Impact of Food Delivery Services and Population Density on Yelp Restaurant Ratings

Project One

Introduction

Online review platforms play a significant role in influencing business performance during this era of digital connectivity and evolving customer behaviors, with Yelp standing out as one of the foremost review sites. Research by Anderson & Magruder (2012) indicates that an additional half-star rating on Yelp can lead to a substantial increase, of over 19 percentage points, in the frequency of prime-time table reservations at restaurants. This observation aligns with Luca's findings in 2011, underscoring the pivotal role that Yelp ratings (in the form of stars) play in shaping business outcomes, directly impacting revenue generation.

Yelp has its own rating system based on the ratings of each review,it involves a weighted average algorithm incorporating review quality, quantity, business rating, recency, and user behavior. The process includes automated filters to detect biased or fake reviews and categorical ratings for specific aspects like service and ambiance. Yelp's recommendation software showcases relevant reviews while excluding low-quality ones. It is worth mentioning that researchers predict star ratings through sentiment analysis of review texts, capturing emotional tones (Li & Zhang, 2014), while it is also considered as the review part in this paper.

My interest is firmly rooted in discerning the other factors that contribute to the determination of Yelp ratings indirectly, utilizing the dataset provided by the Yelp Dataset Challenge (2017). This dataset comprehensively encompasses vital information about Yelp businesses, including ratings, geographical locations, review counts, and categories, spanning across 11 metropolitan areas in four countries. This comprehensive dataset forms the cornerstone of my research investigation.

Having acquired the census dataset for each zip code and the map shape-file from the United States Census Bureau, my focus shifted to discerning crucial determinants aside from the factors of Yelp rating algorithm. Among these, population density surfaced as a compelling variable, supported by evidence indicating that heightened population density engenders a notable reduction in the spatial dispersion of both top-tier and lower-tier restaurants (Mossay et al., 2020). As increased population density often signifies a concentrated market where competition becomes more intense, potentially leading to enhanced quality control and a narrower range of restaurant performance, which could thus affect the restaurant's ratings.

Moving forward, the variable of household income assumes pivotal significance. Established scholarship posits its pivotal role in influencing not only the choice to dine at restaurants but also culinary preferences, quality perceptions, and nutritional decisions (Bellisle, 2006) (Mulamba, 2022). The economic rationale here lies in the notion that higher household incomes can provide individuals with the means to prioritize and afford dining out, thereby potentially shaping both their expectations and evaluations of restaurant experiences. While scholarly literature does not extensively corroborate the notion, it's worth noting Feefo's research findings. This research highlights the potential impact of reviewers' age and sex on their reviews, offering a glimpse into how demographic variables might subtly influence rating behaviors (Gorkana News, 2016). Economically speaking, age and sex can influence dining patterns, satisfaction thresholds, and the criteria individuals use to evaluate their experiences.

In this intricate web of factors, population density mirrors competitive dynamics of the restaurants; and in the consumer side, household income reflects economic capacity and preferences, while age and sex ratio hint at different groups' dining habits and satisfaction thresholds. All these variables, nestling within the census dataset of individual zip codes, are poised to act as independent drivers of Yelp restaurant ratings.

This study embarks on a parallel narrative exploration within the United States, inspired by research centred on China's ongoing urbanization during the recent years. The phenomenon in China has led to a surge in food delivery services and prompted traditional businesses to transition towards online platforms. Notably, it was found that more than one-fifth of China's total population has become part of the O2O food delivery market (Maimaiti et al., 2018). Urbanization, marked by population concentration in larger cities, typically brings about increases in population density and mean income. Drawing inspiration from this, the study aims to unravel a similar narrative within the context of the United States. The focus is on examining the intricate interplay of factors influencing restaurant popularity, the expanding realm of food delivery services, and the dynamic shifts in demographic landscapes. Elements such as population density, household income, age, and sex ratio are under scrutiny, using the case study of Arizona. This study's primary inquiry concerns whether Yelp-listed restaurants that provide food delivery services are more likely to prosper as population densities rise.

Data Cleaning

In [1]:
```
%%capture
!pip install tabulate
!pip install descartes
!pip install qeds fiona geopandas xgboost gensim folium pyLDAvis descartes
```

In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from tabulate import tabulate
import geopandas as gpd
from shapely.geometry import Point
%matplotlib inline
import qeds
```

```
import statsmodels.api as sm
from statsmodels.iolib.summary2 import summary_col
```

In [3]:
```
# Load the Yelp business dataset
base0 = pd.read_csv("yelp_business.csv")
base = pd.DataFrame(base0)
state_counts = base['state'].value_counts()
top_11_states = state_counts.index[:11]
base['state_category'] = base['state'].apply(lambda x: x if x in top_11_states els
```

In [4]:
```
# Data Cleaning: Filter restaurants with "Restaurant", and categorize by "Food deliv
restaurant = base[base['categories'].str.contains('Restaurant', case=False, na=Fal
delivery = restaurant[restaurant['categories'].str.contains('Delivery', case=False,
non_delivery = restaurant[~restaurant['categories'].str.contains('Delivery', case=F

restaurant_delivery = restaurant.copy()
restaurant_delivery['has_food_delivery'] = restaurant_delivery['categories'].str.cor

# Filter restaurants in Arizona (AZ)
AZ = restaurant_delivery[restaurant_delivery['state_category'] == 'AZ']
AZ_T = AZ.loc[restaurant_delivery['has_food_delivery'] == True]
AZ_F = AZ.loc[restaurant_delivery['has_food_delivery'] == False]

AZ_co = AZ.copy()
AZ_co["Coordinates"] = list(zip(AZ_co.longitude, AZ_co.latitude))
AZ_co["Coordinates"] = AZ_co["Coordinates"].apply(Point)
```

In pursuit of our research goal, I focus solely on establishments categorized as "Restaurant.' This refinement helps narrow our study scope.It involves categorizing restaurants based on their state, filtering for those that include ley words like "Restaurant", "food" in their categories. To delve into the influence of food delivery, a binary subcategory termed "Has food delivery" is generated based on the presence or absence of "Food Delivery" in the "categories" column. Additionally, I will import a Population dataset with the Zip code as the key from census data.

As mentioned before, I have gained the access to 13 variables from the Yelp business dataset: "Business id," "name," "neighbourhood," "address," "city," "state," "postal code," "latitude," "longitude," "stars," "Review count," "Is open," and "categories." In this case, The business id, their names, specifc address, and openins status are insignificance to our study because we need a comprehensive view of all businesses; "city," "state," "postal code," "latitude," and "longitude" are indicating the location of the business; as I will concentrate on US restaurants, I will keep "state". "postal code", "latitude", "longitude" will help me locate the restaurants in the map, and use this to merge with the population dataset in the future.

The crux of our investigation revolves around assessing the impact of food delivery services on restaurant success, measured by Yelp ratings ("stars"). Consequently, "stars" takes on the role of the dependent variable (Y variable), capturing the ratings assigned to each restaurant. I will first require the "categories"that forms the foundation of our independent variables (X variables). Notably, the presence of the "Has food delivery" subcategory serves as a critical factor in examining food delivery service effects. Despite a direct correlation not being immediately evident, the "Review count" variable retains its importance due to its potential relevance in gauging restaurant popularity. Hence, it remains a pivotal numerical variable in our analysis.

New merged datasets (For final project)

In [5]:
```
%%capture
# Import the state and Zip code dataset for the map
state_df = gpd.read_file("http://www2.census.gov/geo/tiger/GENZ2016/shp/cb_2016_us_s
zip_df = gpd.read_file("d1/Shapefile_az/tl_2019_us_zcta510.shp")
#Specify the atate to Arizona
arizona_state_df = state_df[state_df['STUSPS'] == 'AZ']
# ZIP codes in Arizona starting with "85" and "86"
arizona_zip_df = zip_df[zip_df['ZCTA5CE10'].str.startswith(("85", "86"))]
```

```
ERROR 1: PROJ: proj_create_from_database: Open of /opt/conda/share/proj failed
```

In [6]:
```
#Import the Population dataset with the Zip code as the Key
popu = pd.read_csv("Population.csv")
pop = pd.DataFrame(popu)
# Replace "NaN" with actual NaN values
pop['Total'] = pop['Total'].replace('NaN', pd.NA)
# Convert 'Total' column to numeric
pop['Total'] = pd.to_numeric(pop['Total'].str.replace(',', ''), errors='coerce')
pop['postal_code'] = pop['Label'].str.extract(r'ZCTA5 (\d+)')
pop_cleaned = pop.dropna()
pop_cleaned = pop_cleaned.reset_index(drop=True)
pop_cleaned = pop_cleaned.sort_values(by='Total', ascending=True)
```

In [7]:
```
# merge the zip data and the population data
zip_states_pop = arizona_zip_df.merge(pop_cleaned, left_on="ZCTA5CE10", right_on="p
zip_states_pop['population_density'] = zip_states_pop['Total'] / zip_states_pop['ALA
```

In [8]:
```
# merge the zip data and the stars data
merged_df = pd.merge(pop_cleaned, AZ_co, on='postal_code')
average_stars_by_postal_code = merged_df.groupby('postal_code')['stars'].mean() # th
average_star = pd.DataFrame(average_stars_by_postal_code)

merged_T = pd.merge(pop_cleaned, AZ_T, on='postal_code')
merged_F = pd.merge(pop_cleaned, AZ_F, on='postal_code')
average_stars_by_postal_codet = merged_T.groupby('postal_code')['stars'].mean()
average_star_T = pd.DataFrame(average_stars_by_postal_codet) # this is the average s
average_stars_by_postal_codef = merged_F.groupby('postal_code')['stars'].mean()
average_star_F = pd.DataFrame(average_stars_by_postal_codef) # this is the average s
```

In [9]:
```
zip_states_star_avg = arizona_zip_df.merge(average_star, left_on="ZCTA5CE10", right_
```

In [10]:
```
zip_states_star_F = zip_states_star_avg.merge(average_star_F, left_on="ZCTA5CE10", r
```

In [11]:
```
zip_states_star_total = zip_states_star_F.merge(average_star_T, left_on="ZCTA5CE10",
# stars_x: Total average stars
# stars_y: Average star for restaurant that no provide food delivery service
# stars: Average star for restaurant that provide food delivery service
```

In [12]:
```
zip_states_total = zip_states_star_total.merge(zip_states_pop, on="ZCTA5CE10", how='
selected_cols_total = ['ZCTA5CE10', 'GEOID10_x', 'CLASSFP10_x', 'MTFCC10_x', 'FUNCST
zip_states_total = zip_states_total[selected_cols_total]
zip_states_total.rename(columns={'GEOID10_x': 'GEOID10'}, inplace=True)
zip_states_total.rename(columns={'CLASSFP10_x': 'CLASSFP10'}, inplace=True)
zip_states_total.rename(columns={'MTFCC10_x': 'MTFCC10'}, inplace=True)
zip_states_total.rename(columns={'FUNCSTAT10_x': 'FUNCSTAT10'}, inplace=True)
zip_states_total.rename(columns={'INTPTLAT10_x': 'INTPTLAT10'}, inplace=True)
zip_states_total.rename(columns={'INTPTLON10_x': 'INTPTLON10'}, inplace=True)
zip_states_total.rename(columns={'geometry_x': 'geometry'}, inplace=True)
zip_states_total.rename(columns={'ALAND10_y': 'ALAND10'}, inplace=True)

zip_states_total.rename(columns={'stars_x': 'Total_stars'}, inplace=True)
zip_states_total.rename(columns={'stars_y': 'Non_FD_star'}, inplace=True)
zip_states_total.rename(columns={'stars': 'FD_star'}, inplace=True)
zip_states_total['population_density'] = zip_states_total['Total'] / zip_states_tota
# Population density is number of people/100 square meters
```

In [13]:
```
%%capture
# Import and clean the Income, Age dataset
Income = pd.read_csv("HouseholdIncome.csv")
selected_income = ['NAME', 'S1902_C03_001E' ]
Income = Income[selected_income]
Income.rename(columns={'NAME': 'postal_code'}, inplace=True)
Income['postal_code'] = Income['postal_code'].str.extract(r'ZCTA5 (\d+)')
Income.rename(columns={'S1902_C03_001E': 'Mean_Income'}, inplace=True) # Estimate!!
Income = pd.DataFrame(Income)
Income = Income.drop(0)
Income = Income.reset_index(drop=True)
Income['Mean_Income'] = pd.to_numeric(Income['Mean_Income'], errors='coerce')
Income['Mean_Income'] = np.log(Income['Mean_Income'])
zip_states_Income = arizona_zip_df.merge(Income, left_on="ZCTA5CE10", right_on="post

Age = pd.read_csv("Ages.csv")
selected_Age = ['NAME', 'DP05_0001E', 'DP05_0004E','DP05_0008E','DP05_0009E']
Age = Age[selected_Age]
```

```
Age.rename(columns={'NAME': 'postal_code'}, inplace=True)
Age['postal_code'] = Age['postal_code'].str.extract(r'ZCTA5 (\d+)')
Age.rename(columns={'DP05_0001E': 'Total_pop'}, inplace=True)
Age.rename(columns={'DP05_0004E': 'Sex_ratio'}, inplace=True)
Age.rename(columns={'DP05_0008E': 'Pop_15_19'}, inplace=True)
Age.rename(columns={'DP05_0009E': 'Pop_20_24'}, inplace=True)
Age = pd.DataFrame(Age)
Age = Age.drop(0)
Age = Age.reset_index(drop=True)
Age["Pop_15_19"] = Age["Pop_15_19"].astype(float)
Age["Pop_20_24"] = Age["Pop_20_24"].astype(float)
Age["Total_pop"] = Age["Total_pop"].astype(float)
Age["Pop_15_24"] = Age["Pop_15_19"] + Age["Pop_20_24"]
Age["Pop_15_24"] = ((Age["Pop_15_24"] / Age["Total_pop"]) * 100).round(2)

# Merge the Income, Age(Percentage of young people from 15-24), Sex (Number of males
Final_total = zip_states_total.merge(Income, left_on="ZCTA5CE10", right_on="postal_c
Final_total = Final_total.merge(Age, left_on="ZCTA5CE10", right_on="postal_code", ho

# Clean the merged dataset and select the variables for the regression
Final_regression = Final_total[['ZCTA5CE10',  'ALAND10', 'Total', 'population_densit
Final_regression['Sex_ratio'] = pd.to_numeric(Final_regression['Sex_ratio'], errors=

# Add new variable which is how many restaurants in total in each zip code
merged_df = pd.merge(pop_cleaned, AZ_co, on='postal_code')
grouped_df = merged_df.groupby('postal_code')
postal_code_counts = grouped_df['postal_code'].value_counts()
Number_res = pd.DataFrame(postal_code_counts)
zip_states_count = arizona_zip_df.merge(Number_res, left_on="ZCTA5CE10", right_on="p

Final_regression = Final_regression.merge(Number_res, left_on="ZCTA5CE10", right_on=
Final_regression_merged = Final_regression.merge(AZ_co, left_on="ZCTA5CE10", right_c
Final_regression_merged = Final_regression_merged.dropna(subset=['ZCTA5CE10']) # Cle
Final_regression_merged['has_food_delivery'] = Final_regression_merged['has_food_del
# New variable: the ratio of food delivery service in each zip code
percentage_delivery = Final_regression_merged.groupby('ZCTA5CE10').apply(
    lambda group: (group['has_food_delivery'] == 1).sum() / group['count'] * 100
).reset_index(name='percentage_delivery')
zip_states_nfd = arizona_zip_df.merge(percentage_delivery, left_on="ZCTA5CE10", righ
# Merge the calculated percentages back to the original DataFrame
Final_regression_merged = Final_regression_merged.merge(percentage_delivery, on='ZCT
print(zip_states_nfd)
```

By integrating population data for each zip code with detailed map shapefile information from the United States Census Bureau, I've obtained a comprehensive understanding of geographic locations (longitude and latitude ranges) and land areas. This data synergy has enabled me to effortlessly compute population density by dividing the population figure by the land area, introducing a crucial variable named "population density." This newly introduced metric has proven to be immensely insightful, unveiling the concentration of residents across distinct zip code regions. Consequently, it serves as a pivotal indicator for evaluating potential customer bases and gauging the demand for food services within these areas.

Moreover, my data aggregation efforts extend to importing the Income and Age dataset from the United States Census Bureau. This dataset encompasses two vital dimensions: mean household income and age demographics. The former provides a nuanced understanding of the economic landscape, allowing for the identification of areas with greater financial capacity to engage in food-related expenditures, including dining out and

catering services. Meanwhile, the age demographics component delves into the composition of each zip code's population. This encompasses the sex ratio, representing the ratio of males to females per 100 individuals, as well as the percentage of young people (aged range from 15 to 24) within the overall population for each zip code in Arizona.

Summary Statstics Tables

In this section, I show the summary statistics tables for the 'stars' and 'review_count' columns in the whole dataset, as well as creating formatted tabular summaries for the ratings and review counts based on different states. After merging the new datasets from the United States Census Bureau, I demonstrate the summary statistics tables for though new variables as the 'population', the land area of each zip code,

In [14]:
```
summary_stats = base[['stars', 'review_count']].describe()
print(summary_stats)
```

```
              stars   review_count
count  174567.000000  174567.000000
mean        3.632196      30.137059
std         1.003739      98.208174
min         1.000000       3.000000
25%         3.000000       4.000000
50%         3.500000       8.000000
75%         4.500000      23.000000
max         5.000000    7361.000000
```

The 'Stars' column shows that the average star rating for restaurants in the dataset is around 3.63, with a median of 3.50. The minimum star is 1, and the maximum is 5.

The 'Review count' column reveals that the average number of reviews for restaurants is about 30.14, with a median of 8.00. The minimum review count is 3, and the maximum is as high as 7361.

In [15]:
```
# Summary table for stars
stars_summary = base.groupby('state_category').agg({
    'stars': ['mean', 'median', 'min', 'max', lambda x: np.percentile(x, 25), lambd
    'categories': 'count'
}).reset_index()
stars_summary.columns = ['State', 'Mean', 'Median', 'Min', 'Max',  '25%', '75%', 'Co
stars_summary['Mean'] = stars_summary['Mean'].round(2)

# Summary table for review counts
review_summary = base.groupby('state_category').agg({
    'review_count': ['mean', 'median', 'min', 'max', lambda x: np.percentile(x, 25)
    'categories': 'count'
}).reset_index()
review_summary.columns = ['State', 'Mean', 'Median', 'Min', 'Max', '25%', '75%', 'Co
review_summary['Mean'] = review_summary['Mean'].round(2)

tabular_stars_summary = tabulate(stars_summary, headers='keys', tablefmt='pretty')
tabular_review_summary = tabulate(review_summary, headers='keys', tablefmt='pretty')
print("Stars Summary:")
print(tabular_stars_summary)
print("\nReview Summary:")
print(tabular_review_summary)
```

Stars Summary:

	State	Mean	Median	Min	Max	25%	75%	Count
0	AZ	3.73	4.0	1.0	5.0	3.0	4.5	52214
1	BW	3.81	4.0	1.0	5.0	3.5	4.5	3118
2	EDH	3.78	4.0	1.0	5.0	3.5	4.5	3795
3	IL	3.49	3.5	1.0	5.0	3.0	4.5	1852
4	NC	3.57	3.5	1.0	5.0	3.0	4.5	12956
5	NFI	3.65	3.5	1.0	5.0	3.0	4.5	1697
6	NV	3.71	4.0	1.0	5.0	3.0	4.5	33086
7	OH	3.54	3.5	1.0	5.0	3.0	4.5	12609
8	ON	3.41	3.5	1.0	5.0	3.0	4.0	30208
9	PA	3.61	3.5	1.0	5.0	3.0	4.5	10109
10	QC	3.66	4.0	1.0	5.0	3.0	4.5	8169
11	WI	3.64	3.5	1.0	5.0	3.0	4.5	4754

Review Summary:

	State	Mean	Median	Min	Max	25%	75%	Count
0	AZ	31.17	9.0	3	2215	4.0	25.0	52214
1	BW	11.35	6.0	3	160	4.0	12.0	3118
2	EDH	12.62	6.0	3	379	4.0	13.0	3795
3	IL	19.64	8.0	3	744	4.0	18.25	1852
4	NC	23.73	8.0	3	1586	4.0	20.0	12956
5	NFI	10.55	5.0	3	557	3.0	9.0	1697
6	NV	55.14	12.0	3	7361	5.0	37.0	33086
7	OH	19.33	7.0	3	927	4.0	17.0	12609
8	ON	21.0	7.0	3	1494	4.0	19.0	30208
9	PA	22.73	7.0	3	1476	4.0	19.0	10109
10	QC	17.92	7.0	3	1953	4.0	16.0	8169
11	WI	23.08	8.0	3	1449	4.0	20.0	4754

These tables show the summary statistics of Stars/Review count for each state.

1. The 'Mean' column shows the average Stars/Review count for each state.
2. The 'Median' column shows the median Stars/Review count for each state.
3. The 'Min' and 'Max' columns indicate the minimum and maximum Stars/Review count.
4. The '25%' and '75%' columns represent the 25th and 75th percentiles, respectively, offering insights into the distribution of Stars/Review count within each state.
5. The 'Count' column shows how many restaurants are in each state.

In [16]:
```
delivery_counts = restaurant_delivery.groupby('has_food_delivery').size().reset_inde
# Grouping by 'state' and counting the occurrences
state_counts = restaurant_delivery.groupby('state_category').size().reset_index(name
print("Restaurant Delivery Counts:")
delivery_table = tabulate(delivery_counts, headers=['has_food_delivery', 'Count'], t
print(delivery_table)
print("\nRestaurant Counts by State:")
state_table = tabulate(state_counts, headers=['State', 'Count'], tablefmt='pretty')
print(state_table)
```

Restaurant Delivery Counts:

	has_food_delivery	Count
0	False	17964
1	True	1078

Restaurant Counts by State:

	State	Count
0	AZ	4194
1	BW	535
2	EDH	553
3	IL	213
4	NC	1512
5	NFI	220
6	NV	2941
7	OH	1689
8	ON	4278
9	PA	1092
10	QC	1254
11	WI	561

The first table shows the count of restaurants based on whether they offer food delivery (True) or not (False). There are 1,078 restaurants that offer food delivery (True). There are 17,964 restaurants that do not offer food delivery (False).

The second table presents the counts of restaurants in each state category. For example, in Arizona (AZ), there are 4,194 restaurants. The table provides a breakdown of restaurant counts for each state category, with the help of this table and histogram below I could select a representitive state for my study.

I could therefore learn from the "Restaurant Delivery Counts" table that the majority of restaurants (17964) do not offer food delivery, while a smaller percentage (1078) do. It means most of the Yelp restaurants have not started providing food delivery services yet, this could be an opportunity. The "Restaurant Counts by State" table displays the distribution of restaurants across different state categories. For instance, Ontario (ON) has the highest restaurant count with 4278 restaurants, followed by Arizona (AZ) with 4194 restaurants. The tables provide insights into the prevalence of food delivery services and the distribution of restaurants among various state categories. This information can be useful for understanding the restaurant landscape in terms of delivery services and geographical distribution at the state level.

New merged datasets (For final project)

In [17]:
```
# Land area, population, Population density, Age
summary_stats_new1 = Final_regression[['ALAND10','Total','population_density', 'Pop_
summary_stats1 = summary_stats_new1.iloc[1:10, :]
summary_stats1 = summary_stats1.round(2)
headers = summary_stats1.columns.tolist()
stat_labels = ["mean", "std", "min", "25%", "50%", "75%", "max"]
summary_stats1.insert(0, "Statistic", stat_labels)
table_data1 = [summary_stats1.columns.tolist()] + summary_stats1.values.tolist()
```

```
formatted_table1 = tabulate(table_data1, headers='firstrow', tablefmt='pretty', numa
print(formatted_table1)
```

```
+------------+--------------+----------+----------------------+----------+
| Statistic  |   ALAND10    |  Total   | population_density   | Pop_15_24 |
+------------+--------------+----------+----------------------+----------+
|    mean    | 75532792.93  | 36270.77 |         0.13         |   12.84   |
|    std     | 160984328.38 | 16896.58 |         0.09         |   5.04    |
|    min     |  4115789.0   |  523.0   |         0.0          |   0.82    |
|    25%     | 21384019.25  | 25752.5  |         0.07         |   10.27   |
|    50%     |  27996834.0  | 37953.0  |         0.13         |   12.85   |
|    75%     | 44340187.75  | 46777.25 |         0.19         |   14.95   |
|    max     | 1010525491.0 | 73551.0  |         0.37         |   45.46   |
+------------+--------------+----------+----------------------+----------+
```

In [18]:
```
# Income, Sex, Education
summary_stats_new2 = Final_regression[['Mean_Income','Sex_ratio','Total_stars', 'cou
summary_stats2 = summary_stats_new2.iloc[1:10, :]
summary_stats2 = summary_stats2.round(2)
summary_stats2['count'] = summary_stats2['count'].round(0)
headers = summary_stats1.columns.tolist()
stat_labels = ["mean", "std", "min", "25%", "50%", "75%", "max"]
summary_stats2.insert(0, "Statistic", stat_labels)
table_data2 = [summary_stats2.columns.tolist()] + summary_stats2.values.tolist()
formatted_table2 = tabulate(table_data2, headers='firstrow', tablefmt='pretty', numa
print(formatted_table2)
```

```
+-----------+--------------+-----------+--------------+-------+
| Statistic | Mean_Income  | Sex_ratio | Total_stars  | count |
+-----------+--------------+-----------+--------------+-------+
|   mean    |    11.25     |   98.25   |     3.31     | 35.0  |
|    std    |     0.39     |    9.21   |     0.42     | 28.0  |
|    min    |    10.43     |    70.4   |      1.5     |  1.0  |
|    25%    |    11.02     |   93.52   |      3.1     | 16.0  |
|    50%    |     11.2     |   97.45   |     3.33     | 30.0  |
|    75%    |    11.55     |   101.1   |     3.54     | 46.0  |
|    max    |    12.45     |   139.6   |      5.0     | 175.0 |
+-----------+--------------+-----------+--------------+-------+
```

1. **ALAND10 (Land Area)**:
   - The mean land area across the zip codes in Arizona is approximately 75,532,792.93 square meters.
   - Land areas vary significantly, as indicated by the standard deviation of around 160,984,328.38.
   - The smallest land area recorded is about 4,115,789 square meters.
   - The 25th percentile suggests that 25% of zip codes have land areas below 21,384,019.25 square meters.
   - The median (50th percentile) land area is around 27,996,834 square meters, indicating the middle value.
   - The 75th percentile indicates that 75% of zip codes have land areas below 44,340,187.75 square meters.
   - The largest land area recorded is approximately 1,010,525,491 square metersunits.

1. **Total (Total Population)**:
   - The mean total population of the zip codes is approximately 36,270.77.
   - Population counts vary significantly, with a standard deviation of around 16,896.58.
   - The smallest population count is 523 individuals.

- The 25th percentile suggests that 25% of zip codes have populations below 25,752.5.
- The median population count is around 37,953, representing the middle value.
- The 75th percentile indicates that 75% of zip codes have populations below 46,777.25.
- The largest population count is 73,551 individuals.

1. **Population Density**:
   - The mean population density is approximately 0.13 individuals per unit area.
   - Population density values have relatively low variability, with a standard deviation of about 0.09.
   - The minimum recorded population density is very low, indicating sparsely populated areas.
   - The 25th percentile suggests that 25% of zip codes have population densities below 0.07 individuals per unit area.
   - The median population density is around 0.13 individuals per unit area.
   - The 75th percentile indicates that 75% of zip codes have population densities below 0.19 individuals per unit area.
   - The highest population density recorded is 0.37 individuals per unit area.

1. **Pop_15_24 (Percentage of Population Age 15-24)**:
   - The mean percentage of the population aged 15 to 24 is approximately 12.84%.
   - Percentage values have relatively low variability, with a standard deviation of about 5.04%.
   - The lowest recorded percentage is 0.82%, indicating areas with a small proportion of young adults.
   - The 25th percentile suggests that 25% of zip codes have populations aged 15-24 below 10.27%.
   - The median percentage is around 12.85%, representing the middle value.
   - The 75th percentile indicates that 75% of zip codes have populations aged 15-24 below 14.95%.
   - The highest recorded percentage is 45.46%, indicating areas with a larger concentration of young adults.

1. **Total_stars (Average Stars in each zip code)**:
   - The mean of it is approximately 3.31.
   - The number of average stars shows relatively low variability, with a standard deviation of about 0.42.
   - The lowest recorded value is 1.5, indicating some areas with very low average stars.
   - The 25th percentile suggests that 25% of zip codes have average stars below 3.1.
   - The median number of average stars is around 3.33, representing the middle value.
   - The 75th percentile indicates that 75% of zip codes have total stars below 3.54.
   - The highest recorded value is 5, indicating areas of restaurants with a very high average stars.

1. **Mean_Income (The log of Mean Income)**:

   - The log of mean income across the zip codes is approximately 11.25.

- Income levels vary, as shown by the standard deviation of around 0.39.
- The lowest recorded log of mean income is about 10.43.
- The 25th percentile suggests that 25% of zip codes have the log of mean incomes below 11.02.
- The median of log of mean income is around 11.2, representing the middle value.
- The 75th percentile indicates that 75% of zip codes have the log of mean incomes below 11.55.
- The highest recorded the log of mean income is approximately 12.45 dollars.

2. **Sex_ratio (Sex Ratio):**

- The mean sex ratio (female to male ratio) across the zip codes is approximately 98.25.
- Sex ratios vary slightly, as indicated by the standard deviation of around 9.21.
- The lowest recorded sex ratio is about 70.4, indicating an area with fewer females.
- The 25th percentile suggests that 25% of zip codes have sex ratios below 93.525.
- The median sex ratio is around 97.45, representing a balanced ratio.
- The 75th percentile indicates that 75% of zip codes have sex ratios below 101.1.
- The highest recorded sex ratio is approximately 139.6.

3. **Count (Total number of Yelp restuarants in each zip code):**

- The mean count value in the dataset is approximately 35.16.
- Counts vary considerably, as indicated by a standard deviation of around 28.15.
- The smallest count value is 1, suggesting that there are zip codes with very low counts.
- The 25th percentile indicates that 25% of the zip codes have counts below 16.
- The median count value is around 30.5, representing the middle value.
- The 75th percentile suggests that 75% of the zip codes have counts below 46.5.
- The largest count recorded is 175, indicating areas with a high count.

These insights provide an overview of the distribution, central tendencies, and variability of each variable within the dataset. It's important to consider these statistics in context to have a deeper understanding of the characteristics of the areas represented by the zip codes in Arizona.

Plots

In this module, A histogram will be made to show how ratings are distributed across the platforms of all the businesses, allowing for a better understanding of the star distribution. Second, a barplot will be created to show the restaurant's performance by the states listed in the summary statistics table. The construction of custom categories will also be centred on the restaurant, specifically the distribution of ratings for eateries that offer or do not offer food delivery services.

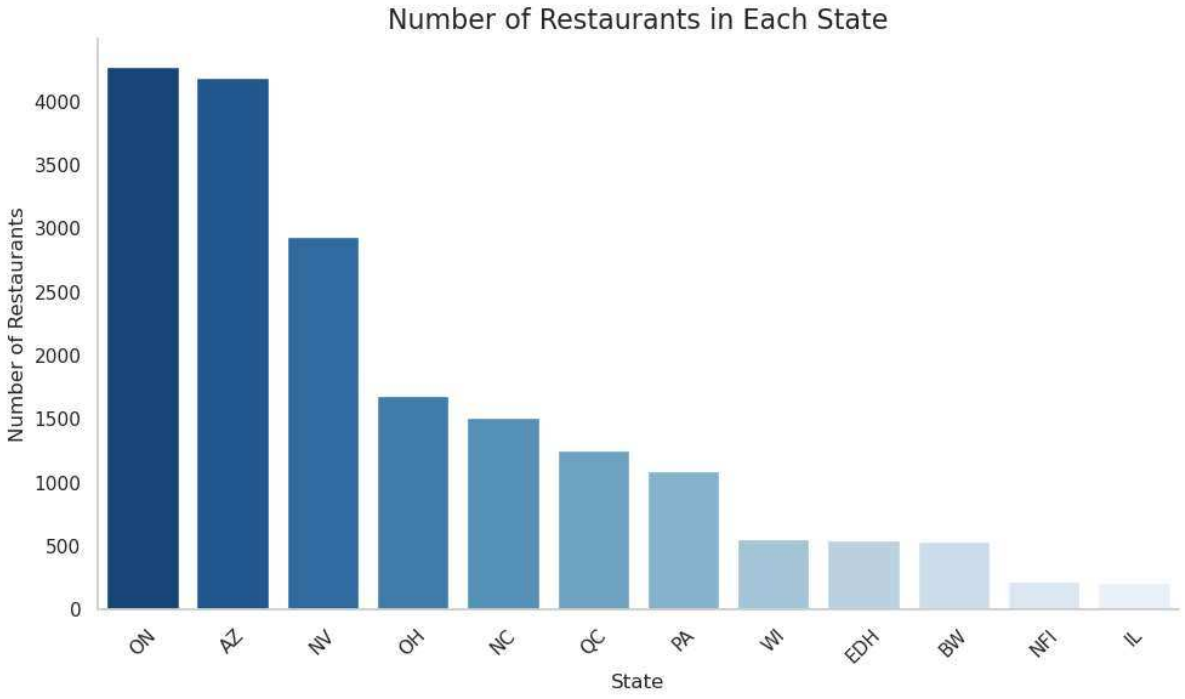
In [34]:
```
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)
x = base['stars'].value_counts().sort_index()
plt.figure(figsize=(10, 6))
color_palette = sns.color_palette("Blues", len(x))
ax = sns.barplot(x=x.index, y=x.values, alpha=0.8, palette=color_palette)
```

```
sns.set_style("whitegrid", {"axes.facecolor": "white"})
ax.yaxis.grid(False)
plt.title("Star Rating Distribution", fontsize=16)
plt.ylabel('Number/Percentage of businesses', fontsize=12)
plt.xlabel('Star Ratings', fontsize=12)
rects = ax.patches
count_labels = x.values
for rect, label in zip(rects, count_labels):
    height = rect.get_height()
    ax.text(
        rect.get_x() + rect.get_width() / 2,
        height + 5,
        label,
        ha='center',
        va='bottom'
    )
p = base['stars'].value_counts(normalize=True).sort_index() * 100
rects = ax.patches
percentage_labels = p.values
for rect, label in zip(rects, percentage_labels):
    height = rect.get_height()
    ax.text(
        rect.get_x() + rect.get_width() / 2,
        height - 15,
        f'{label:.2f}%',
        ha='center',
        va='top'
    )
sns.despine(top=True)
plt.show()
```

This histogram provides an overview of the distribution of star ratings across all businesses in the dataset. The x-axis represents the star ratings, and the y-axis represents the number of businesses with each rating. The result shows that the majority of businesses have ratings above 3 stars (around 80%) and centred around 3.5 to 4 stars (37.54%). This graph explains how star ratings are distributed, which can be used as a performance indicator for

restaurants. Although it doesn't specifically address the aspect of food delivery and population, it gives a broad overview for my study.

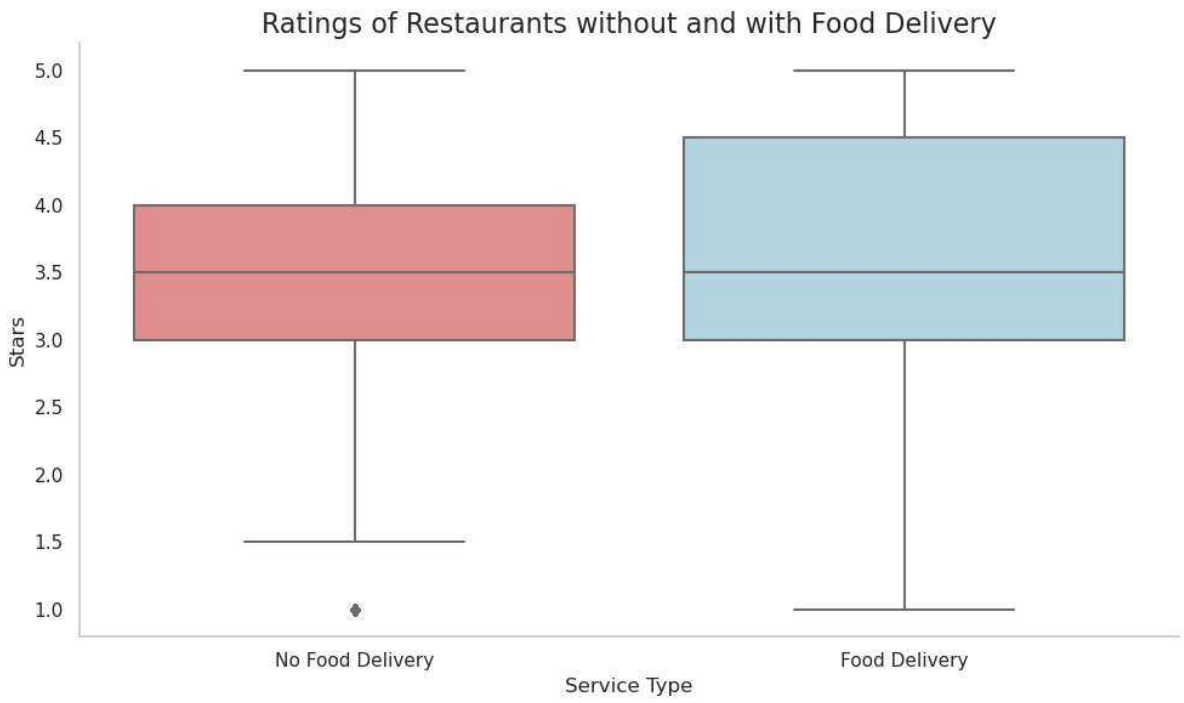
In [35]:
```
warnings.simplefilter(action='ignore', category=FutureWarning)
colors = sns.color_palette('Blues_r', len(restaurant['state_category'].value_counts(
plt.figure(figsize=(10, 6))
sns.countplot(data=restaurant, x='state_category', order=restaurant['state_category'
plt.title('Number of Restaurants in Each State', fontsize=16)
plt.xlabel('State')
plt.ylabel('Number of Restaurants')
plt.xticks(rotation=45)
sns.despine(top=True)
plt.grid(False)
plt.tight_layout()
plt.show()
```



This bar plot illustrates the number of restaurants in each state category. Each bar represents a state, and its height indicates the number of restaurants located in that state. This plot helps me identify which states have the highest concentration of restaurants, providing a potential representative state to focus on for further analysis. Since I want to find a representative state of Yelp restaurant in the US. I will choose Arizona (AZ) because it has the second-largest amount of Yelp restaurants in our dataset following the Ontario (ON) from Canada.

In [36]:
```
warnings.simplefilter(action='ignore', category=FutureWarning)
# The side-by-side boxplot for the ratings of restaurants that provide food delivery
plt.figure(figsize=(10, 6))
sns.set(style="whitegrid")
colors = {True: 'lightblue', False: 'lightcoral'}
boxplot = sns.boxplot(data=restaurant_delivery, x='has_food_delivery', y='stars', pa
boxplot.set_xticklabels(['No Food Delivery', 'Food Delivery'])

plt.title('Ratings of Restaurants without and with Food Delivery', fontsize=16)
plt.xlabel('Service Type')
plt.ylabel('Stars')
sns.despine(top=True)
plt.grid(False)
```

```
plt.tight_layout()
plt.show()
```

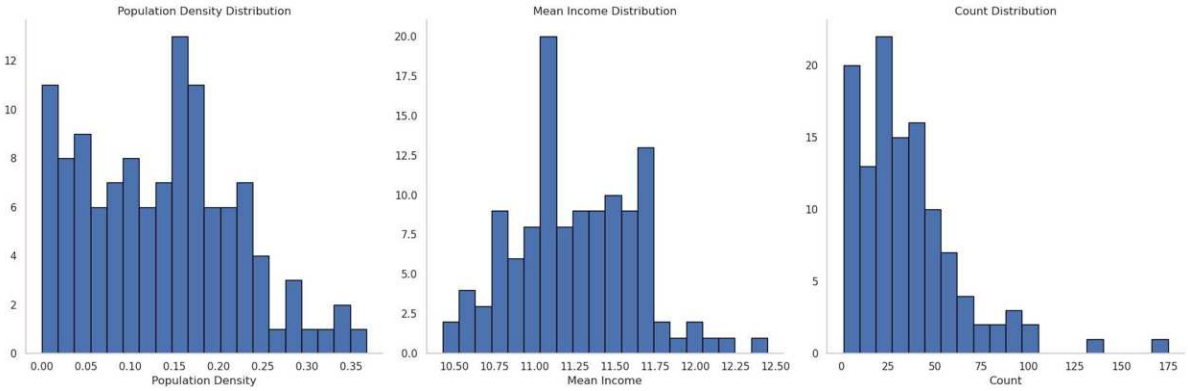


This side-by-side boxplot compares the distribution of star ratings for restaurants that provide food delivery (True) and those that do not (False). Each box represents the interquartile range (IQR) of ratings for each group, with the median marked by a horizontal line in between. This plot directly addresses the impact of food delivery services on restaurant ratings, which is a key aspect of my research question. The medians of both groups are the same (3.5 stars), but there are differences in the quartiles. Restaurants with food delivery have a higher upper quartile (75th percentile) compared to those without, suggesting that restaurants with food delivery may have slightly higher ratings on the upper end.

New merged datasets (For final project)

In [37]:
```
population_density = Final_regression['population_density']
mean_income = Final_regression['Mean_Income']
count = Final_regression['count']

fig, axes = plt.subplots(nrows=1, ncols=3, figsize=(18, 6))

# Population Density Histogram
axes[0].hist(population_density, bins=20, edgecolor='black')
axes[0].set_title('Population Density Distribution')
axes[0].set_xlabel('Population Density')
axes[0].set_ylabel('')
axes[0].grid(False)

# Mean Income Histogram
axes[1].hist(mean_income, bins=20, edgecolor='black')
axes[1].set_title('Mean Income Distribution')
axes[1].set_xlabel('Mean Income')
axes[1].set_ylabel('')
axes[1].grid(False)

# Count Histogram
axes[2].hist(count, bins=20, edgecolor='black')
```

```
axes[2].set_title('Count Distribution')
axes[2].set_xlabel('Count')
axes[2].set_ylabel('')
axes[2].grid(False)

sns.despine(top=True)
plt.tight_layout()
plt.show()
```



Now I am demonstrating three important variables that are listed in the summary statistics table, and the visualization would be a better support to learn more about the distribution of these data:

1. **Population Density Distribution:**
    - This histogram shows the distribution of population density across different zip code areas.
    - The x-axis represents population density values, while the y-axis represents the frequency (number of geographic areas) falling within each population density range.

1. **Mean Income Distribution:**
    - This histogram visualizes the distribution of mean income across different zip code areas.
    - The x-axis represents mean income values, and the y-axis represents the frequency of geographic areas in each income range.

1. **Count Distribution:**
    - This histogram illustrates the distribution of Numbers of restaurants across different zip code areas.
    - The x-axis represents the "count" values, and the y-axis represents the frequency of geographic areas with each "count" value.

Project Two

The Message

The central focus of this investigation revolves around establishing a potential correlation between the success of restaurants listed on Yelp (as indicated by their star ratings) and the prevalence of food delivery services, particularly in the context of rising population densities.

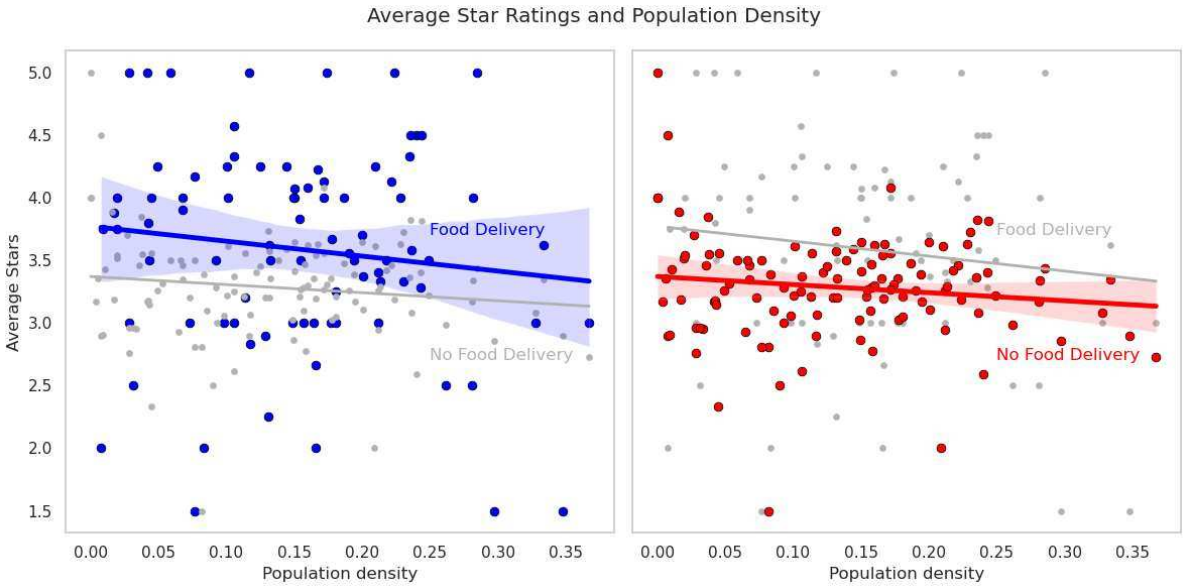
Drawing inspiration from a study on Chinese urbanization, which mentions that heightened population density leads to intensified competition and consequently, the survival of only the finest dining establishments offering exceptional fare and services. This notion suggests that in areas characterized by the presence of high-quality restaurants, the average star ratings should naturally be elevated. This idea is supported by Matti's (2020) study about Competition and Consumer Reviews in Phoenix that clustering is an explanation for why more nearby competitors are associated with higher ratings. To visually illustrate this concept, I intend to employ scatterplots accompanied by regression lines.

While definitive conclusions remain elusive at this juncture, my visualization aims to take you on an insightful expedition, peeling back layers to reveal potential revelations. It will delve deep into the intricate interplay between population density and mean restaurant ratings, dissected by the availability of food delivery services within the confines of Arizona. The plan is to build a split scatter plot with two panels

The left panel will elucidate the correlation between the average star ratings of restaurants offering food delivery services and the density of the population, depicted through the use of blue markers and regression lines. In contrast, the right panel will mirror the same relationship for restaurants that do not provide delivery services, using red markers and corresponding regression lines. Ultimately, my goal is to offer an intuitive exploration of the potential answers to this important question through graphs.

In [38]:
```
sns.set(style="whitegrid")
# Create the scatter plot
plt.figure(figsize=(12, 6))
plt.suptitle('Average Star Ratings and Population Density')

plt.subplot(1, 2, 1)
sns.scatterplot(x='population_density', y='FD_star', data=zip_states_total, color='b
sns.scatterplot(x='population_density', y='Non_FD_star', data=zip_states_total, colo
sns.regplot(x='population_density', y='FD_star', data=zip_states_total, color='blue'
sns.regplot(x='population_density', y='Non_FD_star', data=zip_states_total, color=((

plt.grid(False)
plt.xlabel('Population density')
plt.ylabel('Average Stars')
plt.tick_params(axis='x', which='both', bottom=False, top=False)
plt.title('')
plt.text(0.25, 3.7, 'Food Delivery', color='blue')
plt.text(0.25, 2.7, 'No Food Delivery', color=(0.7, 0.7, 0.7))

plt.subplot(1, 2, 2)
sns.scatterplot(x='population_density', y='FD_star', data=zip_states_total, color=((
sns.scatterplot(x='population_density', y='Non_FD_star', data=zip_states_total, colo
sns.regplot(x='population_density', y='Non_FD_star', data=zip_states_total, color='r
sns.regplot(x='population_density', y='FD_star', data=zip_states_total, color=(0.7,

plt.grid(False)
plt.xlabel('Population density')
plt.ylabel('')
plt.gca().axes.get_yaxis().set_visible(False)
plt.tick_params(axis='x', bottom=False)
plt.title('')
plt.tight_layout()
plt.text(0.25, 2.7, 'No Food Delivery', color='red')
plt.text(0.25, 3.7, 'Food Delivery', color=(0.7, 0.7, 0.7))
```

```
plt.show()
```



1. **Diverse Rating Trends with Population Density:**
    - One striking observation from the visualization is the decrease in average star ratings as population density rises. This phenomenon is evident for both food delivery and non-food delivery Yelp restaurants in Arizona.
2. **Similar Slopes, Contrasting Steadiness:**
    - While the slopes of the regression lines are quite similar for both categories, indicating a shared sensitivity to population density, there are still differences exist in the steadiness of the trends.
    - Restaurants that do not provide food delivery services showcase a flatter regression line, implying that their average ratings respond more consistently to changes in population density.
    - On the other hand, restaurants with food delivery exhibit a steeper regression line, suggesting that the presence of delivery services amplifies the impact of population density on ratings.
3. **Potential Delivery Advantage:**
    - A noteworthy insight surfaces in the form of an elevation in average ratings for restaurants that offer food delivery services. This distinction is approximately 0.2 stars in average higher than their non-delivery counterparts.
4. **Confidence Intervals and Spread:**
    - The variability in data distribution becomes apparent through the spread of dots in the scatter plots. Restaurants providing delivery services display a wider spread compared to their non-delivery counterparts.
    - Consequently, the wider confidence interval associated with delivery restaurants' regression line implies a greater level of uncertainty in their rating response to population density changes.

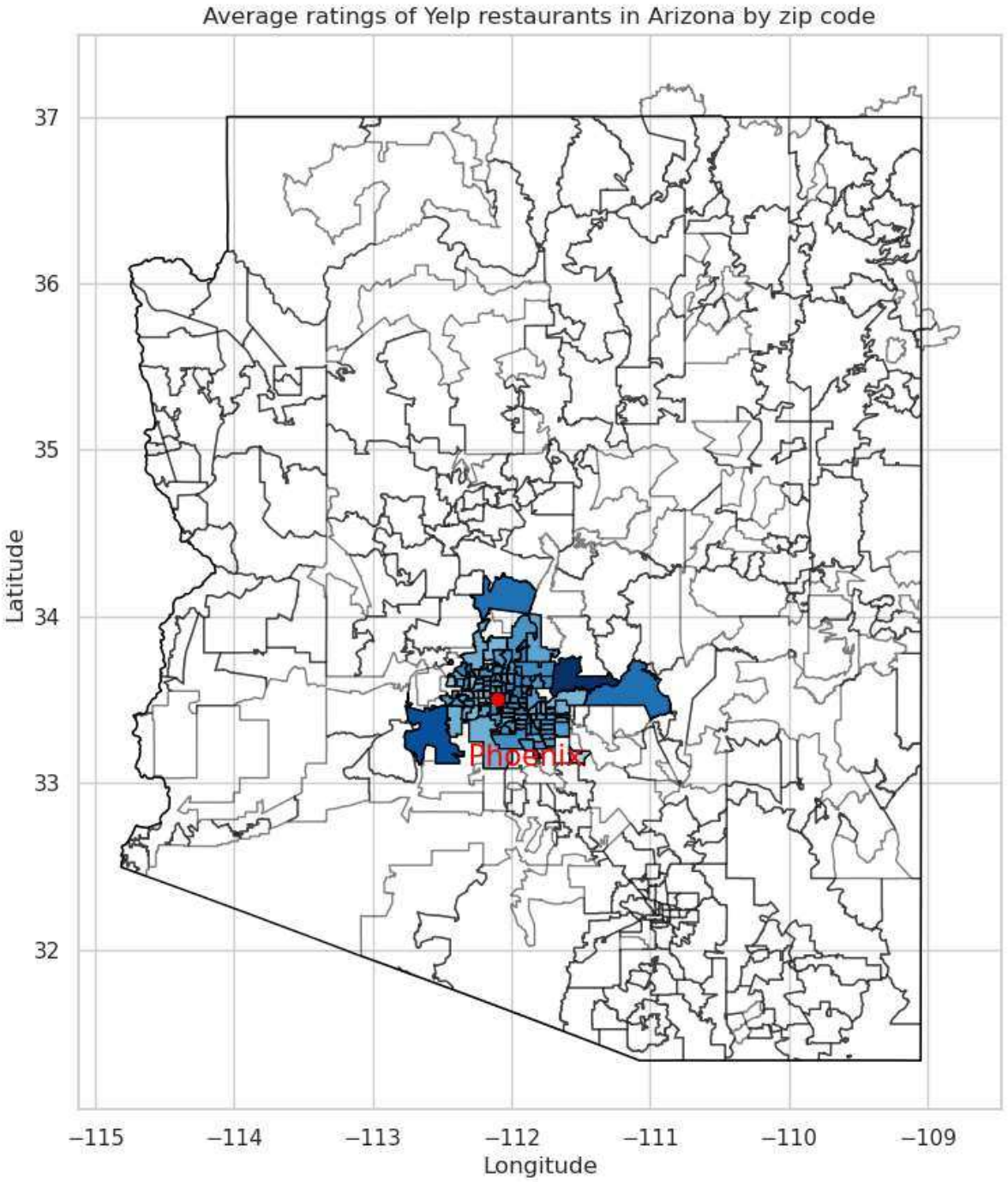
In summary, the visualization sheds light on the relationship between population density, average restaurant ratings, and the presence of food delivery services. Notably, as

population density increases, average ratings for both restaurant categories tend to decline, underscoring the competitive nature of densely populated areas. However, the introduction of food delivery services introduces nuanced dynamics that warrant more in-depth examination. To draw definitive conclusions, additional research involving regression and machine learning models is imperative. This ongoing work is essential to provide a comprehensive understanding of the factors at play in this complex relationship.

Maps and Interpretations

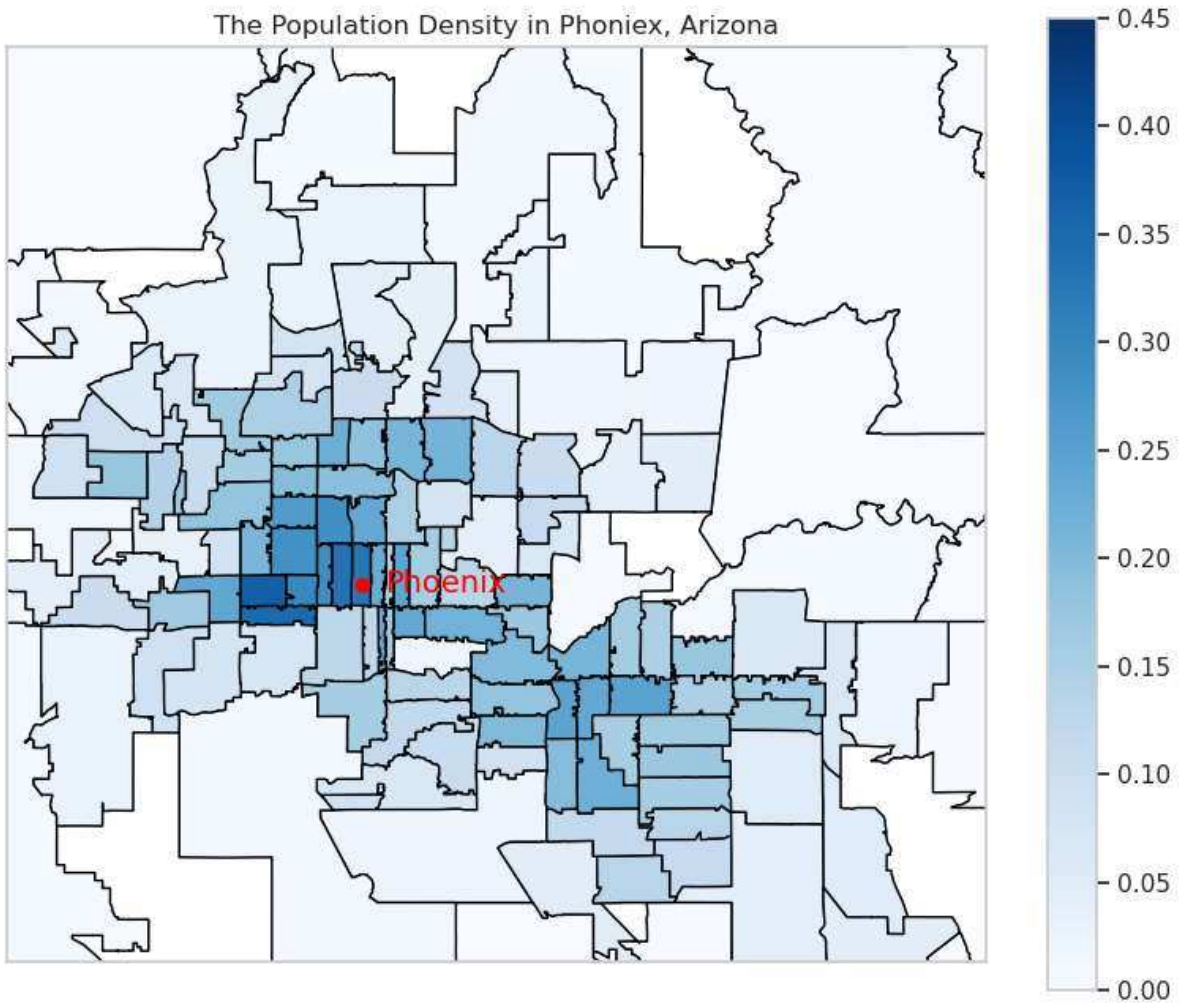
During my study focused on Arizona, I conducted an analysis of Yelp restaurants based on their locations within the state. After plotting the restaurants into the state map, I found that the majority of these restaurants were clustered around the Phoenix area. To delve deeper into this trend, I've decided to narrow my investigation by limiting the latitude and longitude ranges. This refined approach will enable me to conduct a focused case study specifically centered on Phoenix. By filtering the data to encompass a precise geographical range, I aim to gain a more comprehensive understanding of the restaurant landscape in this prominent city. This strategic adjustment will not only allow for a more detailed analysis but also provide insights into the unique culinary scene that Phoenix has to offer.

In [39]:
```
fig, ax = plt.subplots(figsize=(10, 10))
arizona_state_df.plot(ax=ax, edgecolor="black", color="none")
arizona_zip_df.plot(ax=ax, edgecolor="black", color="none", alpha=0.5)

zip_states_star_avg.plot(
    ax=ax, edgecolor='black', column='stars', legend=False, cmap='Blues',
    vmin=1, vmax=5
)

phoenix_coords = (33.5, -112.1)
# Plot a red point for Phoenix and add label
ax.scatter(phoenix_coords[1], phoenix_coords[0], color='red')
ax.text(phoenix_coords[1] - 0.21, phoenix_coords[0]-0.4, 'Phoenix', fontsize=15, col
plt.title("Average ratings of Yelp restaurants in Arizona by zip code")
plt.xlabel("Longitude")
plt.ylabel("Latitude")
plt.show()
```

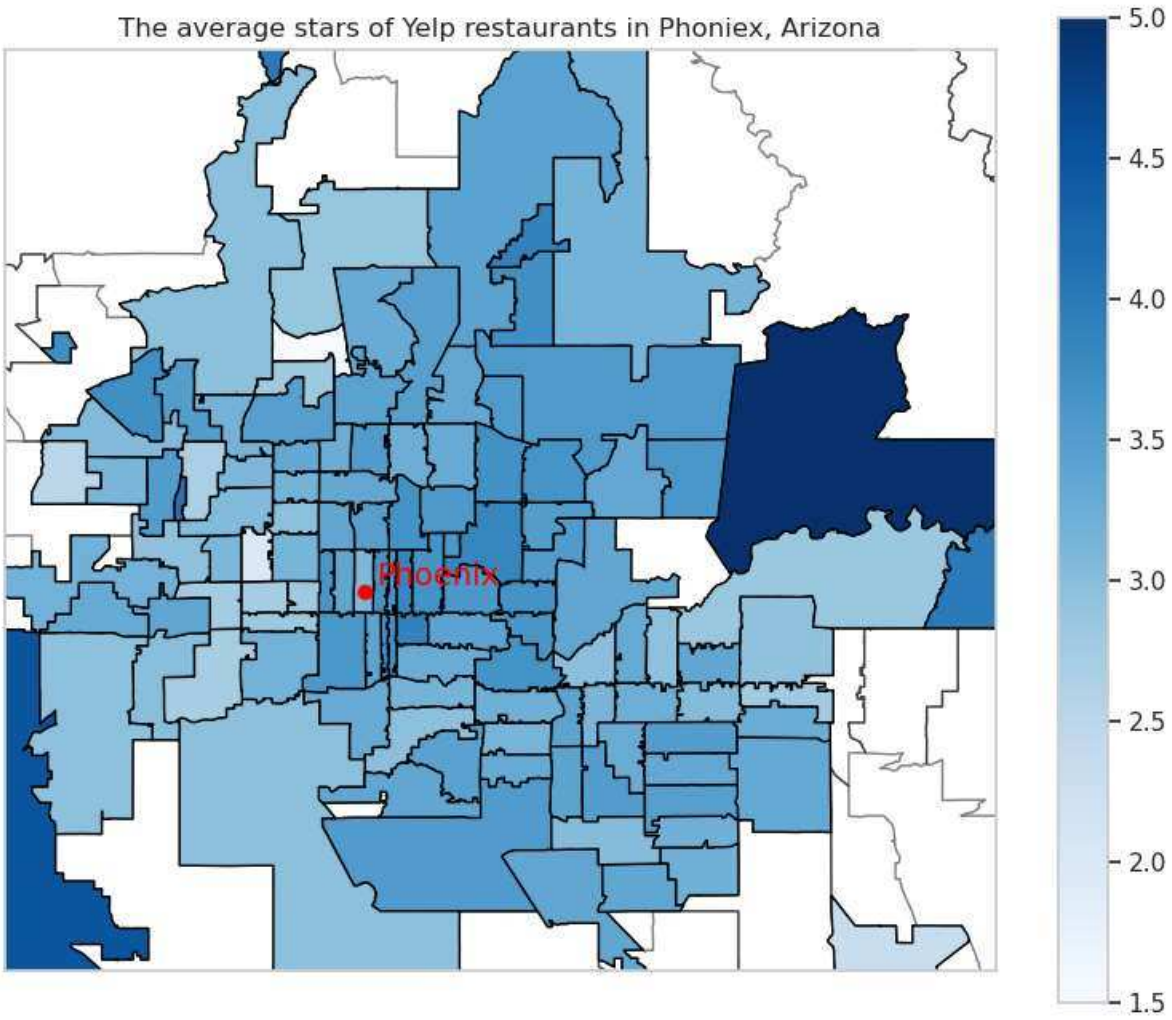


In [40]:
```
# Population density
fig, ax = plt.subplots(figsize=(10, 8))

arizona_state_df.plot(ax=ax, edgecolor="black", color="none")
arizona_zip_df.plot(ax=ax, edgecolor="black", color="none", alpha=0.5)

zip_states_pop.plot(
    ax=ax, edgecolor='black', column='population_density', legend=True, cmap='Blues
    vmin=0, vmax=0.45
)

phoenix_coords = (33.5, -112.1)
ax.scatter(phoenix_coords[1], phoenix_coords[0], color='red')
ax.text(phoenix_coords[1] + 0.025, phoenix_coords[0]-0.007, 'Phoenix', fontsize=14,
ax.set_xlim([-112.5, -111.4])
ax.set_ylim([33.15, 34.0])
plt.title("The Population Density in Phoniex, Arizona")
ax.set_xticks([])
ax.set_yticks([])
plt.xlabel("")
```

```
plt.ylabel("")

plt.show()
```



In [41]:
```
#Average stars
fig, ax = plt.subplots(figsize=(10, 8))
arizona_state_df.plot(ax=ax, edgecolor="black", color="none")
arizona_zip_df.plot(ax=ax, edgecolor="black", color="none", alpha=0.5)

zip_states_star_avg.plot(
    ax=ax, edgecolor='black', column='stars', legend=True, cmap='Blues',
    vmin=1.5, vmax=5
)

phoenix_coords = (33.5, -112.1)
ax.scatter(phoenix_coords[1], phoenix_coords[0], color='red')
ax.text(phoenix_coords[1] + 0.011, phoenix_coords[0]+0.005, 'Phoenix', fontsize=14,
ax.set_xlim([-112.5, -111.4])
ax.set_ylim([33.15, 34.0])
plt.title("The average stars of Yelp restaurants in Phoniex, Arizona")
ax.set_xticks([])
ax.set_yticks([])
plt.xlabel("")
plt.ylabel("")

plt.show()
```



Case Study: Urban Concentration and Restaurant Ratings in Phoenix

Through the intersection of population density and average Yelp restaurant ratings, a captivating spatial narrative emerges on the maps of Phoenix, Arizona. As we explore this dynamic landscape, several crucial observations come to light:

1. Population Density and Geographical Patterns:

- The map meticulously illustrates the spatial distribution of population density around Phoenix, revealing vibrant clusters of red in areas closer to the west of the city center.
- These high-density zones are intriguingly encircled by expanses marked by cooler blue hues, indicative of lower population density.

2. Influence on Restaurant Ratings:

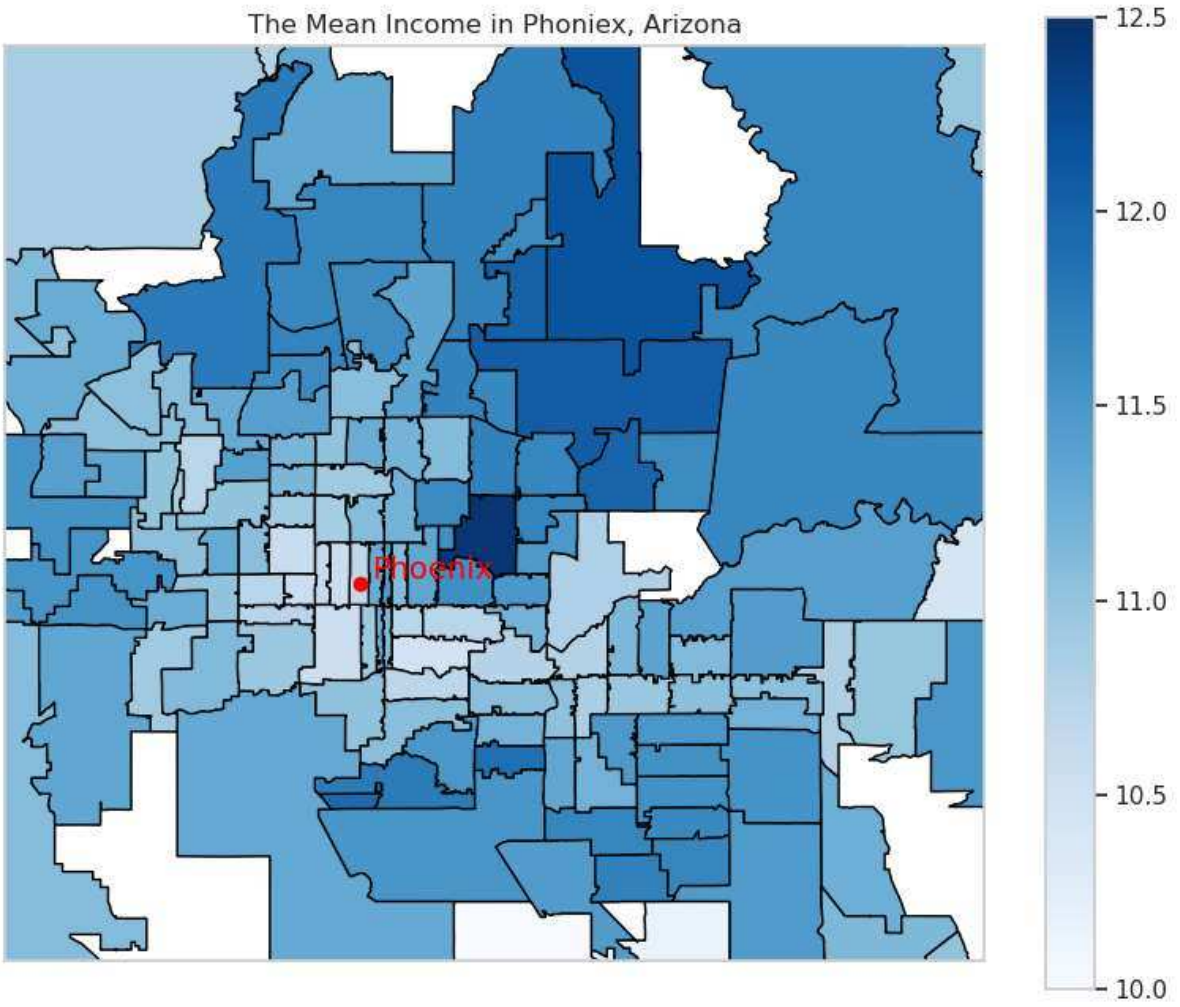
- Curiously, the areas characterized by higher population density exhibit a distinctive pattern in terms of average Yelp restaurant ratings.
- Within these regions of heightened urban concentration, the ratings transition from neutral white to cooler blue tones, indicating a lower average ratings (lower than 3).
- Conversely, the peripheral areas surrounding the high-density zones are characterized by a dominant presence of warmer red shades, suggesting more favorable restaurant ratings (stars of 3.5 to 4.0).

- This comparison highlights an intriguing dynamic that may be related to customer tastes and dining experiences between urban centres and the areas that surround them.

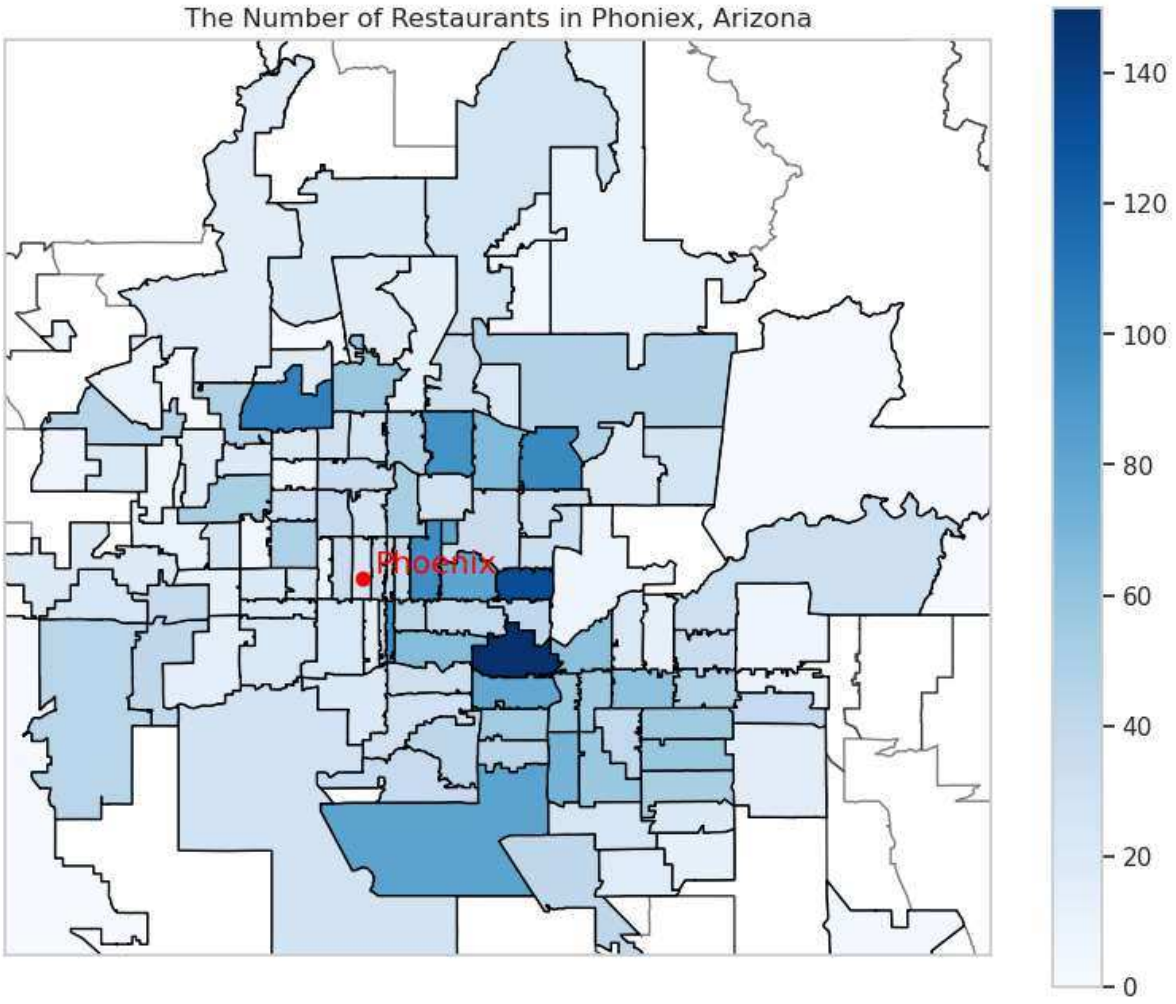
In summary, the fusion of population density and average Yelp restaurant ratings on the map offers a profound glimpse into the dynamics of urban concentration, dining experiences, and customer choices in Phoenix. This visualization provides a unique lens through which to explore the complex interaction between demographic patterns and restaurant performance, shedding light on the intricate interplay that shapes the city's culinary landscape.

Mapping of New merged dataset

In [42]:
```
# Mean Income
fig, ax = plt.subplots(figsize=(10, 8))
arizona_state_df.plot(ax=ax, edgecolor="black", color="none")
arizona_zip_df.plot(ax=ax, edgecolor="black", color="none", alpha=0.5)
zip_states_Income.plot(
    ax=ax, edgecolor='black', column='Mean_Income', legend=True, cmap='Blues',
    vmin=10, vmax=12.5
)

phoenix_coords = (33.5, -112.1)
ax.scatter(phoenix_coords[1], phoenix_coords[0], color='red')
ax.text(phoenix_coords[1] + 0.011, phoenix_coords[0]+0.005, 'Phoenix', fontsize=14,
ax.set_xlim([-112.5, -111.4])
ax.set_ylim([33.15, 34.0])
plt.title("The Mean Income in Phoniex, Arizona")
ax.set_xticks([])
ax.set_yticks([])
plt.xlabel("")
plt.ylabel("")

plt.show()
```

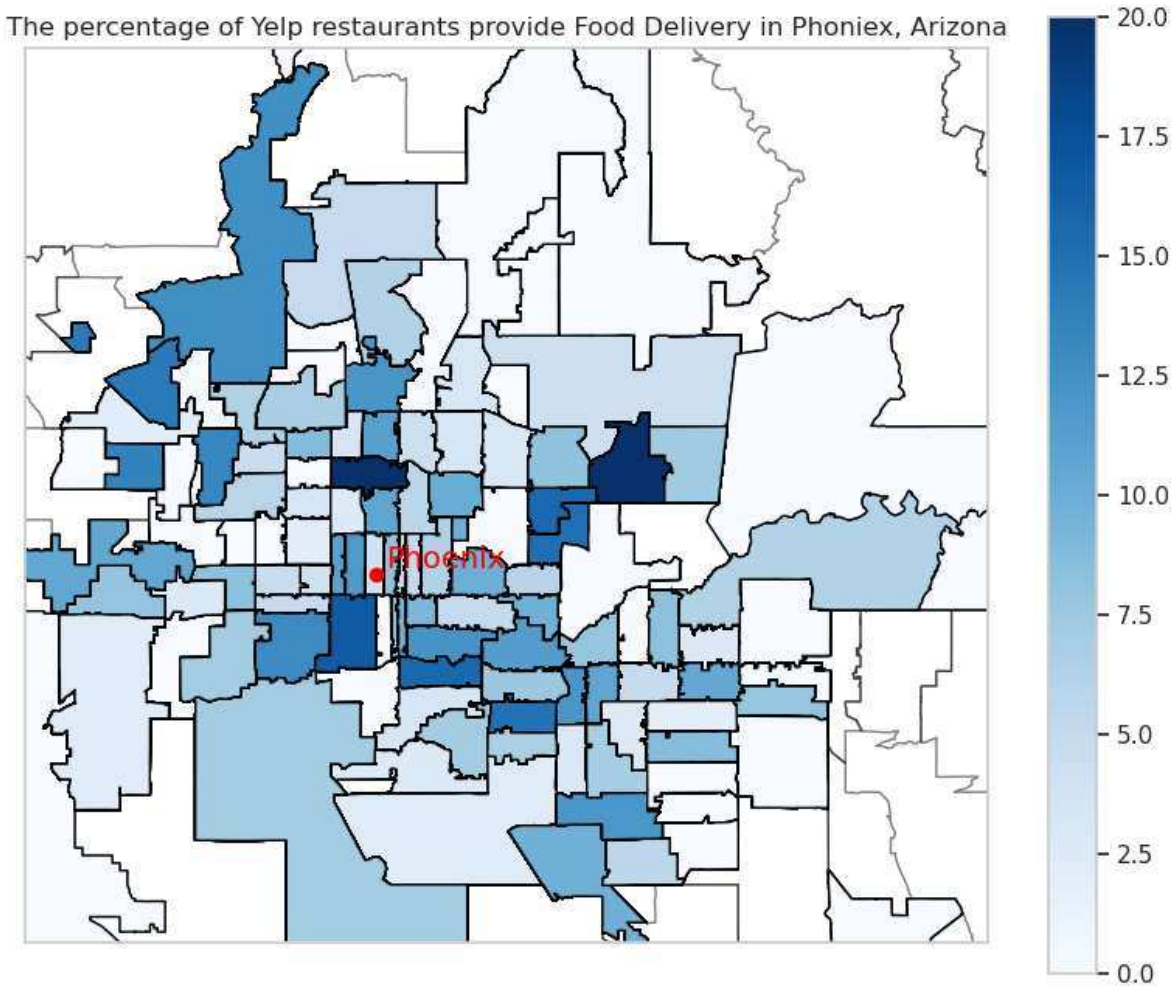


In [43]:
```
# Count
fig, ax = plt.subplots(figsize=(10, 8))
arizona_state_df.plot(ax=ax, edgecolor="black", color="none")
arizona_zip_df.plot(ax=ax, edgecolor="black", color="none", alpha=0.5)
zip_states_count.plot(
    ax=ax, edgecolor='black', column='count', legend=True, cmap='Blues',
    vmin=0, vmax=150
)

phoenix_coords = (33.5, -112.1)
ax.scatter(phoenix_coords[1], phoenix_coords[0], color='red')
ax.text(phoenix_coords[1] + 0.011, phoenix_coords[0]+0.005, 'Phoenix', fontsize=14,
ax.set_xlim([-112.5, -111.4])
ax.set_ylim([33.15, 34.0])
plt.title("The Number of Restaurants in Phoniex, Arizona")
ax.set_xticks([])
ax.set_yticks([])
plt.xlabel("")
plt.ylabel("")

plt.show()
```



In [44]:

```
# percentage_delivery
fig, ax = plt.subplots(figsize=(10, 8))
arizona_state_df.plot(ax=ax, edgecolor="black", color="none")
arizona_zip_df.plot(ax=ax, edgecolor="black", color="none", alpha=0.5)
zip_states_nfd.plot(
    ax=ax, edgecolor='black', column='percentage_delivery', legend=True, cmap='Blue
    vmin=0, vmax=20
)

phoenix_coords = (33.5, -112.1)
ax.scatter(phoenix_coords[1], phoenix_coords[0], color='red')
ax.text(phoenix_coords[1] + 0.011, phoenix_coords[0]+0.005, 'Phoenix', fontsize=14,
ax.set_xlim([-112.5, -111.4])
ax.set_ylim([33.15, 34.0])
plt.title("The percentage of Yelp restaurants provide Food Delivery in Phoniex, Ariz
ax.set_xticks([])
ax.set_yticks([])
plt.xlabel("")
plt.ylabel("")

plt.show()
```



These three supplementary maps offer a deeper comprehension of the analysis by presenting summary statistics and data visualization for three pivotal variables within my study. These variables include the Mean Household Income within each zip code, the count of restaurants per zip code, and the percentage of restaurants that offer food delivery services. These maps are instrumental in enhancing our grasp of how these variables interrelate with the primary factors of population density and Yelp star ratings. By further scrutinizing these relationships, we can enrich our understanding of the research question, facilitating more informed preparation for subsequent regression analyses.

Project Three

OLS Regression

The relationship between the four variables I mentioned before (population density, household income, age, sex ratio) and whether or not a restaurant provides food delivery services with Yelp restaurant ratings is likely to be complex and may not be strictly linear. I will break down each variable and its potential relationship then do the regression :

1. **Population Density**: Population density could have a non-linear relationship with Yelp restaurant ratings. Initially, as population density increases, there might be more potential customers for restaurants, which could lead to higher ratings due to increased demand and competition the relationship should be linear before this point. However, there could be a point of saturation where high population density leads to more intense competition and potentially lower ratings due to higher expectations from customers. Thus the regression tree could be applied to this variable.

2. **Household Income**: Household income could have a linear relationship with Yelp ratings. As household income rises, restaurants in areas with higher income levels are more likely to receive elevated Yelp ratings. This phenomenon can be attributed to the increased ability of individuals with higher incomes to afford dining at upscale establishments, leading to higher expectations and greater appreciation for quality. These customers' frequent dining experiences and exposure to diverse culinary offerings could contribute to a more discerning evaluation of restaurant experiences, thus positively influencing ratings. Moreover, restaurants in such neighborhoods might invest more in exceptional service, ambiance, and culinary innovation, all of which contribute to a superior dining experience and, consequently, higher Yelp ratings.

3. **Age and Sex Ratio**: These demographic variables are likely to have non-linear relationships with Yelp ratings. The non-linear relationship between the percentage of people aged 15 to 24 and Yelp ratings acknowledges that dining preferences and expectations can vary significantly across different age groups. Restaurants situated in areas with a higher percentage of younger individuals might experience non-linear shifts in ratings. For instance, a greater concentration of young adults could lead to both higher and lower ratings, as this demographic tends to seek diverse and often trend-driven dining experiences, affecting their feedback in non-linear ways. Similarly, the male-to-female ratio's non-linear influence on Yelp ratings underscores the intricate interplay between gender dynamics and dining habits. While the direction of the relationship might not be immediately apparent, it reflects the complexity of how gender-specific preferences and cultural norms intersect with restaurant experiences. A significantly skewed ratio might lead to distinct patterns in customer behavior, impacting ratings in non-linear ways due to the differing expectations, preferences, and social dynamics among various gender compositions. Cultural and generational differences further contribute to the non-linear relationship between age, sex ratio, and

Yelp ratings. Factors such as cultural background, lifestyle, and societal norms can lead to nuanced shifts in dining choices and satisfaction thresholds.

4. **Food Delivery Services**: The relationship between providing food delivery services and Yelp ratings could also be linear. Offering food delivery can attract a broader customer base and potentially lead to higher ratings due to convenience.

In practical implementation, data analysis is paramount to discerning the precise nuances of these relationships. I've strategized to employ techniques, including regression analysis and machine learning, to comprehensively evaluate the intricate connections between the variables and their influence on Yelp ratings. Beyond simple linear associations, I intend to delve into interaction effects, recognizing that the impact of one variable can be contingent on the values of another. In order to answer the research question, I will first focus on the dynamics of the interaction between population density and food delivery services.

In [4]:
```
models = []
Final_regression_merged['interaction'] = Final_regression_merged['population_density
```

In [5]:
```
# 1. Stars ~ Population density + Food delivery  Final_regression
X = Final_regression_merged[['population_density', 'has_food_delivery']]
y = Final_regression_merged['stars']
X = sm.add_constant(X)
model1 = sm.OLS(y, X).fit()
models.append(model1)

# 2. Stars ~ Population density * Food delivery + Population density + Food deliver
X = Final_regression_merged[['population_density', 'has_food_delivery', 'interaction
y = Final_regression_merged['stars']
X = sm.add_constant(X)
model2 = sm.OLS(y, X).fit()
models.append(model2)

# 3. Stars ~ Population density * Food delivery + Population density + Food deliver
X = Final_regression_merged[['population_density', 'has_food_delivery', 'interaction
y = Final_regression_merged['stars']
X = sm.add_constant(X)
model3 = sm.OLS(y, X).fit()
models.append(model3)

# 3b. Stars ~ Population density + Food delivery + Income + Count
X = Final_regression_merged[['population_density', 'has_food_delivery', 'Mean_Income
y = Final_regression_merged['stars']
X = sm.add_constant(X)
model3b = sm.OLS(y, X).fit()
models.append(model3b)

# 4. Stars ~ Population density * Food delivery + Population density + Food deliver
X = Final_regression_merged[['population_density', 'has_food_delivery', 'interaction
y = Final_regression_merged['stars']
X = sm.add_constant(X)
model4 = sm.OLS(y, X).fit()
models.append(model4)

# 4b. Stars ~ Population density + Food delivery + Income + Age + Sex
X = Final_regression_merged[['population_density', 'has_food_delivery', 'Mean_Income
y = Final_regression_merged['stars']
X = sm.add_constant(X)
model4b = sm.OLS(y, X).fit()
models.append(model4b)
```

```
# 5. Stars ~ Population density * Food delivery + Population density + Food deliver
X = Final_regression_merged[['population_density', 'has_food_delivery', 'interaction
y = Final_regression_merged['stars']
X = sm.add_constant(X)
model5 = sm.OLS(y, X).fit()
models.append(model5)

# 5b. Stars ~ Population density + Food delivery + Income + Age + Sex + Count
X = Final_regression_merged[['population_density', 'has_food_delivery', 'Mean_Income
y = Final_regression_merged['stars']
X = sm.add_constant(X)
model5b = sm.OLS(y, X).fit()
models.append(model5b)

# 6. Stars ~ Population density * Food delivery + Population density + Food deliver
X = Final_regression_merged[['population_density', 'has_food_delivery', 'interaction
y = Final_regression_merged['stars']
X = sm.add_constant(X)
model6 = sm.OLS(y, X).fit()
models.append(model6)

# 6b. Stars ~ Population density + Food delivery + Income + Age + Sex + Count + per
X = Final_regression_merged[['population_density', 'has_food_delivery', 'Mean_Income
y = Final_regression_merged['stars']
X = sm.add_constant(X)
model6b = sm.OLS(y, X).fit()
models.append(model6b)
```

The regression models I've chosen to run reflect a thoughtful approach to understanding the economic factors that potentially influence restaurant ratings. Each model builds upon the previous one, gradually incorporating additional independent variables. Here's the economic reasoning behind running each of these regressions:

1. **Stars ~ Population Density + Food Delivery**:
    - This initial regression investigates the impact of two fundamental factors - population density and the presence of food delivery services - on restaurant ratings. Population density can reflect market competition, while the availability of food delivery services can affect convenience for customers. By isolating these two variables, you gain insight into their individual contributions to restaurant ratings.
2. **Stars ~ Population Density + Food Delivery + Income + Count**:
    - Adding income and count variables introduces economic capacity and market size considerations. Income levels can reflect the average spending power of residents in an area, influencing the type of restaurants and their quality. Count, which might represent the number of restaurants in an area, can indicate competition levels. Including these variables allows me to explore how economic factors and market size relate to restaurant ratings alongside population density and food delivery.
3. **Stars ~ Population Density + Food Delivery + Income + Age + Sex**:
    - Incorporating age and sex variables recognizes the influence of demographic factors on dining preferences and expectations. Different percentage of age groups and genders might have varying dining habits and satisfaction thresholds, impacting restaurant ratings. This model delves deeper into the social and demographic aspects of restaurant choice and rating behaviors.

4. **Stars ~ Population Density + Food Delivery + Income + Age + Sex + Count**:
    - This regression includes all the listed variables from model 3 and the 'count' variable from Model 3. Comparing with model4, this 'count' variable adds another layer of competition to the analysis. More restaurants can lead to intensified competition, potentially affecting the quality and ratings of individual establishments. This comprehensive model accounts for various economic, demographic, and market factors simultaneously.
5. **Stars ~ Population Density + Food Delivery + Income + Age + Sex + Count + Percentage of delivery**:
    - The final regression includes all the listed variables from model 4 and the 'percentage_delivery' variable. Which is the varaible that shows the percentage of restaurants that provide the food delivery service among all the Yelp restaurant in a Zip code area.

In summary, each regression model builds upon the previous one by incorporating additional economic and demographic variables to provide a more comprehensive understanding of the factors influencing restaurant ratings on Yelp. The goal is to uncover the complex interplay between these factors and their collective impact on restaurant performance.

In [6]:
```
stargazer1 = Stargazer([model1, model3b, model4b, model5b, model6b])
model_names = ['Model 1', 'Model 2', 'Model 3', 'Model 4', 'Model 5']
stargazer1.custom_columns(model_names)
stargazer1.show_model_numbers(False)
HTML(stargazer1.render_html())
```

Out[6]:

	Dependent variable: stars				
	Model 1	Model 2	Model 3	Model 4	Model 5
Mean_Income		0.280***	0.352***	0.245***	0.255***
		(0.047)	(0.052)	(0.054)	(0.055)
Pop_15_24			-0.002	-0.013***	-0.013***
			(0.003)	(0.003)	(0.003)
Sex_ratio			0.016***	0.015***	0.015***
			(0.002)	(0.002)	(0.002)
const	3.346***	-0.054	-2.236***	-0.884	-1.016
	(0.033)	(0.553)	(0.632)	(0.663)	(0.674)
count		0.003***		0.003***	0.003***
		(0.000)		(0.001)	(0.001)
has_food_delivery	0.341***	0.342***	0.347***	0.342***	0.331***
	(0.058)	(0.058)	(0.058)	(0.057)	(0.058)
percentage_delivery					0.004
					(0.003)
population_density	-0.015	0.404*	0.370	0.145	0.166
	(0.192)	(0.230)	(0.231)	(0.232)	(0.233)
Observations	4149	4149	4149	4149	4149
R^2	0.000	0.297	0.307	0.425	0.437
Adjusted R^2	-0.000	0.296	0.307	0.424	0.436
Residual Std. Error	0.273 (df=4146)	0.229 (df=4144)	0.227 (df=4143)	0.207 (df=4142)	0.205 (df=4141)
F Statistic	0.793 (df=2; 4146)	436.762*** (df=4; 4144)	367.871*** (df=5; 4143)	509.367*** (df=6; 4142)	458.702*** (df=7; 4141)
Note:					*p<0.1; **p<0.05; ***p<0.01

In the second table, I introduces an interaction term (Population Density * Food Delivery) to assess whether the joint effect of population density and food delivery is different from their individual effects. It recognizes that these two variables might not have a simple additive relationship. For example, in highly dense areas, the presence of food delivery might have a more significant impact than in less dense areas due to increased competition.

1b. **Stars ~ Population Density * Food Delivery + Population Density + Food Delivery**

2b. **Stars ~ Population Density * Food Delivery + Population Density + Food Delivery + Income + Count**

3b. **Stars ~ Population Density * Food Delivery + Population Density + Food Delivery + Income + Age + Sex**

4b. **Stars ~ Population Density * Food Delivery + Population Density + Food Delivery + Income + Age + Sex + Count**

5b. **Stars ~ Population Density * Food Delivery + Population Density + Food Delivery + Income + Age + Sex + Count + Percentage of delivery**

In [7]:
```
stargazer2 = Stargazer([model2, model3,model4, model5, model6])
model_names = ['Model 1b', 'Model 2b', 'Model 3b', 'Model 4b', 'Model 5b']
stargazer2.custom_columns(model_names)
stargazer2.show_model_numbers(False)
HTML(stargazer2.render_html())
```

Out[7]:

	Dependent variable: stars				
	Model 1b	Model 2b	Model 3b	Model 4b	Model 5b
Mean_Income		0.281***	0.354***	0.248***	0.257***
		(0.047)	(0.052)	(0.054)	(0.055)
Pop_15_24			-0.002	-0.013***	-0.013***
			(0.003)	(0.003)	(0.003)
Sex_ratio			0.016***	0.015***	0.015***
			(0.002)	(0.002)	(0.002)
const	3.339***	-0.078	-2.287***	-0.937	-1.058
	(0.034)	(0.553)	(0.632)	(0.664)	(0.675)
count		0.003***		0.003***	0.003***
		(0.000)		(0.001)	(0.001)
has_food_delivery	0.460***	0.487***	0.533***	0.509***	0.492***
	(0.139)	(0.137)	(0.137)	(0.137)	(0.138)
interaction	-0.761	-0.924	-1.186	-1.067	-1.020
	(0.804)	(0.794)	(0.795)	(0.791)	(0.793)
percentage_delivery					0.003
					(0.003)
population_density	0.031	0.463**	0.444*	0.213	0.229
	(0.198)	(0.235)	(0.236)	(0.237)	(0.238)
Observations	4149	4149	4149	4149	4149
R^2	0.001	0.297	0.308	0.425	0.437
Adjusted R^2	0.000	0.296	0.307	0.424	0.436
Residual Std. Error	0.273 (df=4145)	0.229 (df=4143)	0.227 (df=4142)	0.207 (df=4141)	0.205 (df=4140)
F Statistic	1.077 (df=3; 4145)	349.470*** (df=5; 4143)	306.584*** (df=6; 4142)	436.495*** (df=7; 4141)	401.332*** (df=8; 4140)
Note:					*p<0.1; **p<0.05; ***p<0.01

After investigating the result, I will choose the Model 2 in Table 1 as my preferred regression, by listing several advantages of this model compared to the other four models:

$$\text{Stars} = \beta_0 + \beta_1 \text{population\_density} + \beta_2 \text{has\_food\_delivery} + \beta_3 \text{Mean\_Income} + \beta_4 \text{count} + \epsilon$$

Advantages of Model 2:

1. **Economic Interpretability**:
    - In addition to Model 1, Model 2 includes the key economic indicator "Mean Income" and the number of restaurants ("count"). These variables are often directly linked to consumer behaviour and market dynamics. "Mean Income" reflects the economic capacity of the local population, while "count" represents the level of competition in the area. These factors have straightforward economic interpretations that align well with the context of the restaurant industry.
2. **Simplicity and Relevance**:
    - Model 2 strikes a balance between complexity and relevance. It includes important economic indicators and restaurant count without adding too many other variables. This simplicity makes the model more interpretable and easier to communicate to restaurant owners/investors. Also, these variables are easy to understand and apply in real-world scenarios. The results from Model 2 can be practically useful for restaurant owners and policymakers seeking insights into factors that influence restaurant ratings.
3. **High R-squared**:
    - Model 2 has a relatively high adjusted R-squared among these models, indicating that it explains a greater proportion of the variance in "stars." This suggests that the inclusion of "Mean Income" and "count" contributes to a better fit of the model to the data, compared with Model 1.
4. **Statistical Significance**:
    - In Model 2, all variables exhibit statistically significant coefficients, barring the interaction term. This underscores their substantial contributions in elucidating the fluctuations in restaurant ratings. Significantly, Model 2 is the only model having the "population density" variable as significant among all five regression models, which holds pivotal importance as the cornerstone of my research question.

As I mentioned in the "Advantages of Model 2" section, there are several measures that can be used to assess the performance of the regressions. For instance the R-squared, Adjusted R-squared, Residual Standard Error (RSE), F-statistic, the statistical significance (p-value) and the Mean Squared Error (MSE):

1. **R-squared (Coefficient of Determination)**: R-squared measures the proportion of the variance in the dependent variable that is explained by the independent variables in the model. It ranges from 0 to 1, where higher values indicate that the model explains a larger portion of the variability in the dependent variable. (Miles, 2005) An R-squared value of, for instance, in our model 0.297 means that 29.7% of the variability in the dependent variable is explained by the independent variables in the model. However, a high R-squared does not necessarily mean a good model fit, as it doesn't account for overfitting or the quality of predictions.

2. **Adjusted R-squared**: Adjusted R-squared adjusts the R-squared value for the number of predictors in the model. It penalizes the inclusion of unnecessary variables that may not contribute significantly to the model's fit. (Miles, 2005)

3. **Residual Standard Error (RSE)**: RSE measures the average distance between the observed values and the values predicted by the model. It gives an idea of how well the model's predictions match the actual data points. Smaller values of RSE suggest that the model's predictions are closer to the actual data points, indicating a better fit.In the Model 2, I have teh RSE = 0.229 which is relatively small, means better prediction.

4. **F-statistic**: The F-statistic tests the overall significance of the model. It compares the variation explained by the model to the variation not explained by the model. A significant F-statistic suggests that at least one independent variable in the model contributes to explaining the dependent variable. (Pope & Webster, 1972) The significant F-statistic in my model supports the validity of the model as a whole, indicating that the model's explanatory variables are jointly influencing the dependent variable.

5. **p-values of Coefficients**: The p-values associated with each coefficient indicate whether the corresponding independent variable has a statistically significant effect on the dependent variable. (Frost, 2017) A low p-value (typically p < 0.05 or p < 0.01) suggests that the variable is likely contributing to the model's explanatory power, which is true for all my interested variables.

6. **Mean Squared Error (MSE)**: The mean squared error (MSE) is a measure used in statistics to evaluate the performance of an estimator or predictor. It calculates the average of the squares of the errors, where an error is defined as the difference between an estimated value and the actual value.In my regression model 2, the MSE is about 0.05240.0419 which is considered a small number.This is my formula that calculate the MSE :

$$\frac{1}{N}\sum_{i=1}^{N}((\text{Stars}_i) - (\beta_0 + \beta_1 \text{population\_density} + \beta_2 \text{has\_food\_delivery} + \beta_3 \text{Mean\_Income} + \beta_4 \text{count}))^2$$

In [8]:
```
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)
X = Final_regression_merged[['population_density', 'has_food_delivery', 'Mean_Income
y = Final_regression_merged['stars']
X = sm.add_constant(X)
model3b = sm.OLS(y, X).fit()
y_pred_linear = model3b.predict(X)
from sklearn import metrics
full_mse = metrics.mean_squared_error(y, y_pred_linear)
print('Mean Squared Error:', full_mse)
```

```
Mean Squared Error: 0.868259958903814
```

Therefore, with the result from the regression model, I could conclude the correlations between the dependent variable ("stars") and the independent variables in the Model 3:

$$\text{Stars} = -0.054 + 0.404 * \text{population\_density} + 0.342 * \text{has\_food\_delivery} + 0.280 * \text{Mean\_Income} + 0.003 * \text{count} + \epsilon$$

1. **Population Density**: The population density ("population_density") in an area demonstrates a statistically significant positive correlation with restaurant ratings

("stars"). This suggests that regions with higher population densities tend to have restaurants with higher Yelp ratings.

2. **Food Delivery Services**: The presence of food delivery services ("has_food_delivery") has a statistically significant positive correlation with restaurant ratings ("stars"). This implies that restaurants offering food delivery tend to receive higher Yelp ratings for extra 0.35 stars.

3. **Mean Income**: The average income ("Mean_Income") of the local population shows a statistically significant positive correlation with restaurant ratings ("stars"). This suggests that restaurants located in areas with higher average incomes tend to receive higher ratings on Yelp.

4. **Number of Restaurants**: The count of restaurants ("count") in a specific area exhibits a statistically significant positive correlation with restaurant ratings ("stars"). This indicates that regions with a higher concentration of restaurants are associated with higher Yelp ratings.

These correlations provide insights into how each independent variable is associated with the dependent variable, restaurant ratings. Keep in mind that while these correlations indicate relationships, the direction and strength of the associations might be influenced by other factors as well.

Casual Analysis: Difference-in-Differences (DiD) Analysis:

In Arizona, there has been an observed positive correlation between the availability of food delivery services and higher Yelp restaurant ratings, offering an intriguing area for investigation. While correlation suggests an association, proving causation requires a methodological strategy that takes into account potential confounding factors. I intend to apply a Difference-in-Differences (DiD) Analysis to delve deeper into causation. To reduce selection bias and other confounding factors that might affect restaurant ratings, this technique compares changes in ratings over time after receiving the intervention (food delivery services) and those that did not. This requires me to include more individual business-related variables as control factors to strengthen the validity of the analysis.

I am however aware of the difficulties presented by the lack of comprehensive business data as there are too many NAs in the business detail dataset provided by the Yelp Kaggel challenge, which might make it more difficult to establish treatment and control groups and might even introduce bias.

Exploring the causal relationship using a Difference-in-Differences (DiD) analysis to understand shifts in restaurant ratings over time is both intriguing and meaningful. This approach can unveil how external factors like the introduction of food delivery services or changes in population density might influence restaurant ratings. To ensure the establishment of solid casual relationships, I plan to utilize datasets from before and after 2017. This temporal perspective allows capturing potential changes that occurred over time, bolstering the validity of findings.

Given current data limitations, I intend to postpone this causal research to the future. This approach allows the gathering of necessary data and ensures robust analysis. This topic holds significant potential to provide insights into restaurant industry dynamics, and with a comprehensive dataset, future research will carry greater weight and relevance.

Machine Learning

After finding my best Linear regression model, I want to dig deeper into my dataset and use the Machine Learning tools to help with the analysis. In this senario, I choose using the Regression Tree. A Regression Tree is a machine learning algorithm that operates on the same basic principle as a decision tree, but it's designed for predicting continuous numerical values, making it suitable for regression tasks. It breaks down the data into smaller and more manageable subsets while considering various features and their values. Each node in the tree represents a decision point, where the data is split based on a selected feature and a corresponding threshold. The ultimate goal is to partition the data into homogeneous segments that can be used to predict the target variable (in this case, restaurant ratings).

1. **Node Splitting**: The tree-building process begins with the entire dataset at the root node. The algorithm assesses each feature's potential to split the data into more distinct and predictive subsets. The best feature and corresponding threshold are chosen based on criteria that minimize the variance of the target variable within each subset (MSE).

which is the objective function for the Regression tree:

- For each region, solve

$$\min_{j,s}\left[\sum_{i:x_{i,j}\leq s, x_i\in R1}(Stars_i-\hat{y}_{R1})^2+\sum_{i:x_{i,j}>s, x_i\in R2}(Stars_i-\hat{y}_{R2})^2\right]$$

- Repeat with each of the two smaller rectangles.
- Stop when $|R|=$ some chosen minimum size or when depth of tree $=$ some chosen maximum.
- Prune tree.

$$\min_{tree\subset T}[\sum(\hat{f}(x)-y)^2+\alpha|\mathbf{T}|]$$

1. $(\sum(\hat{f}(x)-y)^2)$: This part represents the sum of squared differences between the predicted values $\hat{f}(x)$ (where x represents the feature values including the independent variables:'population_density', 'has_food_delivery', 'Mean_Income', 'count', 'Pop_15_24', 'Sex_ratio', 'percentage_delivery') and the actual target values y (Yelp restaurants' star ratings). This term aims to minimize the discrepancy between predictions and actual ratings.

2. $\alpha|\mathbf{T}|$: This term introduces regularization to the objective function. Regularization helps prevent overfitting by discouraging the tree from becoming too complex. $|\mathrm{T}|$ represents the number of leaf nodes in the tree, and α is a hyperparameter that

controls the strength of the regularization. A larger α value would lead to a simpler tree with fewer leaf nodes.

In the context of the Regression Tree model, the regularization parameter α controls the complexity of the tree by adding a penalty term to the objective function that encourages simpler trees with fewer leaf nodes.

1. **Small α (Less Regularization)**:
    - When α is small, the penalty for having additional leaf nodes is minor compared to the goal of minimizing the sum of squared differences.
    - This can lead to the model fitting the training data more closely, potentially capturing noise or small fluctuations in the data. The tree might become overly complex, and it could result in overfitting.
    - You might observe that the model performs very well on the training data but might struggle to generalize to new, unseen data.
2. **Large α (More Regularization)**:
    - Increasing α places a stronger penalty on having more leaf nodes, which encourages the model to stay simpler and avoid overfitting.
    - The tree will have fewer leaf nodes, leading to a more generalized representation of the underlying patterns in the data.
    - While the model might not fit the training data as closely as with smaller α, it is likely to perform better on new data. It's less prone to overfitting and is expected to have better generalization.

In summary, the regularization parameter α controls the trade-off between fitting the data closely and maintaining model simplicity. By increasing α, you prioritize simpler models that are less prone to overfitting, while decreasing α allows the model to fit the data more closely but at the risk of overfitting. The optimal value of α depends on your specific dataset and the balance you want to strike between model complexity and generalization performance.

The combined objective of $(\sum(\hat{f}(x) - y)^2)$ and $\alpha|\mathbf{T}|$ is to find the tree structure that minimizes the sum of squared differences between predictions and actual values while considering the regularization term to control the complexity of the tree. This trade-off between fitting the data and keeping the model simple helps strike a balance between capturing the underlying patterns in the data and avoiding noise or overfitting.

1. **Recursive Partitioning**: Once a feature and threshold are selected, the data is divided into two subsets based on whether the feature's value for a given observation is above or below the threshold. These subsets become the child nodes of the current node. This process is recursively applied to each child node, creating a hierarchical structure of nodes and subnodes.

2. **Stopping Criteria**: The recursive splitting process continues until certain stopping criteria are met (T). The criteria in my model is reaching the maximum tree depth (max_depth).

3. **Leaf Nodes and Predictions**: The process ends when no further splits are possible or when stopping criteria are satisfied. At this point, the terminal nodes, known as leaf nodes, contain a subset of data. The average or weighted average of the target variable within a leaf node becomes the prediction for that node's data.

Regression Trees partition data into increasingly homogeneous subsets by repeatedly splitting based on the most predictive features and thresholds. This allows for the creation of a predictive model capable of estimating continuous target variables, making it a powerful tool for understanding complex relationships within my data, explaining and predicting the Yelp restaurant ratings.

Mitigating overfitting is crucial for developing a robust model. One effective approach is to split the data randomly into a training set and a validation set. This allows me to train my model on one subset and then assess its performance on another, helping to detect any signs of overfitting.Moreover, I will employ a technique known as n-fold cross-validation. Here, I split my training set into n subsets or "folds." I will train my model n times, each time using 'n-1' of the folds as the training data and the remaining fold for validation. This process is repeated for each fold, ensuring that every data point is both used for training and validation. This comprehensive assessment provides a better estimate of my model's performance and helps me select the best T. The optimal stopping criteria T, which is assigned to the maximum tree depth (max_depth), that minimizes the Mean Squared Error (MSE). By systematically varying the max_depth parameter and evaluating the MSE across multiple cross-validation runs, I can pinpoint the value of max_depth that yields the lowest MSE, thus ensuring a more balanced model between bias and variance.

In [9]:
```
# The cross validation
from sklearn import tree
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.ensemble import BaggingClassifier, RandomForestClassifier, BaggingRegr
from sklearn.metrics import mean_squared_error,confusion_matrix, classification_rep
from sklearn.model_selection import GridSearchCV

X = Final_regression_merged[['population_density', 'has_food_delivery', 'Mean_Income
y = Final_regression_merged['Total_stars']
X = sm.add_constant(X)

sqft_tree = tree.DecisionTreeRegressor(max_depth=3).fit(X,y)
y_pred_tree = sqft_tree.predict(X)
print('Mean Squared Error:', metrics.mean_squared_error(y, y_pred_tree))

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_stat
```

```
Mean Squared Error: 0.040170927097658925
```

In [10]:
```
simple_tree = tree.DecisionTreeRegressor(max_depth=3).fit(X_train,y_train)
print('Mean Squared Error:', metrics.mean_squared_error(y_test, simple_tree.predict(
```

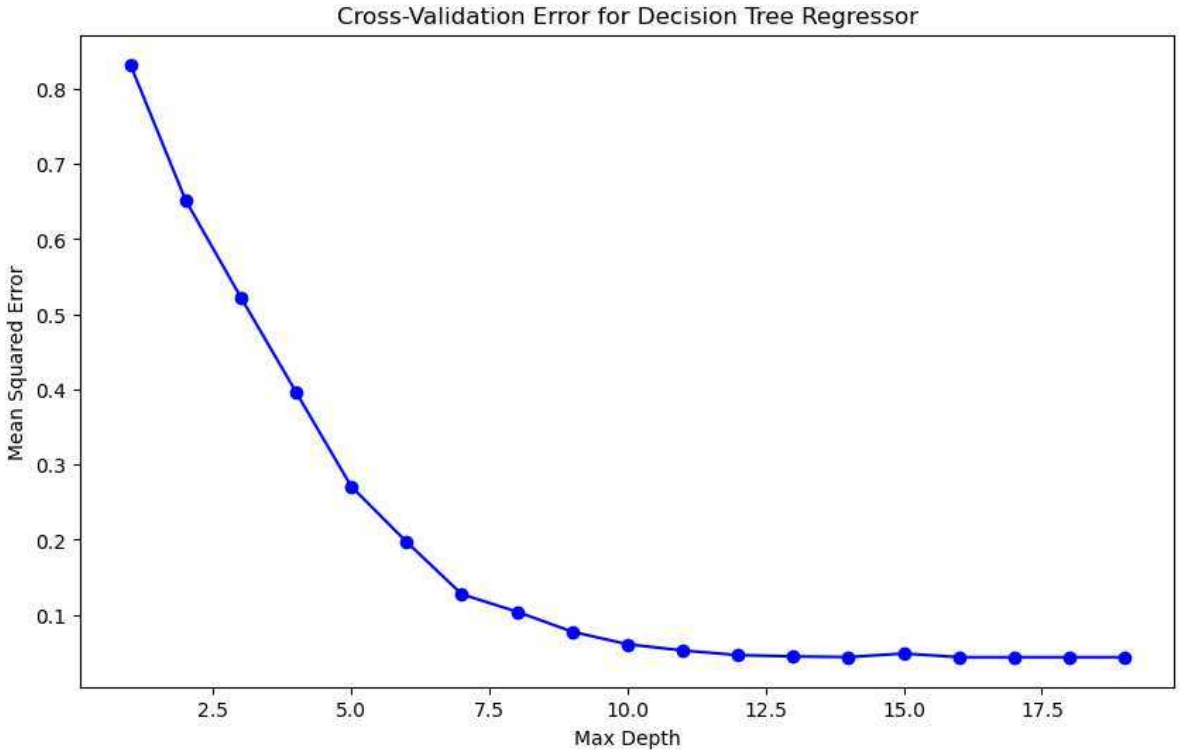
```
Mean Squared Error: 0.045634834823564174
```

In [11]:
```
parameters = {'max_depth':range(1,20)}
clf = GridSearchCV(estimator=tree.DecisionTreeRegressor(random_state=45), param_gri
clf.fit(X=X_train, y=y_train)
best_tree_model = clf.best_estimator_
```

In [12]:
```
best_tree_model
```

Out[12]:
```
DecisionTreeRegressor(max_depth=18, random_state=45)
```

In [13]:
```
# Extract the cross-validation results
mse_scores = 1 - clf.cv_results_['mean_test_score']
depths = parameters['max_depth']

plt.figure(figsize=(10, 6))
plt.plot(depths, mse_scores, marker='o', color='b', label='Mean Squared Error (MSE)'
plt.xlabel('Max Depth')
plt.ylabel('Mean Squared Error')
plt.title('Cross-Validation Error for Decision Tree Regressor')
plt.show()
```



In [14]:
```
# find the error of prediction (MSE)
print('Mean Squared Error:', metrics.mean_squared_error(y_test, best_tree_model.pred
```

```
Mean Squared Error: 0.005360330209727799
```

In [15]:
```
simple_tree = tree.DecisionTreeRegressor(max_depth=3).fit(X_train,y_train)

# find the error of prediction (MSE)
print('Mean Squared Error:', metrics.mean_squared_error(y_test, simple_tree.predict(
```

```
Mean Squared Error: 0.045634834823564056
```

In [16]:
```
sqft_tree = tree.DecisionTreeRegressor(max_depth=10).fit(X_train,y_train)
# use the fitted tree to predict
y_pred_tree = sqft_tree.predict(X)

# find the error of prediction (MSE)
from sklearn import metrics
print('Mean Squared Error:', metrics.mean_squared_error(y_test, sqft_tree.predict(X_
```

```
Mean Squared Error: 0.005718684250263834
```

1. **Initial Decision Tree Model (max_depth=3)**:

   - Mean Squared Error (MSE) on the entire dataset: 0.0402
   - MSE on the testing set: 0.0456

   The initial decision tree model with a maximum depth of 3 performs relatively well on the entire dataset but slightly worse on the testing set. This indicates some overfitting, as the model seems to have memorized the training data rather than generalizing effectively to new data.

2. **Hyperparameter Tuning with GridSearchCV**:

   - The best decision tree model from the grid search has a maximum depth of 18.
   - Cross-validation error decreases as max depth increases, suggesting that more complex trees fit the training data better.

   Similar to the previous analysis, the cross-validation process indicates that deeper trees perform better on the training data. However, since the best model has a max depth of 18, it's crucial to evaluate its generalization performance on the testing set.

3. **Best Model Evaluation (max_depth=18)**:

   - MSE on the testing set with the best model: 0.0054

   The best model from the grid search (max depth of 18) performs exceptionally well on the testing set, indicating that it effectively captures patterns in the data and generalizes well to unseen data. This depth seems to have minimized overfitting.

4. **Comparisons with Other Models**:

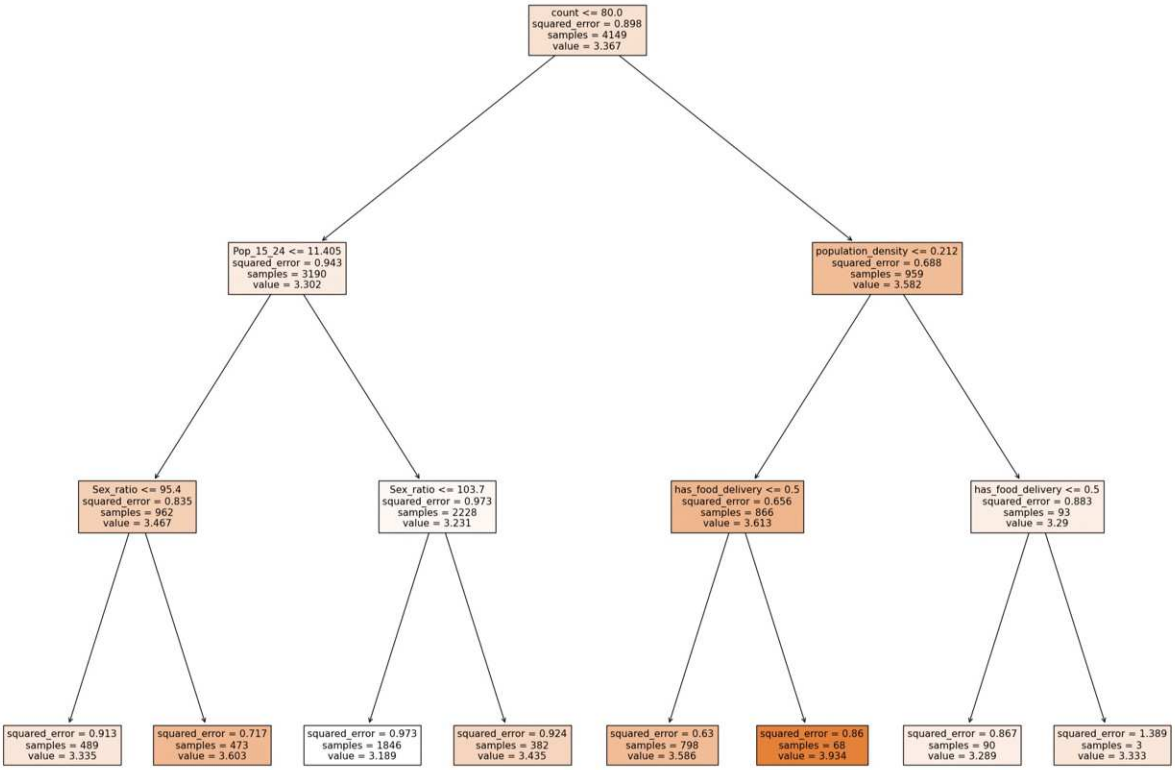
   - The deeper tree (max_depth=10) performs better than the Inital model and slightly worse than the best model (max_depth=18).

Overall, the analysis suggests that the best decision tree model has a maximum depth of 18, which significantly reduces the Mean Squared Error on the testing set. This deeper tree appears to generalize well and provides a strong predictive model for estimating restaurant star ratings based on the given features.

My approach of employing 5-fold cross-validation to determine the optimal max depth has yielded insightful results. The cross-validation process revealed that the Mean Square Error (MSE) is minimized when the max depth is set to 18. Upon examining the graph depicting the relationship between MSE and max depth, a notable observation emerges. After a max depth of 10, the line representing MSE remains relatively steady, suggesting that increasing max depth may not significantly enhance predictive performance and could lead to overfitting. Considering these insights, I have decided to adopt a max depth of 10 for the regression tree. This choice strikes a balance between model complexity and interpretability while also addressing concerns about overfitting. By selecting a max depth that has already demonstrated effectiveness, I am ensuring that my model remains robust and aligned with the data's intricacies, ultimately leading to a more informed and reliable outcome.

In [17]:
```
sqft_tree = tree.DecisionTreeRegressor(max_depth=3).fit(X,y)
y_pred_tree = sqft_tree.predict(X)
```

```
sqrf_fig = plt.figure(figsize=(25,20))
sqrf_fig = tree.plot_tree(sqft_tree, feature_names=X.columns, filled=True)
```



Since depth of 10 has too many brunch to explain, I will only cut the top 3 level to interpret the result. In the regression tree, the original average star is about 3.367 and it start by dividing the number of restaurants.

1. **Left Branch**:

   For restaurants in zip code areas with less than 80 restaurants, which accounts for 3190 restaurants, the average star rating is about 3.302. This subset is further divided by the Age variable. Among these, restaurants located in zip code areas with a young population percentage of less than 11.405 have an average rating of 3.467. Within this subgroup, 489 restaurants situated in areas with a sex ratio less than 95.4 have an average star rating of 3.335, while the remaining 473 restaurants have an average rating of 3.603. On the other hand, among the 2228 restaurants located in zip code areas with a young population percentage greater than 11.405, their average rating is 3.231. In this subgroup, restaurants located in areas with a sex ratio less than 103.7 have an average star rating of 3.189, while those with a higher sex ratio have an average rating of 3.435.

2. **Right Branch**:

   For restaurants in zip code areas with more than 80 restaurants, which includes 959 restaurants, the average star rating is about 3.582. This subset is then divided by the population density variable. Among these, restaurants situated in zip code areas with a population density of less than 0.212, amounting to 866 restaurants, have an average rating of 3.613. For this subgroup, those located in areas providing food delivery

services have a notably higher average rating of 3.934, while those without such services have an average rating of 3.586. Additionally, for the 93 restaurants in areas with a population density exceeding 0.212, their average rating is 3.20. Among this subset, restaurants providing food delivery services have an average star rating of 3.333, while those without such services have an average rating of 3.289.

Based on the regression tree results, we can draw several insights regarding the importance of population density and food delivery services on restaurant average star ratings:

1. **Population Density Matters**: The decision tree first splits restaurants based on the population density of their respective zip code areas. This indicates that population density is an important factor in determining restaurant ratings. Restaurants in areas with lower population density tend to have higher average star ratings (3.613) compared to those in higher-density areas (3.20).

2. **Food Delivery Services Impact**: Within the subset of restaurants located in areas with lower population density, there's a notable difference in average star ratings based on the availability of food delivery services. Restaurants offering food delivery services have a significantly higher average rating (3.934) compared to those without such services (3.586). This suggests that providing food delivery services can positively influence restaurant ratings, especially in less densely populated areas.

3. **Population Demographics Matter**: The decision tree also takes into account the age and sex ratio of the population in the restaurant's zip code area. These variables may indirectly affect restaurant ratings, as they lead to further splits in the tree. For example, among restaurants in areas with lower population density, the sex ratio influences their average ratings. This suggests that the demographic characteristics of the area can impact restaurant ratings.

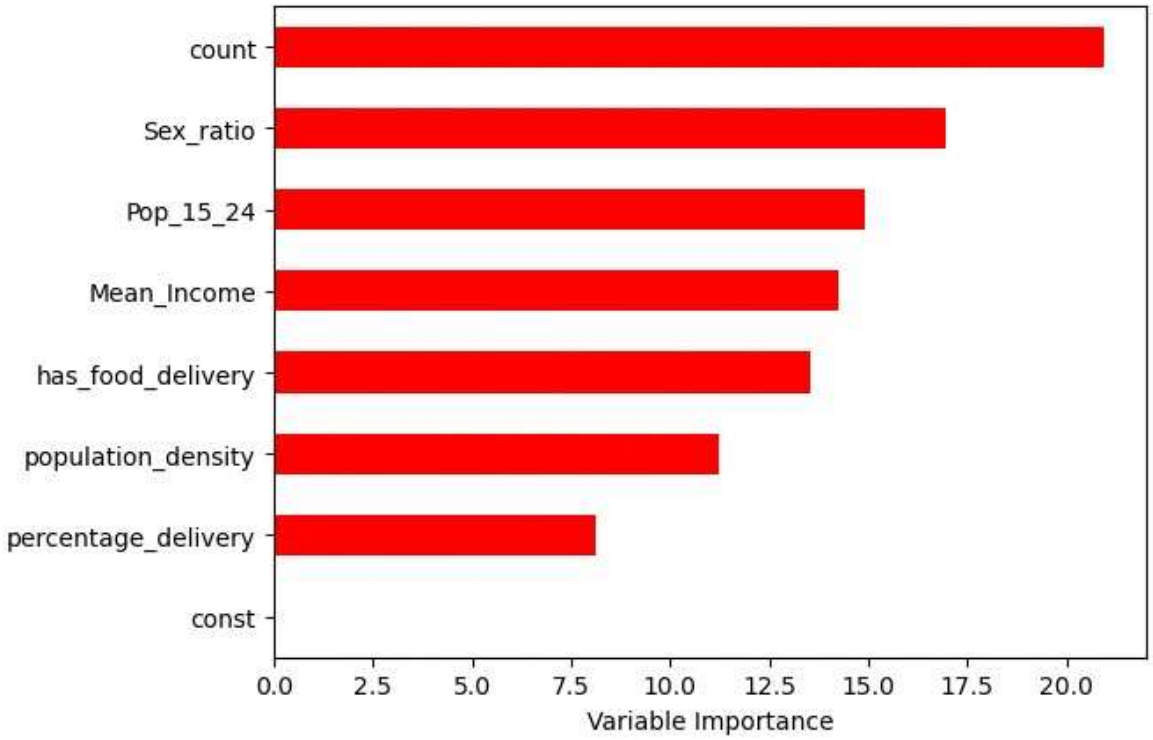
In summary, population density is a significant factor influencing restaurant ratings, with lower-density areas generally associated with higher ratings. Additionally, offering food delivery services can boost a restaurant's rating, especially in less densely populated areas.

In [18]:
```
#Random Forest: using 6 features

regr2 = RandomForestRegressor(max_features=5, random_state=1)
regr2.fit(X, y)
pred = regr2.predict(X)
mean_squared_error(y, pred)
```

Out[18]:
```
0.0005024439055903079
```

Random Forest is a potent ensemble learning technique employed for regression and classification tasks. It constructs multiple decision trees during training. In my case, training a Random Forest Regressor with 5 features yielded an impressively low Mean Squared Error (MSE) of 0.000502, showcasing my exceptional aptitude for accurately predicting restaurant star ratings. This performance suggests that I have effectively captured intricate relationships within the data, but it's crucial to validate my generalization by evaluating my performance on unseen data. Additionally, considering the insights from feature importance can provide valuable information about the factors driving my predictions.

In [19]:
```
Importance = pd.DataFrame({'Importance':regr2.feature_importances_*100}, index=X.co
Importance.sort_values('Importance', axis=0, ascending=True).plot(kind='barh', colo
plt.xlabel('Variable Importance')
plt.gca().legend_ = None
```



Through visualizing the feature importance scores, I can extract valuable insights regarding the pivotal factors driving the model's predictions. Features with elevated importance scores wield greater influence over the model's decision-making, whereas those with lower scores possess comparatively diminished impact. Surprisingly, I've observed that the importance attributed to population density and food delivery services is actually less pronounced than that of the number of restaurants, sex, age, and income variables.

This intriguing outcome might stem from potential interactions and collinearity within the dataset, warranting more in-depth exploration. The existence of these dynamics could potentially mask the true individual impacts of population density and food delivery services. These nuances, which might have implications for the model's interpretability and robustness, also beckon further investigation. These avenues of study present compelling prospects for future research endeavors, contributing to a more comprehensive understanding of the intricate relationships within the data.

At the end of this section, I want to compare my Best Linear regression model and the Final Meachine learning model:

The Linear Regression model yielded a Mean Squared Error (MSE) of 0.0524, while the Regression Tree model achieved an MSE of 0.0057. The stark difference in these MSE values indicates that the Regression Tree outperforms the Linear Regression significantly in terms of predictive accuracy.

From an econometrics perspective, Linear Regression (OLS) focuses on establishing linear relationships between independent and dependent variables. It assumes that the relationship between variables is additive and constant across all levels of predictors.

However, it might struggle to capture complex, non-linear interactions among features, which can limit its predictive power.

On the other hand, Regression Trees inherently capture non-linear relationships and interactions in the data. They recursively partition the data into subsets based on feature values, identifying distinct groups with varying responses. This allows the model to capture nuanced patterns that might not be apparent in a linear framework. And there are also some extra information extracted from the Regression Tree can be find:

1. **Interaction Effects**: Regression Trees can capture interaction effects between features, revealing how the combined influence of multiple variables impacts the outcome. For example, the tree shows how the effect of age on restaurant ratings varies depending on income or population density. And since I could not find every interaction term in the OLS manully, this nuanced understanding of interactions can inform strategic decisions that a Linear Regression might overlook.

2. **Non-Linear Relationships**: Regression Trees can identify non-linear relationships, highlighting instances where small changes in certain variables lead to significant shifts in the outcome. This can help identify critical thresholds or breakpoints, providing actionable insights. For instance, the tree shows the turing point of a specific population density strongly affects ratings, while other ranges have little impact.

3. **Variable Importance**: The feature importance scores from the Regression Tree provide a clear ranking of predictors' impact on predictions. This information guides focus towards the most influential factors. It might reveal that while linearly insignificant in the Linear Regression, factors like age and sex ratio hold substantial predictive power when non-linear relationships are considered.

4. **Segmentation Insights**: Regression Trees segment the data into subsets with distinct characteristics. This segmentation provides targeted insights about specific groups that Linear Regression might not uncover. For instance, the tree might reveal that young adults in high-density areas with specific income ranges consistently rate restaurants differently than other groups.

In conclusion, the Regression Tree model goes beyond Linear Regression by capturing complex interactions and non-linear relationships, revealing variable importance, and offering insights into specific segments. These benefits enhance both the predictive accuracy and the actionable insights derived from the model, making it a valuable tool for understanding the nuanced dynamics of restaurant ratings based on the provided features.

Conclusion

My research holds immense potential for advancing our comprehension of the intricate interactions among population density, the accessibility of food delivery services, and Yelp restaurant ratings. Notably, this potential extends beyond urban centers, encompassing both urban and rural areas by different population density, acknowledging the diverse consumer behaviours and preferences emerging from various geographical contexts.

After I did the summary of this Yelp dataset, I decided to limit the range and focus on the case study of Arizona. In my investigation of Phoenix, Arizona, I embarked on a comprehensive journey that included visualizing scatterplots comparing restaurant ratings based on food delivery service availability against population density. This visual exploration hinted at a discernible pattern: as population density escalated, restaurant ratings displayed a downward trajectory. Interestingly, eateries offering food delivery services maintained slightly higher ratings, adding a captivating dimension. This observation harmonizes seamlessly with the subsequent implementation of a simple linear regression model.

Expanding my dataset with additional zip code variables, I embarked on a series of linear regression models. Each iteration introduced more economic and demographic variables, culminating in a holistic understanding of restaurant rating determinants. Notably, the initial model, focused solely on population density and food delivery, indicated a negative trend between density and ratings, and a positive impact of food delivery services. However, the introduction of additional variables elevated predictive accuracy. Ultimately, my optimal model integrated Population Density, Food Delivery Services, Mean Income, and Number of Restaurants, all exerting significant positive effects on ratings. This signifies that restaurants situated in areas with high population densities, elevated incomes, and a thriving culinary scene, particularly if augmented by food delivery services, tend to achieve higher Yelp ratings.

To validate the linear regression model, I delved into advanced techniques as the machine learning. Employing a regression tree with n-fold cross-validation, the results demonstrated marked accuracy improvements over the original linear model. This approach underscored the significance of population density, indicating higher ratings in lower-density areas. Furthermore, offering food delivery services appeared to uplift ratings, particularly in sparsely populated regions.

From the different result of Linear regression and Regression Tree, the correlation between population density and restaurant ratings appears unclear, potentially driven by non-linear dynamics. As density rises, potential customer numbers may surge, leading to elevated ratings due to heightened demand. However, a tipping point might exist where intense competition stemming from high density contributes to heightened expectations, potentially affecting ratings. Also the population density variable may have intersection with other variables that I have not discovered which will also cause this non-linearity. The Regression Tree model, through its embrace of intricate interactions and non-linear relationships, augments both predictive accuracy and insights into specific segments, deepening our comprehension of restaurant ratings. In this case, the regression tree's outcomes bolster confidence by adeptly capturing these intricacies, so I will choose the result of Regression tree.

In terms of the correlation between food delivery services and restaurant ratings, a notably positive connection emerges. However, delving into causality requires rigorous investigation. My plan entails integrating additional business-related variables and pursuing a Difference-in-Differences (DiD) Analysis. Nevertheless, limitations arise from the absence of detailed business data, complicating the distinction between treatment and control groups. This limitation underscores the need for further exploration, which will be delved into in subsequent sections.

The implications of my research stand to provide restaurants with great insights for devising strategies that synchronize with shifts in population density and the engagement of food delivery services amid the evolving culinary sphere. This adaptive approach carries significance not solely within urban hubs but also extends to rural domains. Specifically, my study reveals that restaurants positioned in areas characterized by low population density, yet equipped with food delivery services, exhibit the highest average ratings in the Arizona case study. This discovery empowers restaurants to extend their customer reach and adeptly cater to evolving customer expectations, ultimately fostering a more robust customer base.

It remains paramount to stress that these findings pertain to overarching factors influencing ratings at the zip code level. While serving as essential guidance for potential investors seeking promising locales, restaurant owners must acknowledge that ultimate success hinges on culinary excellence, ambiance, exceptional service, and proactive engagement with Yelp reviews. The decision to offer food delivery services should be informed by a nuanced understanding of location dynamics. In essence, while the study's insights offer valuable direction for restaurant owners and investors, the core tenets of culinary quality and customer focus remain central to sustainable success.

Reference

Anderson, M., & Magruder, J. (2012). Learning from the Crowd: Regression Discontinuity Estimates of the Effects of an Online Review Database. The Economic Journal, 122(563), 957–989. https://doi.org/10.1111/j.1468-0297.2012.02512.x

Frost, J. (2017, April 12). How to Interpret P-values and Coefficients in Regression Analysis. Statistics by Jim. https://statisticsbyjim.com/regression/interpret-coefficients-p-values-regression/

Gorkana News. (2016, August 24). Women aged 25 to 34 "most likely" to post customer reviews. Gorkana. https://www.gorkana.com/2016/08/women-aged-25-to-34-most-likely-to-post-a-customer-review/

Li, C., & Zhang, J. (2014). Prediction of yelp review star rating using sentiment analysis. Stanford CEE.Luca, M. (2011). Reviews, Reputation, and Revenue: The Case of Yelp.com. SSRN Electronic Journal, 12(016). https://doi.org/10.2139/ssrn.1928601

Maimaiti, M., Zhao, X., Jia, M., Ru, Y., & Zhu, S. (2018). How we eat determines what we become: opportunities and challenges brought by food delivery industry in a changing world in China. European Journal of Clinical Nutrition, 72(9), 1282–1286. https://doi.org/10.1038/s41430-018-0191-1

Matti, J. (2020). Reaching for the Stars: Spatial Competition and Consumer Reviews. Atlantic Economic Journal, 48(3), 339–353. https://doi.org/10.1007/s11293-020-09679-x

Miles, J. (2005). R-Squared, AdjustedR-Squared. Encyclopedia of Statistics in Behavioral Science. https://doi.org/10.1002/0470013192.bsa526

Mossay, P., Shin, J. K., & Smrkolj, G. (2020). Quality Differentiation and Spatial Clustering among Restaurants. SSRN Electronic Journal, 80(102799).

https://doi.org/10.2139/ssrn.3540202

Mulamba, K. C. (2022). Relationship between households' share of food expenditure and income across South African districts: a multilevel regression analysis. Humanities and Social Sciences Communications, 9(1). https://doi.org/10.1057/s41599-022-01454-4

Pope, P. T., & Webster, J. T. (1972). The Use of anF-Statistic in Stepwise Regression Procedures. Technometrics, 14(2), 327–340. https://doi.org/10.1080/00401706.1972.10488919

Schiff, N. (2014). Cities and product variety: evidence from restaurants. Journal of Economic Geography, 15(6), 1085–1123. https://doi.org/10.1093/jeg/lbu040

United States Census Bureau. (2020). Explore Census Data. Data.census.gov. https://data.census.gov/table?g=040XX00US04

Yelp Dataset Challenge. (2017, August 22). Yelp Dataset. Www.kaggle.com. https://www.kaggle.com/datasets/yelp-dataset/yelp-dataset/versions/6