Final Project Coding

April 16, 2023

1 Project #1:

What Determines Business' Yelp Ratings in Toronto?
by Thomas Keough

1.1 Introduction

With broad access to review data on websites, online business ratings are becoming increasingly relevant to business owners. This is especially true of service-oriented businesses, such as restaurants, hair salons, and hotels. Customers prefer to visit well-reviewed businesses, which means businesses should care about their online ratings. By extension, this means that both businesses and customers are interested in identifying what causes a business to earn good reviews. This analysis seeks to understand the determinants of online business ratings by using metrics available on Yelp, one of the most popular websites for reviews.

This analysis using data acquired from Kaggle.com. The data was initially published by Yelp as part of a dataset challenge. It was last updated in August 2017 and includes data from 11 cities in four countries, including the USA and Canada. Additional Yelp data on prices is gathered using the Yelp Fusion API. Population data is acquired from Statistics Canada. The dataset is composed of files pertaining to business attributes, hours of operation, reviews, check-ins, tips, and users. This analysis merges the files for business attributes, hours of operation, check-ins, prices, and population for businesses in Toronto, Ontario. The dependent variable is business ratings, measured in Yelp stars. The chosen covariates are business' total number of Yelp reviews, total weekly hours of operation, daily average number of Yelp check-ins, business prices as measured by the number of dollar signs on Yelp, and population by forward sortation area (FSA).

The results of this analysis demonstrate that the number of reviews, hours of operation, and number of check-ins are all positively correlated with business ratings. The effect of the number of reviews depends in part on the type of business. By splitting the observations into three categories (shops, restaurants, other), restaurants have the lowest average rating while shops have the highest. Neither business price levels nor the population of a business's FSA appears to be correlated with businesses ratings. In order to determine the strength and magnitude of these correlations, a multiple regression analysis should be conducted.

Many studies have been conducted using Yelp data thanks to it being easily accessible through Yelp's Fusion API. Luca (2016) found that a one-star increase in Yelp ratings generates between five and nine percent higher revenues for restaurants in Seattle. Luca utilized a regression discontinuity design to infer causality with this relationship. The pursuit of rating maximization is therefore worthwhile for restaurants (at least). However, Luca's study does not draw conclusions about what

a business can do to increase its Yelp rating. Given that higher ratings create more revenue for restaurants, this study's objective stands to be very relevant for businesses in the internet age.

Other research has been able to identify what factors are negatively correlated with Yelp ratings. Byers, Mitzenmacher, and Zervas (2012) find that businesses who provide Groupon promotions suffer from a significant decline in Yelp ratings on average. They suggest that Groupon customers are often treated more poorly relative to others. Naturally, their result contributes to identifying the determinants of Yelp ratings since it illustrates that poor service is punished with negative Yelp reviews. Conversely, it is reasonable to assume that good service is to be rewarded with positive reviews. Though this is self-evident, it is an important foundation on which the intitution for rating predictors can be built. With this in mind, this paper seeks to measure "good service" through business accessible through Yelp. Vinson, Dale, and Jones (2019) discover that reviewers tend to systematically bias their present reviews away from their previous reviews. Though their study is focused on an application to human cognition, it still reveals that holding the reviwer constant, Yelp reviews are in part determined by previous reviews. This demonstrates that Yelp ratings are not necessarily an objective measure of business quality. Therefore, there must factors that contribute to biases in reviews beyond the dichotomy of good versus bad service. These factors may be measurable through variables that account for customer access (i.e. weekly hours of operation) or social popularity (i.e. average number of check-ins). Thus, this paper will partially account for reviewer bias in analyzing predictors of Yelp ratings.

1.2 Data Cleaning

(i) Setting Up The Project

Y: Business Rating ('stars')
This analysis seeks to identify the determinants of business ratings using quantitative data gathered from Yelp. The outcome variable is the business rating, which is referred to as "stars" in the dataset. This variable takes a value between 1 and 5. It is discrete data; potential values must be multiples of 0.5. This analysis includes three covariates of interest.

X1: Number of Reviews ('review_count')
The first covariate is the total number of reviews for each business. It is discrete. The review system is an extremely prominent feature of Yelp and is used frequently, so the number of reviews should account for a significant proportion of the variation in business ratings. There are two valid hypotheses for the correlation between the number of reviews with business ratings. For instance, a business that has exceptionally good service can encourage Yelp users to leave positive reviews. This would suggest that the number of reviews is positively associated with business ratings. However, the opposite effect is also feasible; exceptionally poor service may encourage Yelp users to leave negative reviews, meaning the number of reviews would be negatively correlated with business ratings. The dominant hypothesis, should one exist, will manifest itself through an empirical analysis of the relationship between review count and ratings.

X2: Number of Operating Hours per Week ('wk_op_hours')
The second covariate is the number of operating hours in a week for a business. Businesses differ largely in hours of operation on a day-to-day basis, so using the weekly total of operating hours will account for those differences across the days of the week. This variable will provide insight on the effect of customer access to businesses, which may be a component in evaluating business quality from the perspective of customers. There are several hypotheses for the direction of the relationship between operating hours and business ratings. It could be that businesses that are open longer

throughout the week earn more reviews because they can serve more customers. Therefore, the direction of the relationship would be dependent on the ambiguous effect of the number of reviews on business ratings. Alternatively, businesses that are open for fewer hours per week may instill a sense of exclusivity in its customers. For example, fine dining venues, nightclubs, and other businesses that have limited weekly operating hours may observe more positive reviews because attendees feel exclusive.

X3: Average Number of Check-ins per Day ('daily_checkin_avg')
The third covariate is the average number of Yelp check-ins per day. It is important to note that this calculation of the daily check-in average ignores days wherein businesses had zero check-ins. In other words, it is the average number of check-ins for days that had at least one check-in. "Checking in" is a Yelp feature that allows users to inform their Yelp following of the businesses they visit. When a user "checks in" to a business, their attendance is published on their profile. Users can also earn badges and special offers at businesses they check in at. These two components provide increased incentives for Yelp users to visit high-quality businesses; users can show off their attendance at a popular business and potentially earn discounts in the process. Comparatively, poor-quality businesses would have fewer check-ins since users would not want to publish their attendance at them. Therefore, this variable accounts for customer perception of businesses. Therefore, check-ins should be positively correlated with a business's ratings in theory. An OLS regression that utilizes this covariate and the number of reviews would provide insight on how the number of reviews affects a business's rating while holding constant the customers' perception of the business. This would provide insight on the causality of business ratings.

(ii) Data Cleaning

[ ]:
```
import pandas as pd
import matplotlib.pyplot as plt
import geopandas as gpd
from shapely.geometry import Point
import numpy as np
import json
import statsmodels.api as sm
from stargazer.stargazer import Stargazer
from IPython.core.display import HTML
import sklearn
%matplotlib inline
```

[ ]:
```
# load data
df = pd.read_csv('/Users/thomas/Documents/schoolwork/eco225/yelp_data/
 ↪yelp_business.csv')
df2 = pd.read_csv('/Users/thomas/Documents/schoolwork/eco225/yelp_data/
 ↪yelp_checkin.csv')
df3 = pd.read_csv('/Users/thomas/Documents/schoolwork/eco225/yelp_data/
 ↪yelp_business_hours.csv')

# make copies
bsn_df = df.copy()
checkin_df = df2.copy()
```

```
hours_df = df3.copy()
```

[ ]:
```
# build Toronto businesses dataframe:

# make a df for Toronto businesses
tor_bsn = bsn_df.loc[bsn_df['city'] == 'Toronto']

# generate a restaurant dummy
tor_bsn['restaurant'] = tor_bsn['categories'].str.contains('Restaurant').
 ↪astype(int)

# generate a shops dummy
tor_bsn['shop'] = tor_bsn['categories'].str.contains('Shopping').astype(int)

# Incorporate check-in data:

# find average daily checkins for each business
all_checks = checkin_df.groupby('business_id').mean().
 ↪rename(columns={'checkins': 'daily_checkin_avg'})

# inner join on tor_bsn to get checkin data for each business in Toronto
tor_bsn = tor_bsn.merge(all_checks, on='business_id')

# === Incorporate weekly operating hours: ===

# build a function to clean hours_df

def count_hours(operating_hours: str) -> float:
    """Returns a float given a string of the form "HH:MM-HH:MM" that contains a␣
 ↪business's operating hours."""

    # if business is closed on the given day:
    if operating_hours == 'None':
        return 0.0

    # '2000-01-01' is needed in the string that is converted to datetime to␣
 ↪avoid being out of pandas' accepted date range
    hours = operating_hours.split('-')
    opn = pd.to_datetime('2000-01-01 ' + hours[0])
    close = pd.to_datetime('2000-01-01 ' + hours[1])

    # if the business is open past midnight:
    if close < opn:
        end_date = '2000-01-02'
    else:
        end_date = '2000-01-01'
```

```
    # calculate a timedelta that accounts for hours of operation past midnight
    window = pd.to_timedelta(pd.to_datetime(f'{end_date} ' + hours[1]) - pd.
 ↪to_datetime('2000-01-01 ' + hours[0]))

    # return a float reprsenting the number of hours open in the given day
    return window.total_seconds() / (60 * 60)

# merge hours_df with tor_bsn to avoid running count_hours on unneeded␣
 ↪businesses
tor_bsn = tor_bsn.merge(hours_df, on='business_id')

# generate total weekly operating hours for each Toronto restaurant
tor_bsn['wk_op_hours'] = 0

# sum operating hours for each day - monday thru sunday are columns 8-14
for day in tor_bsn.columns[-8:-1]:
    tor_bsn['wk_op_hours'] += tor_bsn[day].apply(count_hours)

# remove daily operating hours columns
tor_bsn = tor_bsn.drop(columns=['monday', 'tuesday', 'wednesday', 'thursday',␣
 ↪'friday', 'saturday', 'sunday'])

# save cleaned data to csv
# tor_bsn.to_csv('/Users/thomas/Documents/schoolwork/eco225/yelp_data/
 ↪toronto_businesses.csv')

# order columns
tor_bsn = tor_bsn[['business_id', 'name', 'stars', 'review_count',␣
 ↪'daily_checkin_avg', 'wk_op_hours', 'restaurant', 'shop', 'categories']]

# Toronto dataframe is now clean
tor_bsn
```

```
/var/folders/ms/q_hrvhnj60188mzvbg992ch80000gn/T/ipykernel_1477/4038026341.py:7:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  tor_bsn['restaurant'] =
tor_bsn['categories'].str.contains('Restaurant').astype(int)
/var/folders/ms/q_hrvhnj60188mzvbg992ch80000gn/T/ipykernel_1477/4038026341.py:10
: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  tor_bsn['shop'] = tor_bsn['categories'].str.contains('Shopping').astype(int)
```

[ ]:

	business_id	name	stars	\
0	l09JfMeQ6ynYs5MCJtrcmQ	"Alize Catering"	3.0	
1	lHYiCS-y8AFjUitv6MGpxg	"Starbucks"	4.0	
2	VSGcuYDV3q-AAZ9ZPq4fBQ	"Sportster's"	2.5	
3	1K4qrnfyzKzGgJPBEcJaNQ	"Chula Taberna Mexicana"	3.5	
4	AtdXq_gu9NTE5rx4ct_dGg	"DAVIDsTEA"	4.0	
…	…	…	…	
14845	sEAKw3MZkER1u_1fzIeD3g	"Gol Take-Out"	4.0	
14846	1HplwLVbBid-Bgw1sEPGFg	"Dumpling Melody Bistro"	2.0	
14847	dWoAayHRyIrkk1dcvBxv3Q	"Art Ink Collective"	3.5	
14848	SvW3WsatQWvR8c1iwAD_QA	"Urban House Cafe"	4.0	
14849	nGjEV4bn0DPk8bcb0C6Aig	"Sweet Serendipity Bake Shop"	4.5	

	review_count	daily_checkin_avg	wk_op_hours	restaurant	shop	\
0	12	1.000000	91.0	1	0	
1	21	4.577586	115.5	0	0	
2	7	2.125000	70.0	0	0	
3	39	1.333333	103.5	1	0	
4	6	1.272727	70.5	0	0	
…	…	…	…	…	…	
14845	15	1.000000	0.0	1	0	
14846	12	1.125000	0.0	1	0	
14847	3	1.000000	43.0	0	1	
14848	32	2.000000	91.0	1	0	
14849	22	1.000000	45.0	0	0	

	categories
0	Italian;French;Restaurants
1	Food;Coffee & Tea
2	Bars;Sports Bars;Nightlife
3	Tiki Bars;Nightlife;Mexican;Restaurants;Bars
4	Coffee & Tea;Food;Tea Rooms
…	…
14845	Food;Restaurants;International Grocery;Ethnic …
14846	Restaurants;Chinese
14847	Shopping;Beauty & Spas;Piercing;Art Galleries;…
14848	Nightlife;Restaurants;Sandwiches;Bars;Canadian…
14849	Bakeries;Food

[14850 rows x 9 columns]

1.3 Summary Statistics

[ ]:
```
# generate summary statistics for y, x1, x2, x3
sums = tor_bsn[['stars', 'review_count', 'wk_op_hours', 'daily_checkin_avg']].
 ↪describe()
sums = sums.rename(columns={'stars': 'Yelp Rating', 'review_count': 'Review␣
 ↪Count', 'wk_op_hours': 'Hours Open per Week', 'daily_checkin_avg': 'Mean␣
 ↪Daily Check-ins'})
sums
```

[ ]:

	Yelp Rating	Review Count	Hours Open per Week	Mean Daily Check-ins
count	14850.000000	14850.000000	14850.000000	14850.000000
mean	3.494007	28.295556	47.315771	1.522870
std	0.841782	56.253223	35.878516	1.090018
min	1.000000	3.000000	0.000000	1.000000
25%	3.000000	5.000000	0.000000	1.000000
50%	3.500000	10.000000	54.000000	1.187500
75%	4.000000	28.000000	74.500000	1.555556
max	5.000000	1494.000000	167.883333	31.042553

Stars

The summary statistics for *stars* indicate that the data is discrete. Moreover, it demonstrates that the minimum and maximum ratings are 1 and 5 respectively. The average review is roughly 3.5 stars. Assuming that an average business should have a rating near the midpoint between the minimum and maxmimum, the mean may indicate that there is a systematic bias towards over-rating businesses by 0.5 stars at the aggregate level. The interquartile range is between 3 and 4 stars, which further suggests this bias; it demonstrates that half of all businesses in Toronto are above the rating midpoint. In other words, only 25% of business can be considered poor-quality (i.e. a rating below 3 stars).

Number of Reviews

There appears to be a great variation in the number of reviews across businesses but it is primarily due to an enormous right skew. The standard deviation (56) is double the mean review count (28). The maximum review count of 1494 is indicative of right skew since it is 53 times larger than the average. Moreover, 75% of all Toronto businesses have 28 reviews or less. This may yield low power in a regression analysis since the vast majority of businesses have very similar review counts, despite the extreme variation suggested by the standard deviation.

Weekly Operating Hours

On average, businesses in Toronto are open for roughly 47 hours per week. The standard deviation is roughly 36 hours per week which indicates that there is a large variation in weekly operating hours across businesses. There are several signs of right skew in this variable, which may be the cause of the large standard deviation. The 50th percentile (54) is larger than the mean, which suggests that outliers on the right side of the distribution are positively biasing the mean. Moreover, this variable's lower bound of zero suggests that most of the variation would occur above the mean. The maximum value observed for this variable is 167.88, which is 0.12 hours less than the total number of hours in a week. This observation may need to be dropped from analysis if it biases the trends of interest, since the vast majority of businesses cannot operate for nearly every hour of the week.

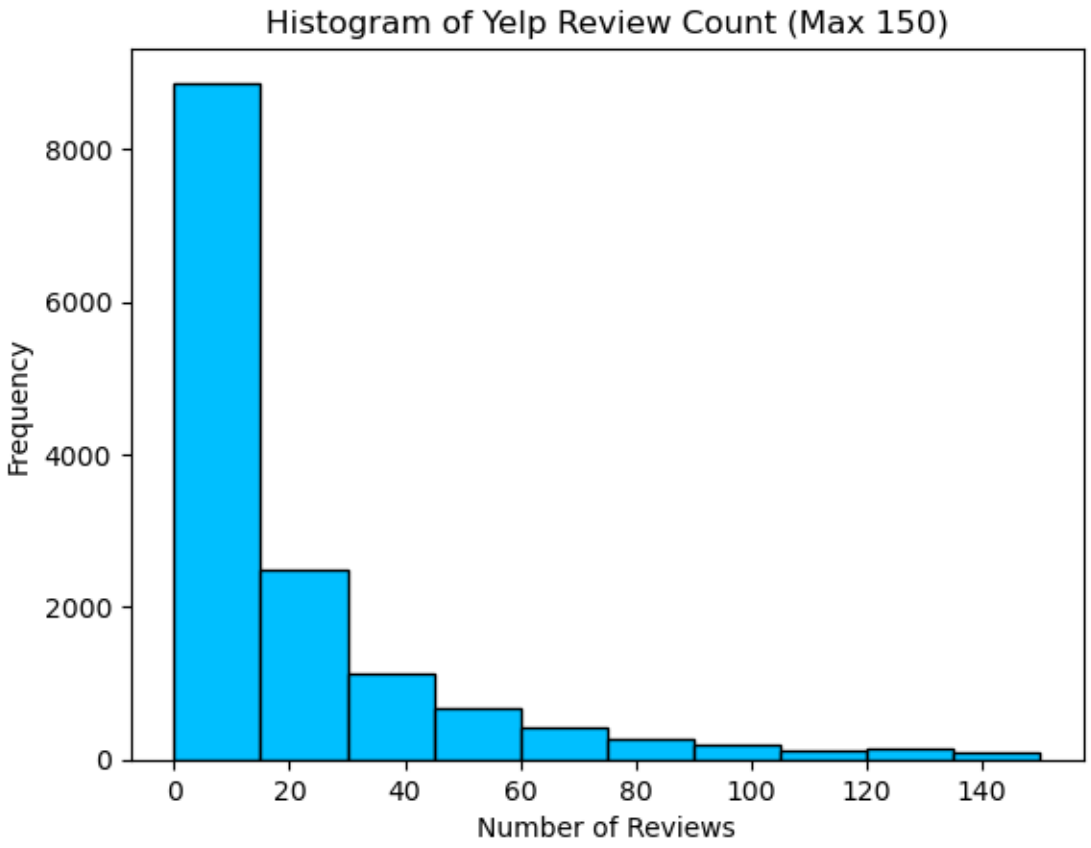
Daily Check-in Average

There is very little variation across businesses in their daily check-in averages. Despite the mean and standard deviations being approximately 1.52 and 1.10 respectively, 75% of the observations lie between 1 and 1.55. There is a significant right skew in this data since the maximum value observed is roughly 31 check-ins on average. This variable may prove irrelevant in a regression analysis due to the uniformity of the data: there is not enough variation to properly calculate how an outcome variable changes given a change in the daily check-in average.

1.4 Plots & Figures

[ ]:
```
# histograms of review_count
fig, ax = plt.subplots()

tor_bsn['review_count'].plot(ax=ax, kind='hist', color='deepskyblue',␣
↪edgecolor='black', grid=False, title='Histogram of Yelp Review Count (Max␣
↪150)', range=(0,150))
plt.xlabel('Number of Reviews')
```

[ ]:
```
Text(0.5, 0, 'Number of Reviews')
```



As demonstrated by the summary statistics table, review_count has significant right skew. Ignoring businesses with more than 150 reviews yields a clearer picture: the distribution of reviews is unimodal with a peak between 0 and ~15. Given the lack of variation, the number of reviews may not be a significant determinant of business ratings. In other words, the variation in business ratings cannot be meaningfully explained by the number of reviews alone because the majority of businesses are very similar in this dimension. This variable must be used in a multiple regression alongside additional covariates to generate statistically significant coefficients and identify the relevant determinants of business ratings.

[ ]:
```
# relationship between business rating and review count
fig, ax = plt.subplots()

tor_bsn.plot(ax=ax, kind='scatter', c='deepskyblue', edgecolor='black',␣
 ↪x='review_count', y='stars', xlim=(0,700), title='Number of Reviews &␣
 ↪Business Rating, Without Outliers', linewidth=0.5)
plt.xlabel('Review Count')
plt.ylabel('Yelp Rating')
```

```
/Users/thomas/opt/anaconda3/lib/python3.9/site-
packages/pandas/plotting/_matplotlib/core.py:1114: UserWarning: No data for
colormapping provided via 'c'. Parameters 'cmap' will be ignored
  scatter = ax.scatter(
```

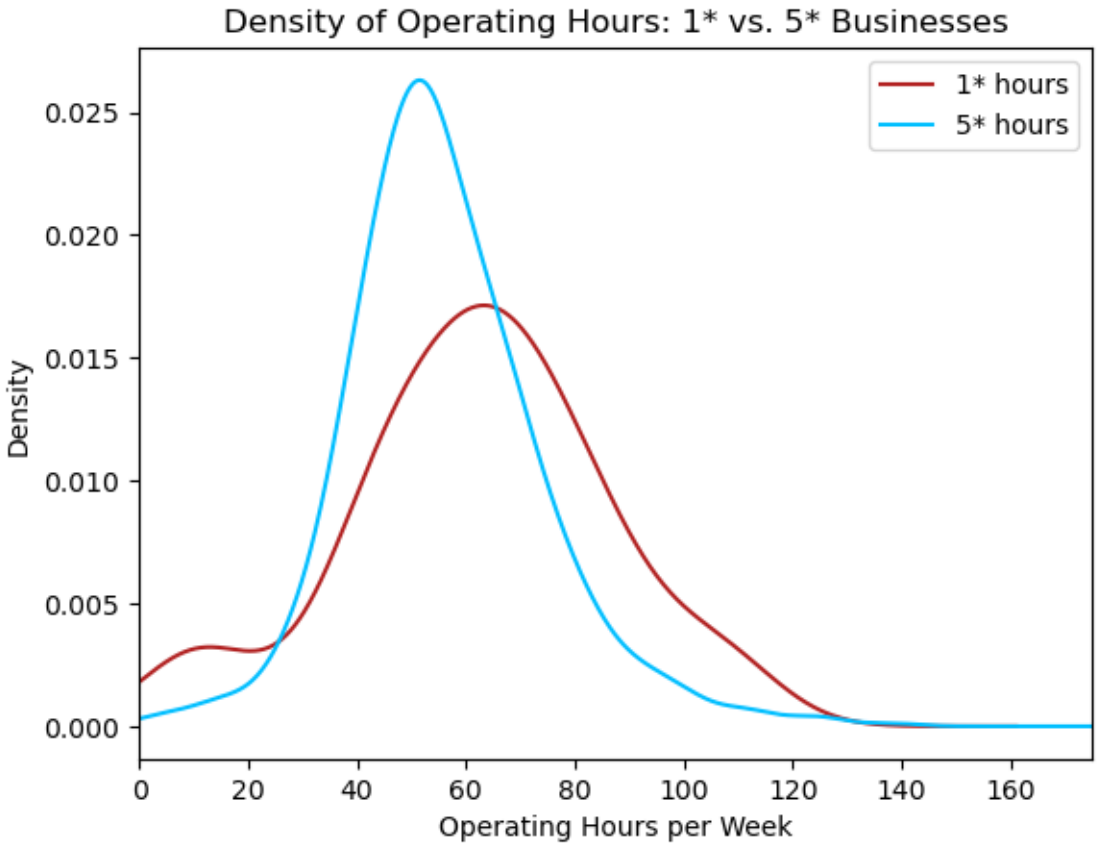
[ ]:
```
Text(0, 0.5, 'Yelp Rating')
```

This scatter plot demonstrates that a business's number of reviews is positively correlated with the business's rating. This suggests that the effect of good service dominates the effect of bad service in regard to generating reviews for businesses. If the poorest quality businesses had the greatest number of reviews, then the effect of bad service would dominate. There appears to be non-linearity in this relatationship: the marginal benefit of an additional review decreases significantly between the 200 and 300 review marks. However, this non-linearity may simply be a result of the data for *stars* being discrete. The true effect of the number of reviews on a business's rating is not clear through this plot. Controlling for additional variables, such as the number of checkins on average per day (i.e. customer perception), may present a relationship that differs from the one seen here.

[ ]:
```
# density of operating hours for businesses with 1*, 5* reviews
tor_1s = tor_bsn.groupby('stars').get_group(1).rename(columns={'wk_op_hours':␣
 ↪'1* hours'})
tor_1s = tor_1s[tor_1s['1* hours'] != 0]
tor_5s = tor_bsn.groupby('stars').get_group(5).rename(columns={'wk_op_hours':␣
 ↪'5* hours'})
tor_5s = tor_5s[tor_5s['5* hours'] != 0]
```

```
tor_1s['1* hours'].plot(kind='kde', c='firebrick', legend=True, title='Density␣
 ↪of Operating Hours: 1* vs. 5* Businesses', xlim=(0,175)).
 ↪set_xlabel("Operating Hours per Week")
tor_5s['5* hours'].plot(kind='kde', c='deepskyblue', legend=True)
```

[ ]: <AxesSubplot: title={'center': 'Density of Operating Hours: 1* vs. 5* Businesses'}, xlabel='Operating Hours per Week', ylabel='Density'>



This density plot demonstrates how weekly operating hours differ across businesses at either end of the rating spectrum. Five-star businesses are more likely to have between 45-65 hours of operation in any given week and a narrower spread than one-star businesses. On the contrary, one-star businesses have much more variation and are more likely to have less than 30 hours of operation. This evidences the hypothesis that customer access plays a role in determining business ratings. This density plot ignores businesses that had no weekly operating hours listed on Yelp in order to portray a clearer picture of the spread of operating hours across ratings.

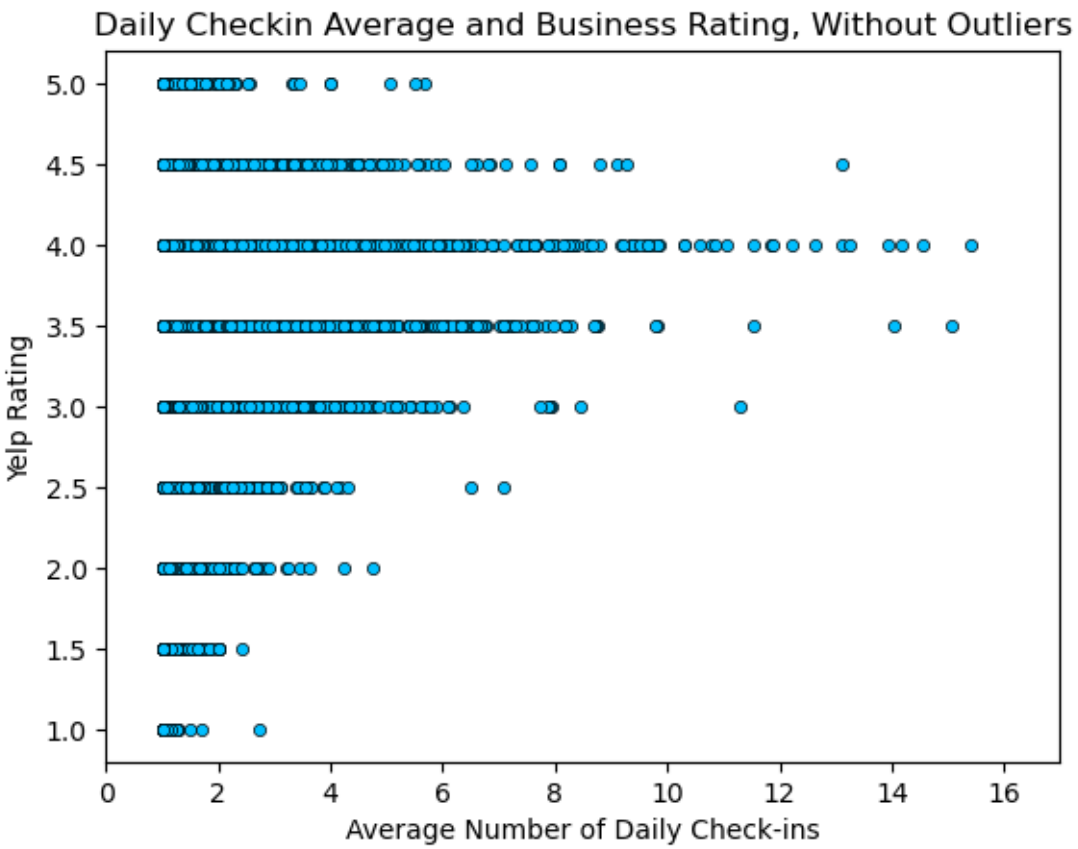
The wider variation in operating hours per week for one-star businesses suggests that there are more one-star businesses than five-star businesses that operate for more than 70 hours per week. This may be random noise, but it could indicate that businesses' hours of operation are weakly correlated with their ratings. In other words, hours of operation may be a poor determinant of business ratings. In order to identify the strength of a correlation between operating hours and

ratings, a linear regression should be done with these two variables.

[ ]:
```
# relationship between businsess rating and checkins
fig, ax = plt.subplots()

tor_bsn.plot(ax=ax, kind='scatter', edgecolor='black', x='daily_checkin_avg',␣
 ↪y='stars', c='deepskyblue', title='Daily Checkin Average and Business␣
 ↪Rating, Without Outliers', linewidth=0.5).set_xlim(0, 17)
plt.xlabel('Average Number of Daily Check-ins')
plt.ylabel('Yelp Rating')
```

```
/Users/thomas/opt/anaconda3/lib/python3.9/site-
packages/pandas/plotting/_matplotlib/core.py:1114: UserWarning: No data for
colormapping provided via 'c'. Parameters 'cmap' will be ignored
  scatter = ax.scatter(
```

[ ]:
```
Text(0, 0.5, 'Yelp Rating')
```



There is a clear positive association between the average number of check-ins per day and the business rating. Non-linearity also appears to be present but this may be a result of business ratings being measured discretely. There are several massive outliers beyond roughly 17 average

check-ins per day, which should be excluded from a formal regression analysis as they exacerbate the non-linear trend. Overall, this relationship is very similar to that of the total number of reviews and business ratings. As a result, controlling for both variables in a regression may be redundant and reduce power. The average daily check-in variable is likely to be a worse predictor of business ratings than the count of reviews since it has relatively less variation.

Using the check-in data as a covariate may present issues with reverse causation as well. Assuming customers use the check-in feature to boast about the businesses they visit, a higher Yelp rating would increase the number of check-ins a business receives. This complicates the identification of a causal relationship between these variables because either the business rating or the average number of check-ins must be used as a dependent variable. Though the issue of reverse causation may not be easily solved with the current dataset, a multiple regression including the other covariates (review_count, wk_op_hours) would provide greater insight into the causality of business ratings.

2 Project 2

2.1 The Message

What are the most relevant determinants of a business's Yelp Rating, and do they differ depending on the type of business?

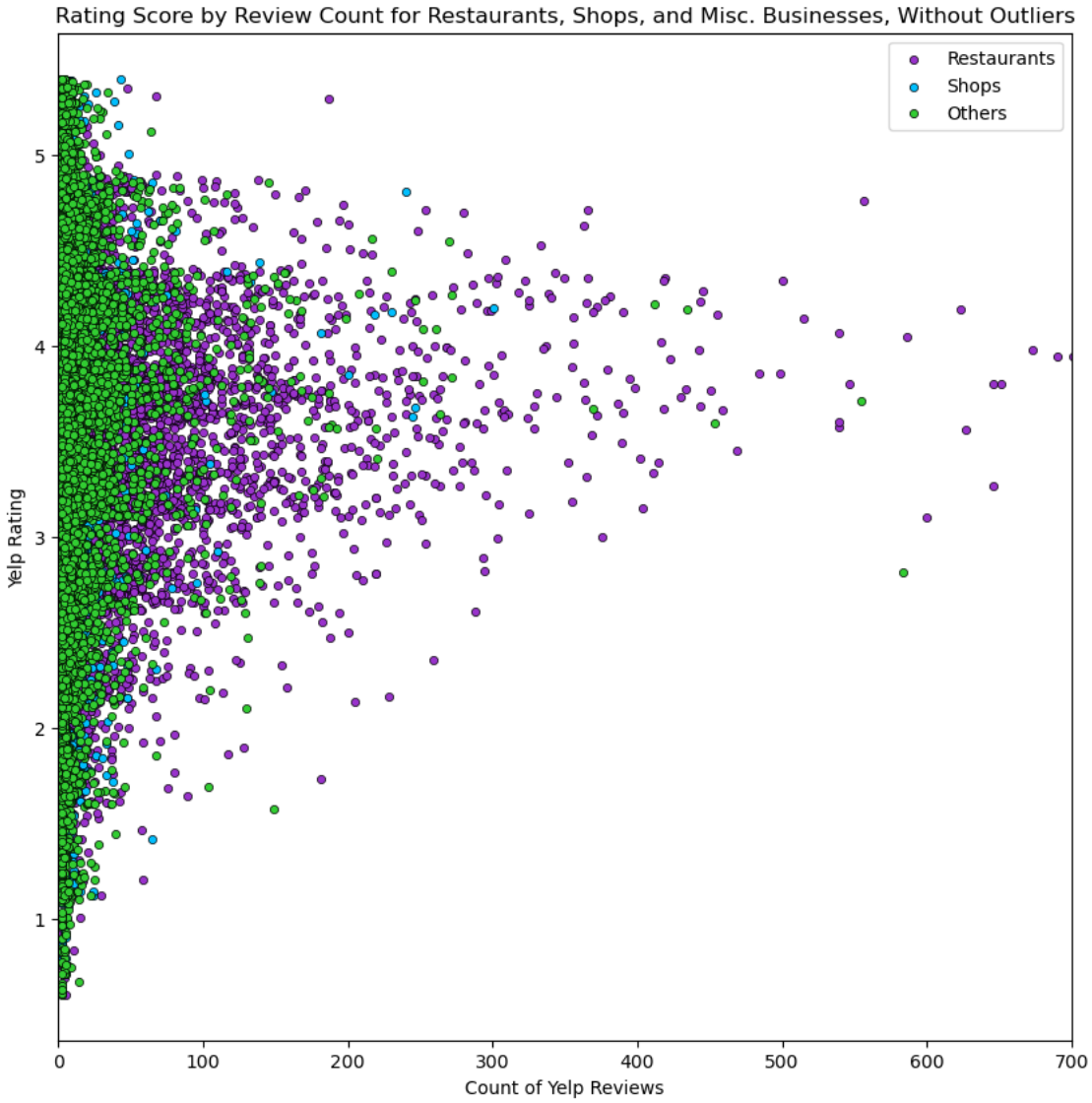
[ ]:
```
# create jitter function for modeling discrete data
def add_jitter(data: pd.Series, jitter: float) -> pd.Series:
    """adds jitter to data. makes discrete data look more clear on graphs."""
    data = data.copy() # copy the series: don't want to modify it
    jitter_amt = (data.max() - data.min()) * jitter # find amount of jitter to␣
 ↪generate: calculated as a % of data's range

    for i in range(len(data)):
        data.iloc[i] = (data.loc[i] + np.random.uniform(low=-jitter_amt,␣
 ↪high=jitter_amt)) # adds jitter to the data

    return data
```

[ ]:
```
# build scatter plot with stars, review_count for restaurants, shops,␣
 ↪miscellaneous
fig, ax = plt.subplots(figsize=(10,10))

not_rst_shop = tor_bsn[tor_bsn['shop'] == 0]
not_rst_shop = not_rst_shop[not_rst_shop['restaurant'] == 0]

#     # find variables to graph based on business type
rst = tor_bsn[['stars', 'review_count']][tor_bsn['restaurant'] == 1]
shop = tor_bsn[['stars', 'review_count']][tor_bsn['shop'] == 1]
other = not_rst_shop[['stars', 'review_count']]

# store relevant data
types_d = {'Restaurants': (rst, 'darkorchid'),
```

```
            'Shops': (shop, 'deepskyblue'),
            'Others': (other, 'limegreen')}

for _type in types_d:
    data, col = types_d[_type]
    data = data.reset_index()
    data['stars'] = add_jitter(data['stars'], 0.1)
    data.plot(ax=ax, kind='scatter', x='review_count', y='stars', c=col,␣
 ↪edgecolor='black', linewidth=0.5)

plt.title('Rating Score by Review Count for Restaurants, Shops, and Misc.␣
 ↪Businesses, Without Outliers')
plt.xlabel('Count of Yelp Reviews')
plt.ylabel('Yelp Rating')
plt.xlim(0, 700)
plt.legend(['Restaurants', 'Shops', 'Others'])
```

```
/Users/thomas/opt/anaconda3/lib/python3.9/site-
packages/pandas/plotting/_matplotlib/core.py:1114: UserWarning: No data for
colormapping provided via 'c'. Parameters 'cmap' will be ignored
  scatter = ax.scatter(
/Users/thomas/opt/anaconda3/lib/python3.9/site-
packages/pandas/plotting/_matplotlib/core.py:1114: UserWarning: No data for
colormapping provided via 'c'. Parameters 'cmap' will be ignored
  scatter = ax.scatter(
/Users/thomas/opt/anaconda3/lib/python3.9/site-
packages/pandas/plotting/_matplotlib/core.py:1114: UserWarning: No data for
colormapping provided via 'c'. Parameters 'cmap' will be ignored
  scatter = ax.scatter(
```

```
[ ]: <matplotlib.legend.Legend at 0x7fa9e54f46d0>
```



This scatter plot demonstrates the relationship between a business's count of Yelp reviews and its Yelp rating. Different colours are used to portrary different types of businesses: restaurants are in purple, shops are in blue, and miscellaneous (non-shops, non-restaurants) are in green. Each type of business also has its own line of best fit, which is calculated with the OLS method. This plot ignores outliers that have more than 700 reviews.

For restaurants, there appears to be significant non-linearity in the distribution of Yelp ratings across reviews counts. Specifically, the distribution appears to be logarithmic. Each additional review provides much greater returns to ratings prior to the one-hundred review count threshold. Beyond this threshold, the returns of an additional review diminish dramatically. After a restaurant has earned roughly two-hundred reviews, the marginal benefit to ratings of an additional review is virtually zero on average.

This nonlinearity is not evident in the relationship between ratings and review count for shops

or miscellaneous businesses. In fact, neither shops nor iscellaneous businesses appear to have any significant correlation between ratings and count of reviews. Shops appear to have significant variation across review counts but there are so few shops in the data that drawing conclusions about the strength of a relationship is difficult. Inversely, there are many miscellaneous business in the data but there is very little variation in their review counts. In reality, miscellaneous businesses may have a similar nonlinear relationship but are perhaps inherently less likely to receive greater numbers of reviews. In order to remedy this in a regression, additional controls should be included.

All in all, this plot provides significant insight with regard to this paper's main message. It demonstrates that the number of Yelp reviews that a restaurant has is correlated with its Yelp rating and is therefore a relevant determinant. However, there are significant diminishing returns to ratings for addtional reviews beyond two-hundred. Moreover, it shows that the effect of a determinant of ratings likely depends on the type of business, at least in the case of review counts. This is because shops and miscellaneous businesses differ from restaurants in regard to how an additional review affects their respective Yelp ratings.

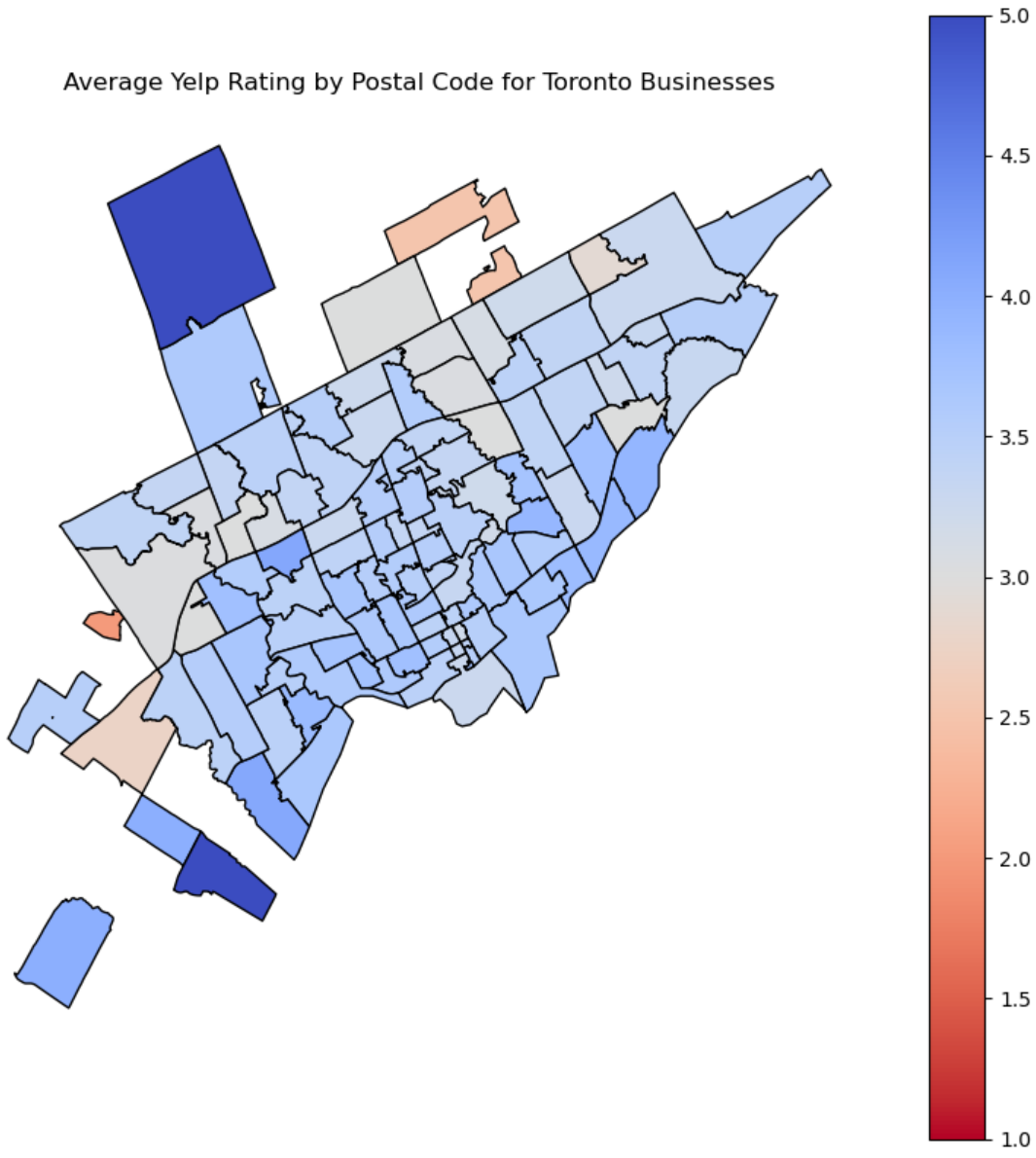
2.2 Maps and Interpretations

[ ]:
```
# use toronto business data with postal code, longitude, latitude columns
df = pd.read_csv('/Users/thomas/Documents/schoolwork/eco225/yelp_data/
 ↪toronto_businesses.csv')
df = df.copy()

# drop irrelevant postal codes
for i in range(2):
    df.drop(df['longitude'].idxmax(), axis=0, inplace=True)

# build geometry column
df['coordinates'] = list(zip(df['longitude'], df['latitude']))
df['coordinates'] = df['coordinates'].apply(Point)
gdf = gpd.GeoDataFrame(df, geometry='coordinates')

# clean postal_code column for merge
gdf['postal_code'] = gdf['postal_code'].str[:3]

# import shp file
toronto = gpd.read_file('/Users/thomas/Documents/schoolwork/eco225/shp/
 ↪lfsa000a21a_e/lfsa000a21a_e.shp')
toronto = toronto.rename(columns={'CFSAUID': 'postal_code'})

# drop more irrelevant postal codes
toronto = toronto[toronto['PRNAME'] == 'Ontario']

# ratings map
gdf_stars = gdf.groupby('postal_code')[['stars']].mean()
t_stars = toronto.merge(gdf_stars, how='inner', on='postal_code')

# review count map
```

```
gdf_reviews = gdf.groupby('postal_code')[['review_count']].mean()
t_reviews = toronto.merge(gdf_reviews, on='postal_code')

# checkins map
gdf_checks = gdf.groupby('postal_code')[['daily_checkin_avg']].mean()
t_checks = toronto.merge(gdf_checks, on='postal_code')

# gdf.to_csv('/Users/thomas/Documents/schoolwork/eco225/yelp_data/master/
 ↪geodata_toronto.csv', index=False) # save gdf
```

[ ]:
```
# plotting code for ratings maps
fig, gax = plt.subplots(figsize=(10,10))
t_stars.plot(ax=gax, edgecolor='black', column='stars', cmap='coolwarm_r',␣
 ↪vmin=1, vmax=5, legend=True)

plt.title('Average Yelp Rating by Postal Code for Toronto Businesses')
plt.axis('off')
```

[ ]:
```
(7196587.255142894, 7248967.602000039, 905011.9602857478, 959920.1825714611)
```

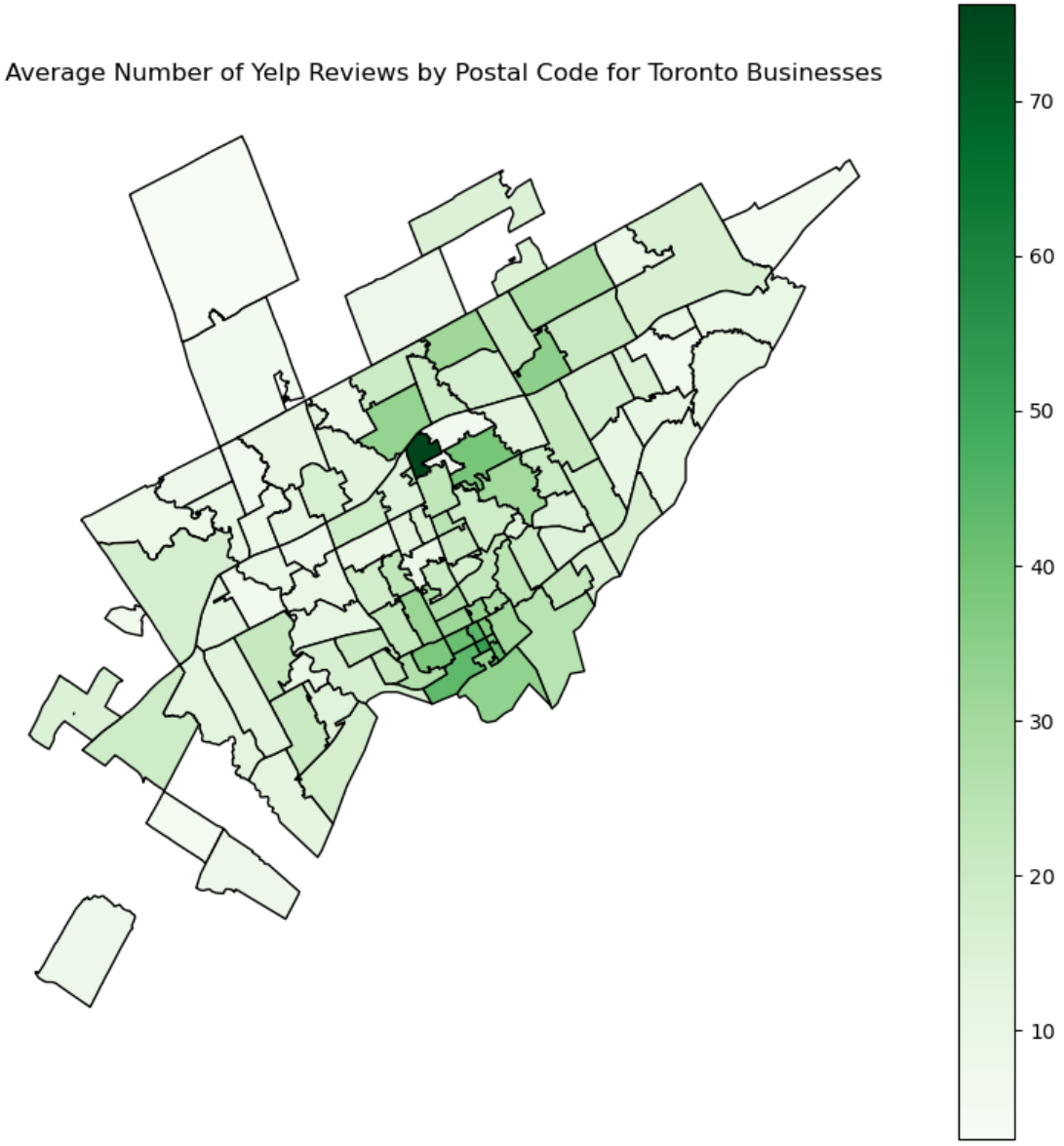


The map of average Yelp ratings by postal code demonstrates that physical geography does not play a significant role in determining Toronto businesses' ratings. Most postal codes have very typical ratings, especially in regions with greater population density. Outliers become more common as distance from downtown increases, but the direction of these outliers seems to be random. This suggests that the prominence of outliers may be the result of each business having relatively fewer reviews. With fewer reviews in these regions, the expected rating is more likely to deviate from the true mean rating.

[ ]:
```
# plotting code for operating hours map
fig, gax = plt.subplots(figsize=(10,10))
```

```
t_reviews.plot(ax=gax, edgecolor='black', column='review_count', cmap='Greens',␣
 ↪legend=True)

plt.title('Average Number of Yelp Reviews by Postal Code for Toronto␣
 ↪Businesses')
plt.axis('off')
```

[ ]: (7196587.255142894, 7248967.602000039, 905011.9602857478, 959920.1825714611)

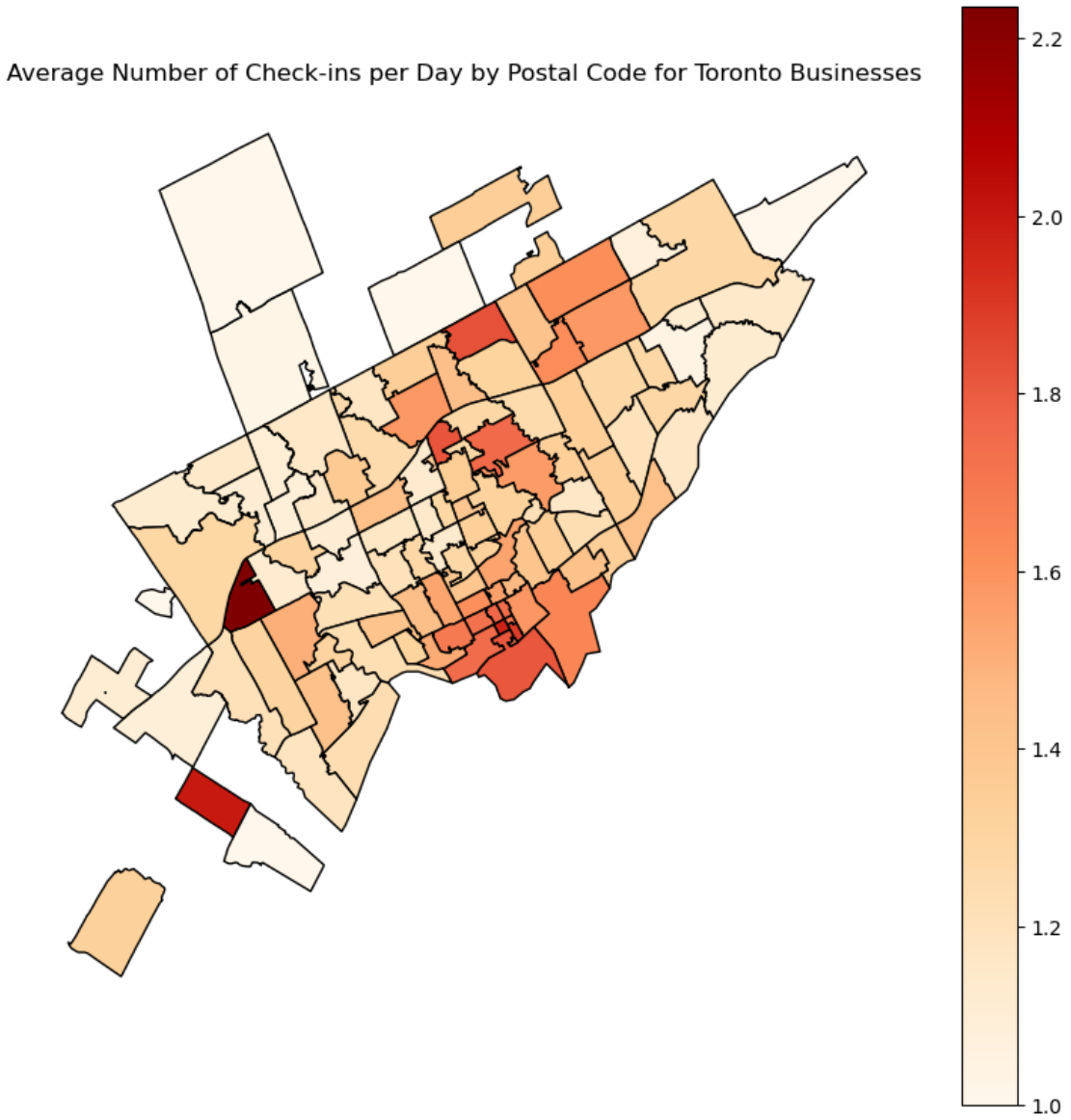


Contrary to the map of average Yelp ratings, the average number of Yelp reviews seems to be influenced by geography. The postal codes with the highest average number of reviews are seen

in regions with the highest population density, such as those in downtown Toronto. This is rather obvious; businesses that operate in areas with high population density will have more potential customers than thoe in low-density regions. This hints at why this paper has been unable to identify a significant positive correlation between the count of reviews and Yelp ratings: without controlling for population density, businesses in regions with larger populations will see more reviews on average regardless of their actual quality. Including a variable that measures the population for each postal code in a regression with the count of reviews would therefore increase the chances of finding a significant relationship between review count and ratings. The population variable would control for the amount of potential customers that businesses may have, thereby isolating the effect of an additional review on ratings.

[ ]:
```
# plotting code for checkins map
fig, gax = plt.subplots(figsize=(10,10))
t_checks.plot(ax=gax, edgecolor='black', column='daily_checkin_avg',␣
 ↪cmap='OrRd', legend=True)

plt.title('Average Number of Check-ins per Day by Postal Code for Toronto␣
 ↪Businesses')
plt.axis('off')
```

[ ]:
```
(7196587.255142894, 7248967.602000039, 905011.9602857478, 959920.1825714611)
```



Similar to the map for average number of reviews, the number of Yelp check-ins a business gets appears to be influenced by geography. The businesses that are located in regions of greater population density, such as downtown, have more check-ins per day on average. However, the number of check-ins seems to be much more volatile than the number of reviews. This map has outliers of greater magnitude and with greater frequency since many postal codes outside of downtown seem to have high average daily check-in counts. This is likely because check-ins are used infrequently on Yelp to begin with. It does not take much for a business to have a much higher average number of check-ins, especially when the mean for Toronto is 1.5 per day. One especially passionate customer could raise the average number of check-ins per day by 1 on their own. This hypothetical example is especially plausible for restaurants, since returning customers are commonplace in the food industry.

3 Project 3

3.1 Potential Data to Scrape

The purpose of this analysis is to identify the determinants of business ratings on Yelp and how these determinants vary across types of businesses. A relevant metric that this paper has been unable to capture so far is the price of products sold by businesses. Often, the price of a product is a significant factor in a consumer's decision to buy. There are two hypotheses for the effect of prices on Yelp ratings. Firstly, the cheap-positive hypothesis: cheaper products may be positively correlated with business ratings. Customers who spend less money on a purchase may make them happier, which in turn improves their perspective on the quality of the quality of the business they were shopping at. Furthermore, customers can shop more frequently at businesses that are affordable. Repeat customers may be more likely to leave positive reviews. On the other hand, the expensive-positive effect may dominate. Cheaper products may suggest low quality for some customers, and would thereby yield negative reviews. By contrast, businesses that sell high-priced goods often advertise themselves as being high-quality. This may affect customer perception and lead to better reviews, even if the objective quality of whatever good or service being transacted is no greater than that of a cheaper alternative. So regardless of a business's price level, it will appeal to some customers and repel others. Including business price level in this analysis would help identify whether the cheap-positive or expensive-positive effect dominates with regard to a business's Yelp rating. This would provide a clearer picture of the determinants of Yelp ratings. The price level of a business is also an important control variable for a multiple regression in this analysis. For example, cheaper businesses may see more customers on average compared to expensive businesses which suggests that cheaper businesses would have more reviews.

On Yelp, the price level of a business is denoted by an integer quantity of dollar signs. The minimum is one dollar sign, which indicates that the business's good or service is priced low. The maximum is four dollar signs, which suggests that the business is among the most expensive in the region. Examples can be found here. This data can be web scraped using Yelp's Fusion API. This analysis will use a library built to simplify Yelp Fusion for Python, called yelpapi. Only Toronto businesses need to be scraped, in accordance with the scope of this study. Once scraped, an outer merge on this analysis's Toronto businesses dataset would provide the number of dollar signs for each business (for which price data is available).

3.2 Potential Challenges

Scraping on Yelp's website is against their terms of service, so performing HTML scraping would be difficult. However, Yelp's Fusion API allows users to access Yelp data free of charge. As a result, the main technical challenge involved with accessing Yelp data is learning how to use Yelp Fusion. Moreover, Yelp Fusion only allows 5000 calls to be made per day per user. Since the dataset it will be merged with has 14 850 businesses, it may take several days before all businesses can be accounted for. This assumes that each business in this dataset actually has price data, which is likely not the case. Additionally, current Yelp data may not represent the state of businesses in 2017, which is when the data that this analysis uses was collected. Relative prices may have increased or decreased since then and many businesses were forced to close during the COVID-19 pandemic. So even if price data can be gathered for most businesses through Yelp Fusion, its relevance to 2017 data may be limited. If there are very few businesses with price data available, then any regressions that are conducted on the dataset would have limited applicability and potential bias.

3.3 Scraping Data from a Website

[ ]:
```
from yelpapi import YelpAPI

# query API for Toronto businesses
yelp =␣
 ↪YelpAPI('WhtldkCQ1Pi_KwG2OSCkT4lrN0tELggS0DKVbB6je__22XCVRyKASKtkmgoogNfRQVdgAY7XSsbZ94YSod
ids_ = tor_bsn['business_id']
scraped = []

# for business in ids_[:14850]: # have to run in increments of 5000 ids
#     try:
#         scraped.append(yelp.business_query(business))
#     except:
#         pass # won't exit the loop if the business id can't be found

# save data to a file so I don't have to use more API calls when I want to␣
 ↪access it
# code for the first time I saved the data:
# with open('toronto_scraped.json', 'w') as out_file:
#     json.dump(scraped, out_file)

# code for subsequent saves:
with open('/Users/thomas/Documents/schoolwork/eco225/scraped/toronto_scraped.
 ↪json', 'r') as f:
    data = json.load(f) # load existing file with data

data.extend(scraped) # add the newly scraped data to the loaded data

with open('/Users/thomas/Documents/schoolwork/eco225/scraped/toronto_scraped.
 ↪json', 'w') as f:
    json.dump(data, f) # dump all data back into the json

yelp.close() # recommended practice for this api
```

3.4 Merging the Scraped Dataset

[ ]:
```
scraped_df = pd.DataFrame(columns=['business_id', 'price_level']) # build␣
 ↪dataframe to collect data

# add scraped data to the dataframe
i = 0
with open('/Users/thomas/Documents/schoolwork/eco225/scraped/toronto_scraped.
 ↪json', 'r') as f:
    data = json.load(f)
    for business in data: # data refers to a list, business refers to a␣
 ↪dictionary
```

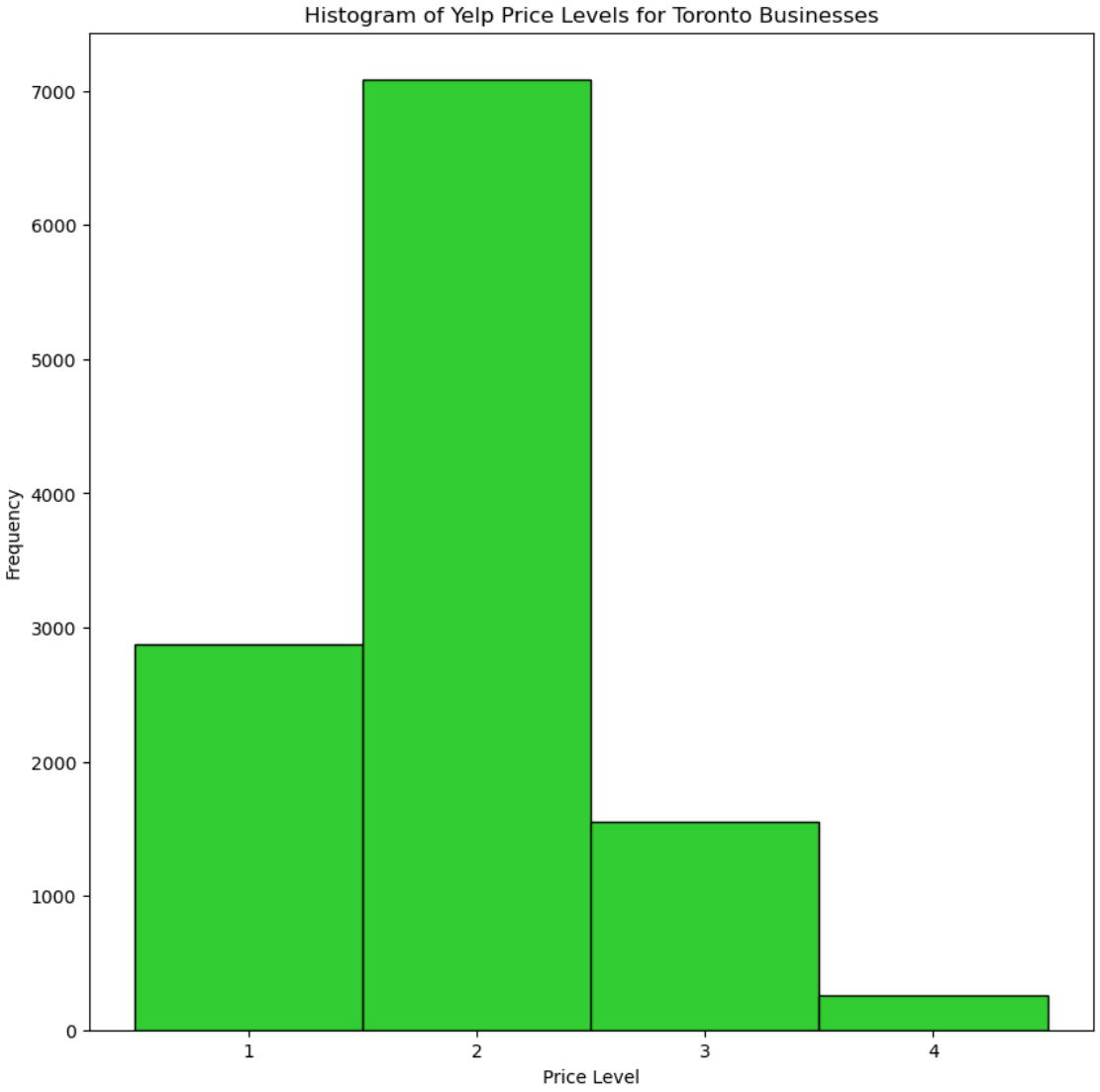
```
        try:
            scraped_df.loc[i] = [business['id'], len(business['price'])]
            i += 1
        except:
            pass # won't exit the loop if the business has no price attribute

tor_df = pd.read_csv('/Users/thomas/Documents/schoolwork/eco225/yelp_data/
 ↪master/geodata_toronto.csv') # load data
tor_price = tor_df.merge(scraped_df, how='outer', on='business_id') # merge␣
 ↪data with dataframe
tor_price = tor_price.drop(columns=['Unnamed: 0']) # drop unneeded columns
tor_price['price_level'] = tor_price['price_level'].fillna(0).astype(int) #␣
 ↪clean new column

# output to csv: create master dataset
# tor_price.to_csv('/Users/thomas/Documents/schoolwork/eco225/yelp_data/master/
 ↪toronto_yelp.csv', index=False)
```

There are 14 849 observations in this dataset after merging the scraped data.

3.5 Visualizing the Scraped Dataset

[ ]:
```
# histogram of price levels
fig, ax = plt.subplots(figsize=(10,10))

tor_price['price_level'].plot(kind='hist', color='limegreen', bins=[0.5, 1.5, 2.
 ↪5, 3.5, 4.5], ec='black', title='Histogram of Yelp Price Levels for Toronto␣
 ↪Businesses')
ax.locator_params(axis='x', integer=True)

plt.xlabel('Price Level')
```

[ ]:
```
Text(0.5, 0, 'Price Level')
```



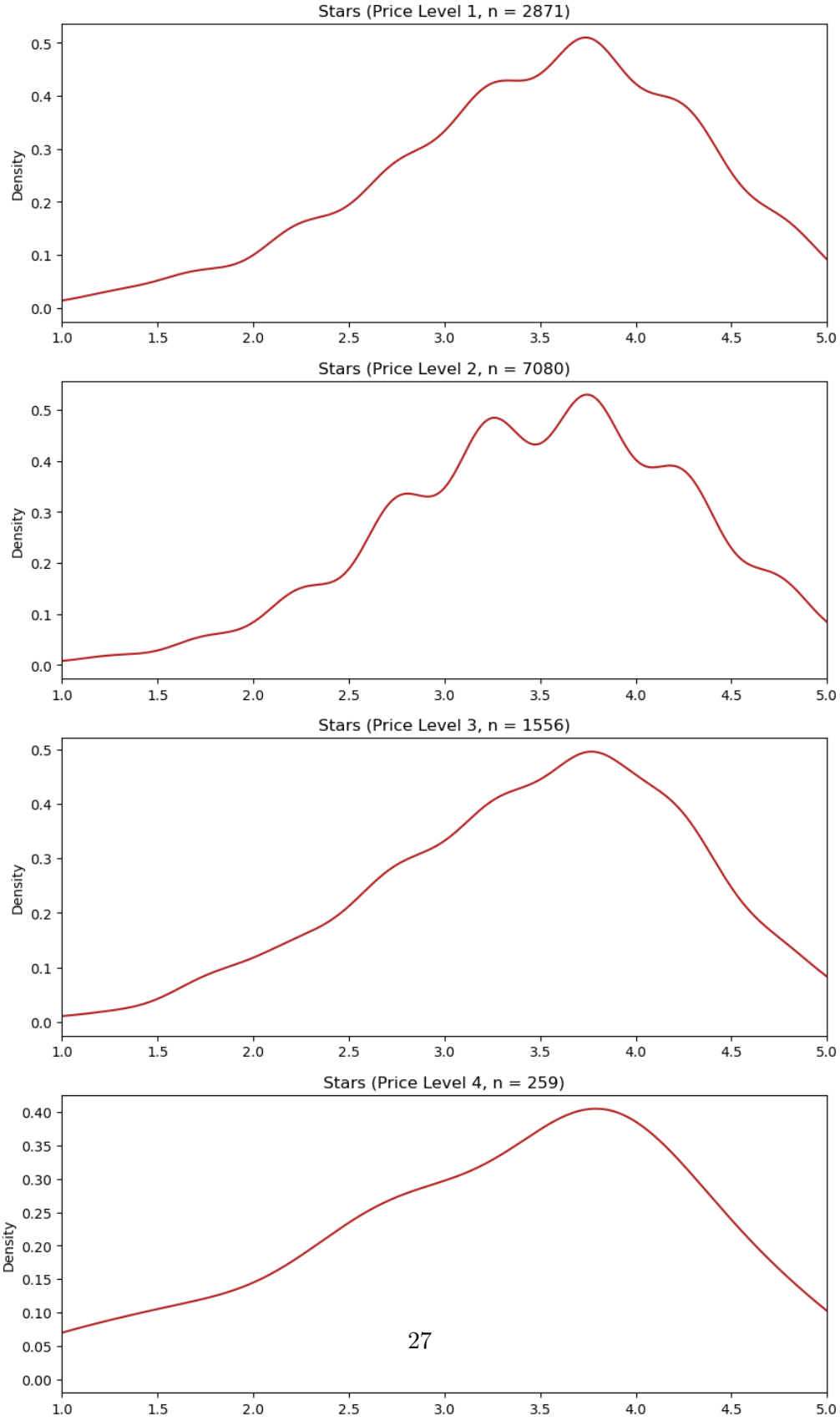
The price levels of most businesses in Toronto are two dollar signs or below on Yelp. Since there are many more observations at a price level of two or less compared to a price level of 3 or more, this analysis may be unable to determine any relationship between the price level and Yelp ratings. This is because there is a great lack of variation in the data. As the vast majority of businesses have an average price level of roughly 2, a regression including the price level would probably be unable to determine a statistically significant effect.

[ ]:
```
fig, (ax1, ax2, ax3, ax4) = plt.subplots(4,1, figsize=(10,18))

axs = (ax1, ax2, ax3, ax4)

# density plot to see distribution of ratings across price levels
```

```
prices_df = [tor_price.loc[tor_price['price_level'] == price] for price in␣
 ↪range(1, 5)]

for price_level in range(4):
    data = prices_df[price_level]['stars'].reset_index().drop(columns='index')
    count = data.count()[0]
    stars = add_jitter(data, 0.1)
    stars.plot(ax=axs[price_level], kind='kde', c='firebrick', xlim=(1,5),␣
 ↪legend=False).set_title(f"Stars (Price Level {price_level + 1}, n =␣
 ↪{count})")

plt.suptitle('Distribution of Yelp Ratings for Each Price Level')
```

[ ]: Text(0.5, 0.98, 'Distribution of Yelp Ratings for Each Price Level')

Distribution of Yelp Ratings for Each Price Level



These density plots use jitter to mitigate the serrated pattern that would be otherwise generated, since the price level is a discrete variable. The shape of the distribution of Yelp ratings appears very similar across all price levels. This suggests that the price level is not a determinant of Yelp ratings among businesses in Toronto. The average Yelp rating across all price levels is 3.5, which is approximately the average for each individual price level as well. However, this plot does not demonstrate the strength of this correlation. While Yelp ratings do not necessarily change with the price level, it could be that the price level has precise null effects on the Yelp rating. A formal regression analysis would provide insight in this regard.

Neither the cheap-positive and expensive-positive hypotheses mentioned in section 3.1 are evidenced by this plot. The lack of an apparent correlation between Yelp ratings and the price level could indicate that both hypotheses are true, but cancel each other out. Some customers dislike businesses with expensive goods while others prefer them. The same may be true for businesses with cheap goods. Therefore, each business will gain and lose customers regardless of how they price their products. A regression model that uses a polynomial term for the price level may present non-linearity and potentially demonstrate if there is a price level sweetspot with regard to the maxmization of Yelp ratings.

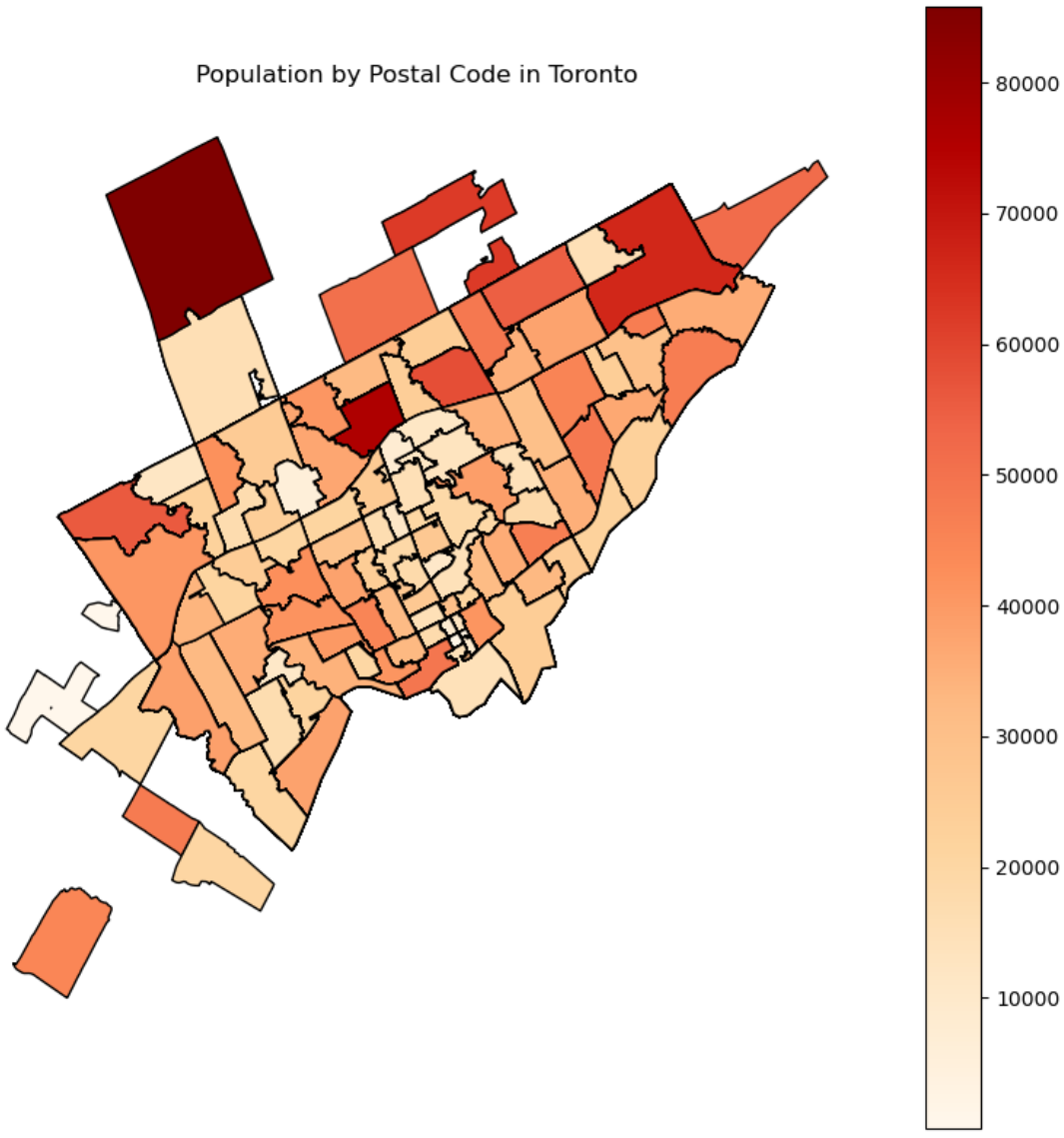
3.6 Adding a New Dataset

This data was acquired from Statistics Canada. It lists the population of each forward sortation area (FSA) in Canada. The first three characters of Canadian postal codes refer to the FSA, which means this data can be merged with this analysis's datasets on the postal code. Following the inferences made in section 2.2, population may be causing omitted variable bias in identifying the magnitude of the effect of the number of Yelp reviews on the Yelp rating for Toronto businesses. By including a population variable in the dataset, the relationship between population, the number of reviews, and ratings can be determined.

[ ]:
```
# merge population data
pop_df = pd.read_csv('/Users/thomas/Documents/schoolwork/eco225/fsa_pop/fsa_pop.
 ↪CSV') # load population data
tor_df = pd.read_csv('/Users/thomas/Documents/schoolwork/eco225/yelp_data/
 ↪master/toronto_yelp.csv') # load master data

pop_df = pop_df[['Geographic code', 'Population, 2016']]
pop_df = pop_df.rename(columns={'Geographic code': 'postal_code', 'Population,␣
 ↪2016': 'postal_pop'})
tor_pop = tor_df.merge(pop_df, how='inner', on='postal_code') # recall GDF␣
 ↪refers to the dataframe with all variables and geo data

# save to csv
# tor_pop.to_csv('/Users/thomas/Documents/schoolwork/eco225/yelp_data/master/
 ↪toronto_yelp.csv', index=False)
```

[ ]:
```
# population map
t_pop = toronto.merge(tor_pop[['postal_code', 'postal_pop']], how='inner',␣
 ↪on='postal_code')

fig, gax = plt.subplots(figsize=(10,10))
t_pop.plot(ax=gax, edgecolor='black', column='postal_pop', cmap='OrRd',␣
 ↪legend=True)

plt.title('Population by Postal Code in Toronto')
plt.axis('off')
```

[ ]:
```
(7196587.255142894, 7248967.602000039, 905011.9602857478, 959920.1825714611)
```

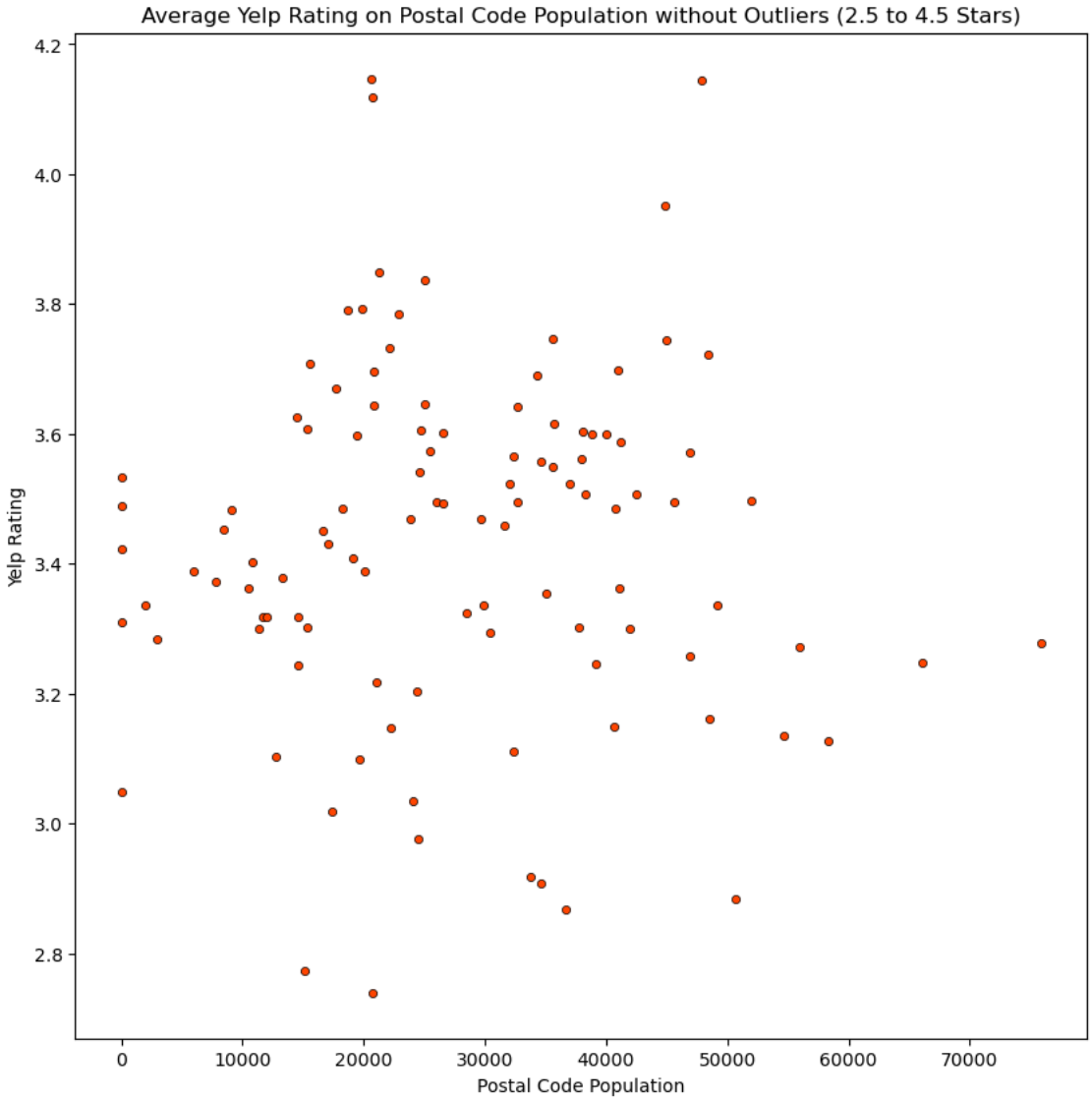


This plot demonstrates that majority of downtown postal codes in Toronto actually have relatively smaller populations than regions outside of downtown. As a result, there is no clear correlation between the population of a postal code and the number of reviews for businesses in Toronto. The hypothesis proposed by section 2.2 is not substantiated. However, it does not prove that the population is not a confounding variable in this paper's analysis of business ratings. Population may still be correlated with another determinant of business ratings, such as the daily average number of check-ins. Alternatively, population may itself be a determinant of business ratings. This likely means that the determinants for a business's number of a reviews depends on the quality of the business, its products, and its service as opposed to their characteristics of its locality.

[ ]:
```
# create aggregated postal code data
tor_pop_agg = tor_pop.groupby('postal_code')[
    ['postal_pop', 'stars']].mean().reset_index()
tor_pop_agg['jittered_stars'] = add_jitter(tor_pop_agg['stars'], 0.05)
tpa_no_outs = tor_pop_agg[tor_pop_agg['stars'] < 4.5]
tpa_no_outs = tpa_no_outs[tpa_no_outs['stars'] > 2.5]

# scatterplot
fig, ax = plt.subplots(figsize=(10, 10))
tpa_no_outs.plot(ax=ax, kind='scatter', x='postal_pop', y='jittered_stars',␣
 ↪c='orangered',
                 ec='black', linewidth=0.5, title='Average Yelp Rating on␣
 ↪Postal Code Population without Outliers (2.5 to 4.5 Stars)')

plt.xlabel('Postal Code Population')
plt.ylabel('Yelp Rating')
```

```
/Users/thomas/opt/anaconda3/lib/python3.9/site-
packages/pandas/plotting/_matplotlib/core.py:1114: UserWarning: No data for
colormapping provided via 'c'. Parameters 'cmap' will be ignored
  scatter = ax.scatter(
```

[ ]:
```
Text(0, 0.5, 'Yelp Rating')
```



A business's postal code population is uncorrelated with its Yelp ratings. As a result, it is not a confounding variable in this analysis. This suggests that a business's Yelp rating will be any better or worse depending on the population of its region. This is beneficial to business owners that are seeking to maximize their Yelp rating because good reviews can still be earned while operating in less-busy areas. In theory, lower population areas suggest fewer customers for businesses, fewer necessary hours of operation, and cheaper rent. Should business owners choose to locate themselves in low-density areas, costs can be kept low without making their Yelp ratings suffer.

4 Final Project

4.1 Regressions

4.1.1 Hypotheses

The relationship between Yelp ratings and each variable is likely non-linear. This is demonstrated in section 2.1, as restaurants appear to have a positive logarithmic relationship between review count and ratings. The same is likely true for daily average check-ins, modeled in section 1.4, which has a very similar scatter plot to the one in section 2.1. The density plot of operating hours per week in section 1.4 suggests that the relationship between ratings and operating hours may linear and negative. This plot does not graph the distribution of operating hours for businesses with ratings between one and five stars, so this hypothesis may not be accurate. The price level is likely not correlated with Yelp ratings, in accordance with the distribution plots of each price level in section 3.4 all having roughly the same rating average. Lastly, postal code population is not related to business ratings as demonstrated in section 3.6.
Choosing Covariates In accordance with the theories of this paper so far, the count of reviews, the average number of daily check-ins, and the number of operating hours per week should be included in regressions for the Yelp rating. There are clear non-linear relationships with Yelp ratings for the count of reviews and average check-ins. There is also evidence of a potential relationship between weekly operating hours and ratings. As a result, the inclusion of these primary variables of interest should explain some of the variation in Yelp ratings.

Though the variables measuring postal code population and business price level are not obviously correlated with Yelp ratings, they are still important control variables. Including these in a regression model will enable the isolation of variation in the primary variables of interest. As a result, they may contribute to increasing statistical significance for regression results.

4.1.2 Regression Table

[ ]:
```
tor_pop['constant'] = 1
tor_pop['ln_review'] = np.log(tor_pop['review_count'])
tor_pop['ln_checkin'] = np.log(tor_pop['daily_checkin_avg'])
tor_pop['review2'] = tor_pop['review_count'] ** 2
tor_pop['checkin2'] = tor_pop['daily_checkin_avg'] ** 2
tor_pop = pd.get_dummies(tor_pop, columns=['postal_code'], drop_first=True)
```

[ ]:
```
pc = tor_pop.columns[24:]
vars_1 = ['constant', 'review_count', 'wk_op_hours', 'daily_checkin_avg',␣
 ↪'price_level']
vars_2 = vars_1.copy()
vars_2.extend(pc)
vars_3 = ['constant', 'ln_review', 'wk_op_hours', 'ln_checkin', 'price_level']
vars_4 = vars_3.copy()
vars_4.extend(pc)
vars_5 = ['constant', 'review_count', 'review2', 'wk_op_hours',␣
 ↪'daily_checkin_avg', 'checkin2', 'price_level']
vars_6 = vars_5.copy()
vars_6.extend(pc)
```

```
r1 = sm.OLS(tor_pop['stars'], tor_pop[vars_1], missing='drop').
 ↪fit(cov_type='HC0')
r2 = sm.OLS(tor_pop['stars'], tor_pop[vars_2], missing='drop').
 ↪fit(cov_type='HC0')
r3 = sm.OLS(tor_pop['stars'], tor_pop[vars_3], missing='drop').
 ↪fit(cov_type='HC0')
r4 = sm.OLS(tor_pop['stars'], tor_pop[vars_4], missing='drop').
 ↪fit(cov_type='HC0')
r5 = sm.OLS(tor_pop['stars'], tor_pop[vars_5], missing='drop').
 ↪fit(cov_type='HC0')
r6 = sm.OLS(tor_pop['stars'], tor_pop[vars_6], missing='drop').
 ↪fit(cov_type='HC0')

sg = Stargazer([r1, r2, r6, r4])
sg.covariate_order(['constant', 'review_count', 'review2', 'ln_review',␣
 ↪'daily_checkin_avg', 'checkin2', 'ln_checkin', 'wk_op_hours', 'price_level'])
sg.custom_columns(['Linear', 'Linear w/ Postal Codes', 'Squared Reviews,␣
 ↪Checkins w/ Postal Codes', 'Log Reviews, Checkins w/ Postal Codes'],␣
 ↪[1,1,1,1])
HTML(sg.render_html())
```

```
/Users/thomas/opt/anaconda3/lib/python3.9/site-
packages/statsmodels/base/model.py:1871: ValueWarning: covariance of constraints
does not have full rank. The number of constraints is 117, but rank is 108
  warnings.warn('covariance of constraints does not have full '
/Users/thomas/opt/anaconda3/lib/python3.9/site-
packages/statsmodels/base/model.py:1871: ValueWarning: covariance of constraints
does not have full rank. The number of constraints is 119, but rank is 110
  warnings.warn('covariance of constraints does not have full '
/Users/thomas/opt/anaconda3/lib/python3.9/site-
packages/statsmodels/base/model.py:1871: ValueWarning: covariance of constraints
does not have full rank. The number of constraints is 117, but rank is 108
  warnings.warn('covariance of constraints does not have full '
```

[ ]: <IPython.core.display.HTML object>

4.1.3 Justification

These regressions consider the primary variables of interest with slight modifications across specifications. Regression 1 uses linear specifications for each variable and does not control for postal code. Regression 2 uses the same variables and their specifications but controls for the postal code. Regressions 3 and 4 are parallel to regression 1 but includes quadratic and logarithmic terms for the number of reviews and the average number of check-ins per day, respectfully. Their inclusion is necessary to identify the nature of these variables' relationships with business ratings. Moreover, it will indicate whether there is a sweetspot for review count and daily check-ins. Regressions 2 through 4 and their comparison to regression 1 will also demonstrate whether there is significant

variation across postal codes with regard to the determinants of business ratings. ### Preferred Specification The best specification among the ones constructed is regression 3. This is because it is tied for the greatest R^2 value, meaning it accounts for the most amount of variation among the specifications used. Regression 3's F-stat is also the largest, meaning the covariates included in its specification have the greatest joint significance. Nearly all of the covariates of interest are statistically significant at the 1 percent level, except for the squared term on review count which has a p-value greater than 0.05. By extension, it accounts for the nonlinear relationships between Yelp ratings, review count and average check-ins in the most statistically significant way. ### Evaluating Regressions In order to identify the relevance of the chosen covariates in determining Yelp ratings, the R^2 is an especially important statistic. It represents the amount of variation in ratings that is explained by the set covariates for each regression. A higher R^2 suggests that a specification's covariates explain more of the variation in the dependent variable. In this context, a higher R^2 means a specification is better at predicting ratings. The F-statistic for each regression is also an important criterion for evaluating the performance of specifications because it represents the joint significance of the chosen covariates. A high F-statistic means each covariate is likely to be correlated with the dependent variable. Similar to the R^2 value, this means that a higher F-statistic demonstrates that the covariates are better predictors of ratings. Lastly, the significance of each individual determinant of ratings is also important for evaluating specifications. Low standard errors and large magnitudes for coefficients suggest that the relationship measured between an independent and dependent variable is practically relevant and precise. However practically insignificant they may be, coefficients with high statistical significance are valuable because they indicate that a correlation exists between a covariate and Yelp ratings. In other words, statistically significant coefficients are relevant determinants of ratings. ### Results The most obvious insight presented by this suite of regressions is that there is significant variation across postal codes. Regression 1, which does not control for postal codes, has an R^2 of 0.012. All specifications that control for the suite of dummies for postal codes have an R^2 of 0.058. This means that roughly five percentage points more variation in Yelp ratings are explained by differences across postal codes in Toronto, and therefore a business's postal code partially determines its Yelp rating. That said, 0.058 is still a very low R^2. This means that much more significant determinants of ratings exist which are unaccounted for in these regressions.

The number of reviews a business has is also a significant determinant of Yelp ratings in every specification, though to varying degrees of practical significance. Regressions 1 through 3 indicate that an increase of 100 in a business's count of reviews is correlated with a 0.1-point increase in the Yelp rating on average. Regression 4 suggests that a 1-percent increase in the number of reviews is correlated with a 0.034 point increase in ratings. As ratings are rounded to the nearest half-point, this correlation would only be visible on Yelp for businesses that have 500 more ratings. Considering that the average number of reviews is roughly 28.3 and the seventy-fifth percentile in the distribution of the review count variable is 28, the magnitude of the coefficient for the number of reviews is remarkably insignificant in the practical sense. Moreover, regression 3 suggests that the number of reviews may have diminishing returns to ratings since the coefficient on the squared term for review count is negative. However, this finding is not resolute because this coefficient has low statistical significance (P<0.1). While review count is certainly a statistically significant predictor of ratings, it is not feasible for a business to increase its Yelp rating by increasing its number of reviews alone. This is because reviews are not inherently good or bad: without controlling for business quality, every additional review is equally likely to increase or decrease the business's rating.

The average daily number of check-ins is a significant predictor of Yelp ratings. Regression 3

indicates that there is significant non-linearity in the relationship between check-ins and ratings. This suggests that there is diminishing returns to ratings for the number of check-ins that a business receives. An increase of 10 in the average number of check-ins per day is correlated with a 0.43 increase in ratings, which is rather large considering that would almost be enough to increase the rating portrayed on a business's Yelp page. This means that the average number of daily check-ins is a significant predictor of ratings. This makes intuitive sense since a check-in is a clear indicator that a customer enjoys shopping at a business; if a customer didn't like a business, they wouldn't want to show off that they are shopping there, so they would not check in.

The number of weekly operating hours and price level are not relevant predictors of ratings. Each specification finds that one additional hour of operation per week is correlated with a 0.002 higher rating. Since there are only 168 hours in a week, operating hours can only be correlated with a 0.336 ($168 \cdot 0.002$) increase in ratings points at most. Similarly, the price level is an integer quantity between 0 and 4. Using the largest coefficient found for this variable (regression 4), the price level can be correlated with a 0.01 ($3 \cdot -0.033$) ratings point decrease at most. Curiously, the coefficient on the price level is negative and statistically significant in each regression. This suggests that higher prices are not conducive to higher ratings, though the negative effect they have on ratings is negligible.

All in all, the only practically significant determinants of ratings found are the average daily number of check-ins and postal code. Though statistiaclly significant, the number of reviews, price level, and number of hours of operation per week are practically insignificant.

4.2 Machine Learning

4.2.1 Objective Function for a Regression Tree

Let j refer to the number of mutually-exclusive regions in the regression space. Let s refer to the optimal threshold on which to split the regression space in order to minimize the sum of squared residuals (RSS). The objective function is:

$$\min_{j,s} \left(\sum_{i:x_{i,j} \leq s, x_i \in R1} (stars_i - \hat{stars}_{R_1})^2 + \sum_{i:x_{i,j} > s, x_i \in R2} (stars_i - \hat{stars}_{R_2})^2 \right)$$

where $\hat{y}_{R_1}$ and $\hat{y}_{R_2}$ refer to the average of Yelp ratings in each regression space:

$$\hat{stars}_{R_m} = Average(stars_i | x_i \in R_m)$$

R_1 and R_2 refer to partitions of the regression space given by:

$$R_1 = \{X | X_j < s\} R_2 = \{X | X_j \geq s\}$$

X belongs to the chosen covariates:

$$X \in \{review, checkin, hour, price\}$$

For each partition, the objective function is re-evaluated with sub-partitions of the regression space. The tree is then pruned according to the following function:

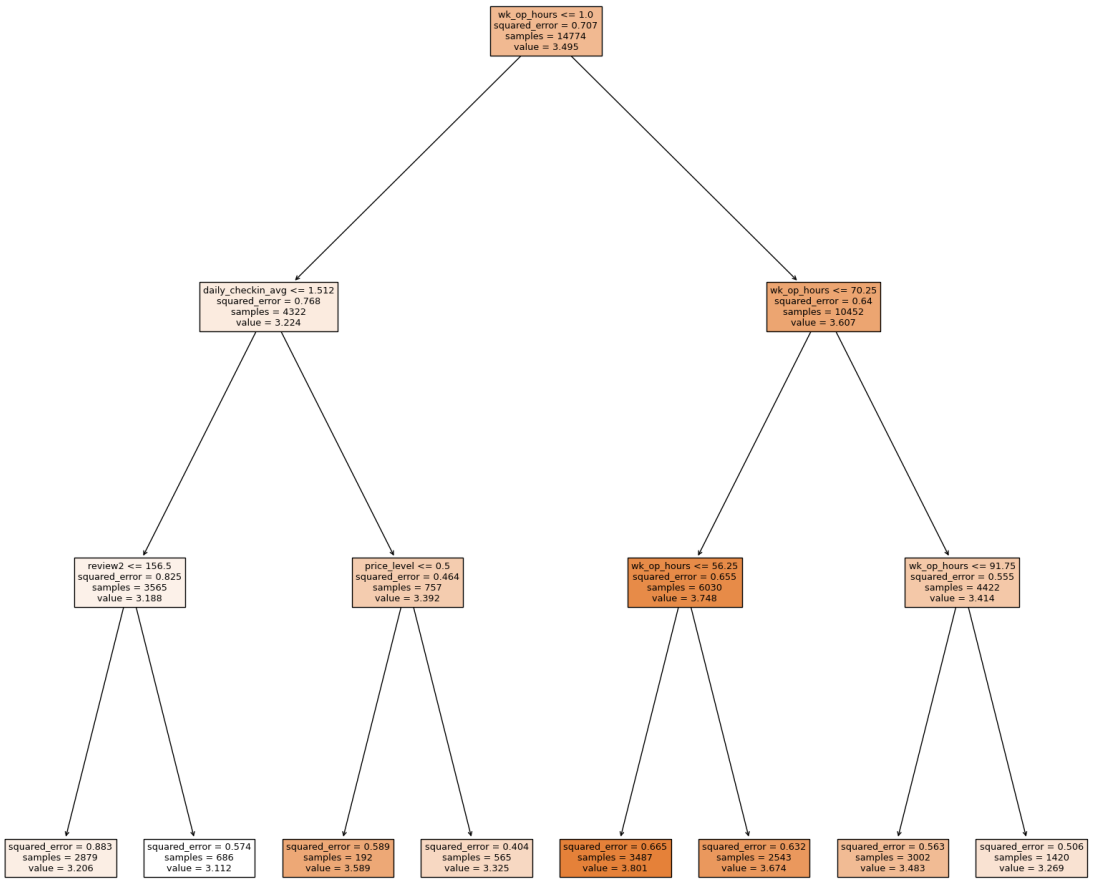
$$\min_{tree \subset T} \sum (\hat{f}(x) - stars)^2 + \alpha |\text{terminal nodes in tree}|$$

where α is chosen to minimize this equation.

4.2.2 Regularization Parameters

There are three main regulization parameters in this model. They are the minimum leaf size, maximium tree depth, and the alpha parameter. The minimum leaf size contributes to preventing prediction from being too granular. Hyperspecific groupings have little relevance because they are incapable of producing significant predictions for observations that differ even slightly from them. Leafs must not be too small in order to ensure a reasonable prediction can be made about an observation that is not identical to the ones that were used to construct the regression tree. The maximum tree depth is important to regulate for a similar reason. Splitting on many variables is conducive to producing smaller leaves, which would make the predictions of a regression tree not generalizeable. The alpha parameter also contributes to balancing complexity and quality of the regression tree by increasing the MSE for highly-specified models. For each additional terminal node in a regression tree, the MSE is increased by alpha (which must be greater than 0). ### Creating the Regression Tree

[ ]:
```
# fit tree
from sklearn import tree
ratings_tree = tree.DecisionTreeRegressor(max_depth=3).fit(tor_pop[vars_5],␣
 ↪tor_pop['stars'])

# plot tree
ratings_plot = plt.figure(figsize=(20,20))
ratings_plot = tree.plot_tree(ratings_tree, feature_names=tor_pop[vars_5].
 ↪columns, filled=True)
```



4.2.3 Results

This regression tree indicates that there are many business that have no operating hours listed on Yelp (< 1.0) in this dataset. Among them, the businesses with the greatest ratings have fewer than 1.5 check-ins per day on average and no price data listed on Yelp (< 0.5). For businesses that do have operating hours listed on Yelp, operating hours appear to be negatively correlated with ratings. The best ratings overall are found among the businesses that have fewer than 56.25 hours of operation per week.

[ ]:
```
# predict
ratings_pred = ratings_tree.predict(tor_pop[vars_5])
tree_mse = sklearn.metrics.mean_squared_error(tor_pop['stars'], ratings_pred)
```

```
# mse for regression 3
from sklearn import linear_model
ratings_lr = linear_model.LinearRegression()
ratings_lr.fit(tor_pop[vars_6], tor_pop['stars'])
lm_mse = sklearn.metrics.mean_squared_error(tor_pop['stars'], ratings_lr.
 ↪predict(tor_pop[vars_6]))

# print mse
print(f'MSE (tree): {tree_mse}\nMSE (lm): {lm_mse}')
```

```
MSE (tree): 0.6505375440177438
MSE (lm): 0.6662607367051033
```

4.2.4 Error of Prediction

The MSE for the regression tree is 0.651. The average error is therefore ~0.81 stars ($\sqrt{0.651}$). As ratings must be between 1 and 5, this is extremely large for the error of prediction for each individual business's rating. This indicates that the chosen covariates are poor predictors of Yelp ratings.

4.2.5 Regression Tree vs. Multivariate Linear Regression

Both the OLS and regression tree models predict very low magnitudes of correlation among most of the covariates. The range between the minimum and maximum value of ratings among leaf nodes is less than 0.7. Since the mean rating is ~3.5, a variation of 0.7 is only enough to increase a business's displayed rating by 0.5. The regression tree indicates that this the range between the lowest and highest predicted values of business ratings, so most of the variation in ratings must not be determined by the chosen covariates. Similarly, the OLS models in section 4.1.3 have coefficients of very low magnitude. Though most are statistically significant, that is likely because the number of observations (>14000) yields great statistical power. The OLS and regression tree models both indicate that the chosen covariates are not relevant determinants of business ratings.

The regression tree also demonstrates that much of the variation in ratings occurs in relation to the number of operating hours per week. This is evidenced by all subtrees of the right subtree, and the tree's root itself, being split on this variable. Moreover, the regression tree indicates that having fewer weekly operating hours is correlated with higher Yelp ratings. This runs contrary to the results of the OLS model, which indicated that weekly operating hours were slightly positively correlated with ratings. This is likely because the OLS model controls for the other covariates, whereas the regression tree does not. This suggests that operating hours are confounded by one of the other covariates in the analysis, since their influence on ratings is much larger when variables are not controlled for. As a result, the regression tree overestimates the returns of a business's operating hours to its rating.

4.3 Conclusion

Prior to this paper, the relevant literature did not identify how business metrics on Yelp may affect business ratings. This analysis demonstrates that the number of reviews and average number of check-ins per day appear to be slightly positively correlated with business ratings in Toronto on Yelp. According to the OLS results, there are precise null effects for weekly hours of operation

on business ratings. In other words, high-quality businesses tend to have more reviews, are open for longer, and receive more check-ins on average relative to poor-quality businesses. Check-ins have a significant quadratic relationship with ratings, meaning there are diminishing returns to having high numbers of check-ins. Businesses that are located in high-population regions do not seem to gain any advantage or disadvantage with regard to ratings. That said, a relatively large portion of variation in ratings is captured by postal codes. Since there is no apparent correlation between geographical location and ratings in Toronto, it could be that variation across postal codes is essentially random. The price level of a business is negatively correlated with ratings but is neglible. While these relationships are all statistically significant, only the number of check-ins seems to have practical significance in determining ratings.

Since the number of reviews, weekly operating hours, postal code population, and price level are all relatively insignificant, this study has been largely unsucessful in identifying the determinants of Yelp ratings for Toronto businesses. Regression results indicated that this analysis accounted for no more than six percent of the variation in business ratings. It is clear that most of the data presented on Yelp is actually not relevant in predicting business's Yelp ratings. This makes intuitive sense, since businesses can very greatly in price, availability, and location; these descriptors, though relevant to consumers, are not inherent indicators of business quality. Businesses can be very popular and earn many reviews, but they may not be of high quality. Similarly, businesses can be open for long hours but are not necessarily better as a result. The price level of a business may be a prominent indicator of popularity, but is not a clear predictor of ratings. Cheaper products may be of lesser quality relative to expensive products, but expensive products are less accessible and not inherently better than cheaper products.

This study has also been unable to identify causal relationships. It is likely that Yelp ratings can themselves cause increase in the number of reviews or check-ins. Businesses with high ratings likely garner more attention, which may lead to an influx of customers willing to write reviews and check-in. Though they are significant, the prospect of reverse causation further suggests that these variables are not strong predictors of ratings.

The invalidity of these covariates demonstrates that business ratings are better predicted by variables that more precisely measure business quality. However, such variables may be more difficult to acquire. Since tangible measures of consumer experience (such as cost and access) are not reliable predictors of ratings, variables should be constructed based on more intagible metrics. For instance, text-based analysis of Yelp reviews may identify keywords and sentiments in reviews for well-performing businesses. By omtting words that obviously indicate a favourable rating like "good" or "fun", qualitative characteristics for businesses with high ratings may be identifiable. Yelp has an immense amount of review data for most businesses that are listed on it, so there would be no issues with data shortages. Due to Yelp Fusion's daily call limit of 5000, it may take weeks to acquire all the data needed. As a result, a text-based analysis of reviews may generate more insightful results regarding the determinants of ratings.