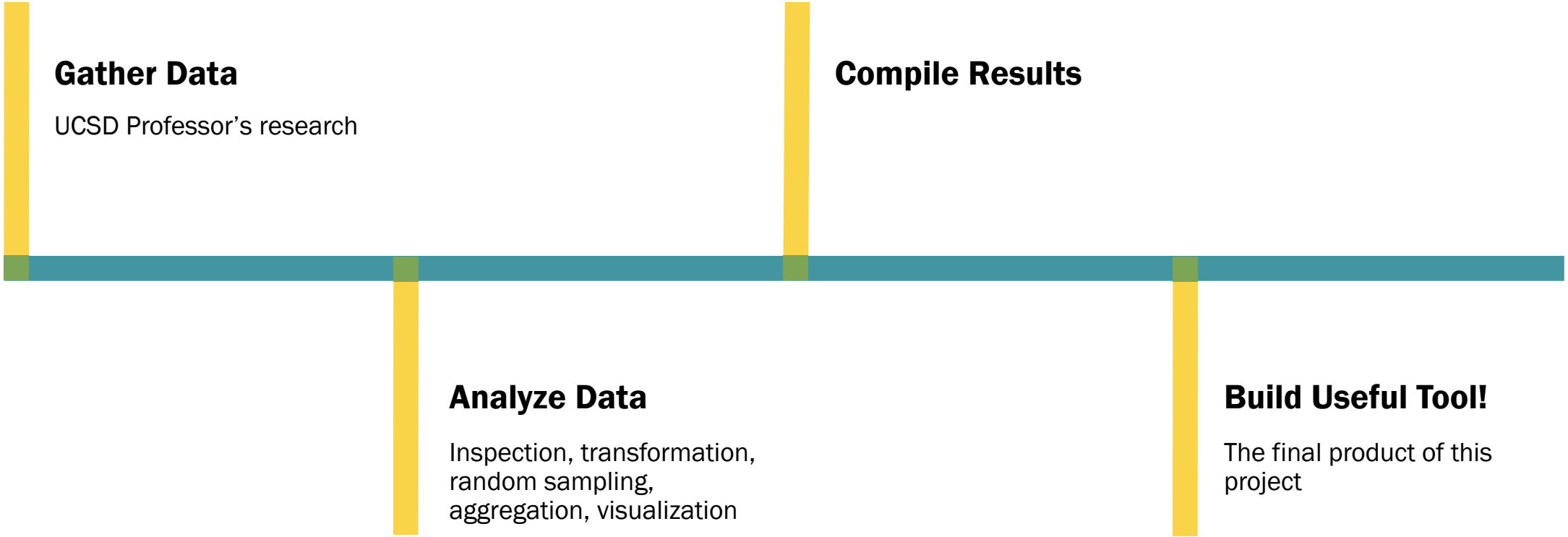
Sample size cutoff

- 300-review minimum for a product will yield results within a reasonable confidence interval of about +-0.10 stars.

With this info we can:

- Correlate a product's average rating to a percentile of all products on Amazon.
- Determine if this result is statistically meaningful, or subject to extreme variation as more customer reviews are written.

Process


Gather Data
UCSD Professor's research
Analyze Data
Inspection, transformation, random sampling, aggregation, visualization
Compile Results
Build Useful Tool!
The final product of this project

Construction of Reviews Analysis Tool

1. Generate concise data for tool

Small, compact datasets and information tables

2. Decide on desired output

Non-trivial issue… How to encourage appropriate use of tool by someone who doesn't fully understand it?

3. Create Python script for Tool

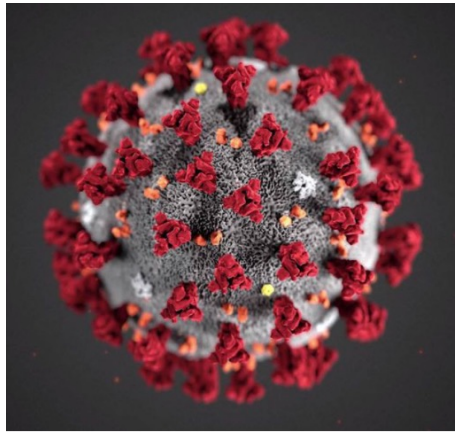
Incorporate code chunks from my analysis, and new code, into an interactive Python script.

4. Test/debug the script

Runs in the terminal.
Gets user input.
Returns output.

5. Demonstration!

Future Work



Newer dataset

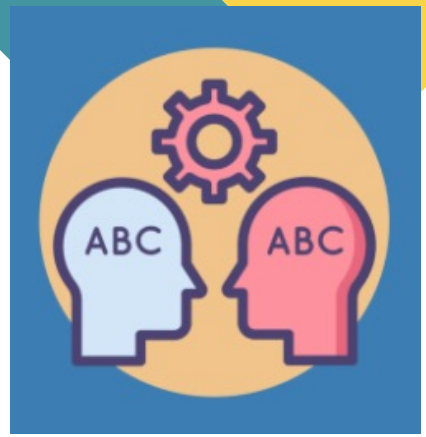
Accelerated change (COVID).
Consumer behavior.
Wider Demographic.
Online Shopping.

Extend to Services

Analyzed products, now try services.
Yelp is a top business review website.
Has rich dataset available for academic use.

Analyze New Domains

Do these results extend more generally to sites with product/services reviews?



Increased Utility

Go from descriptive to prescriptive using NLP for positive and negative individual reviews.

Questions?
Answers.