**Overview**

Uber Eats is an online food ordering and delivery platform launched by Uber in 2014. Meals are delivered by couriers using cars, scooters, bicycles, or on foot. As of 2021, it operates in more than 45,000 cities. The delivery process is still handled by Uber drivers, who typically only take a few minutes to complete a service. Where Uber Eats is available, Uber will partner with multiple restaurants each day to serve meals to riders.

**Goals**

1. To analyze the geographic coverage of restaurants by selecting the state, city and street where they are located.

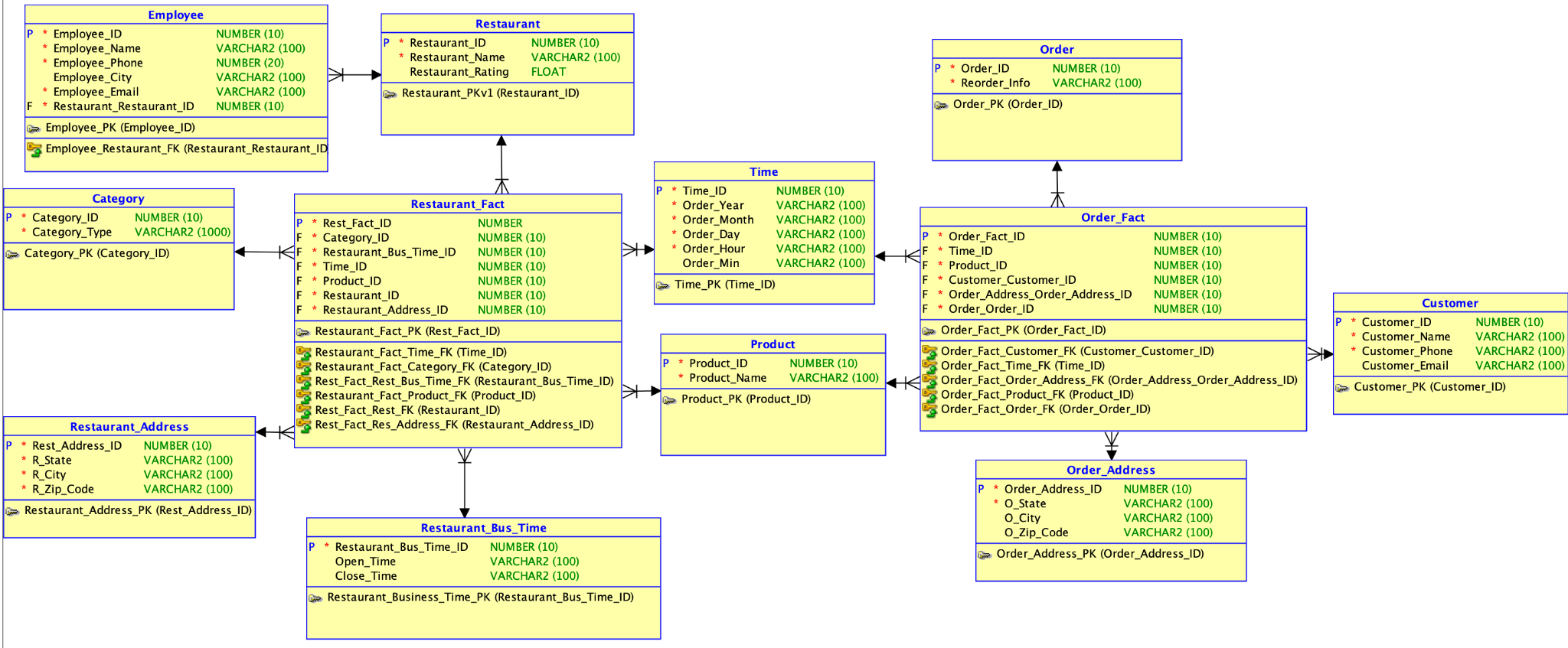
2. Analyze the impact of business hours on sales and profits.

3. Analyze the impact of individual product prices, average dish prices and highest (low) dish prices on merchant sales.

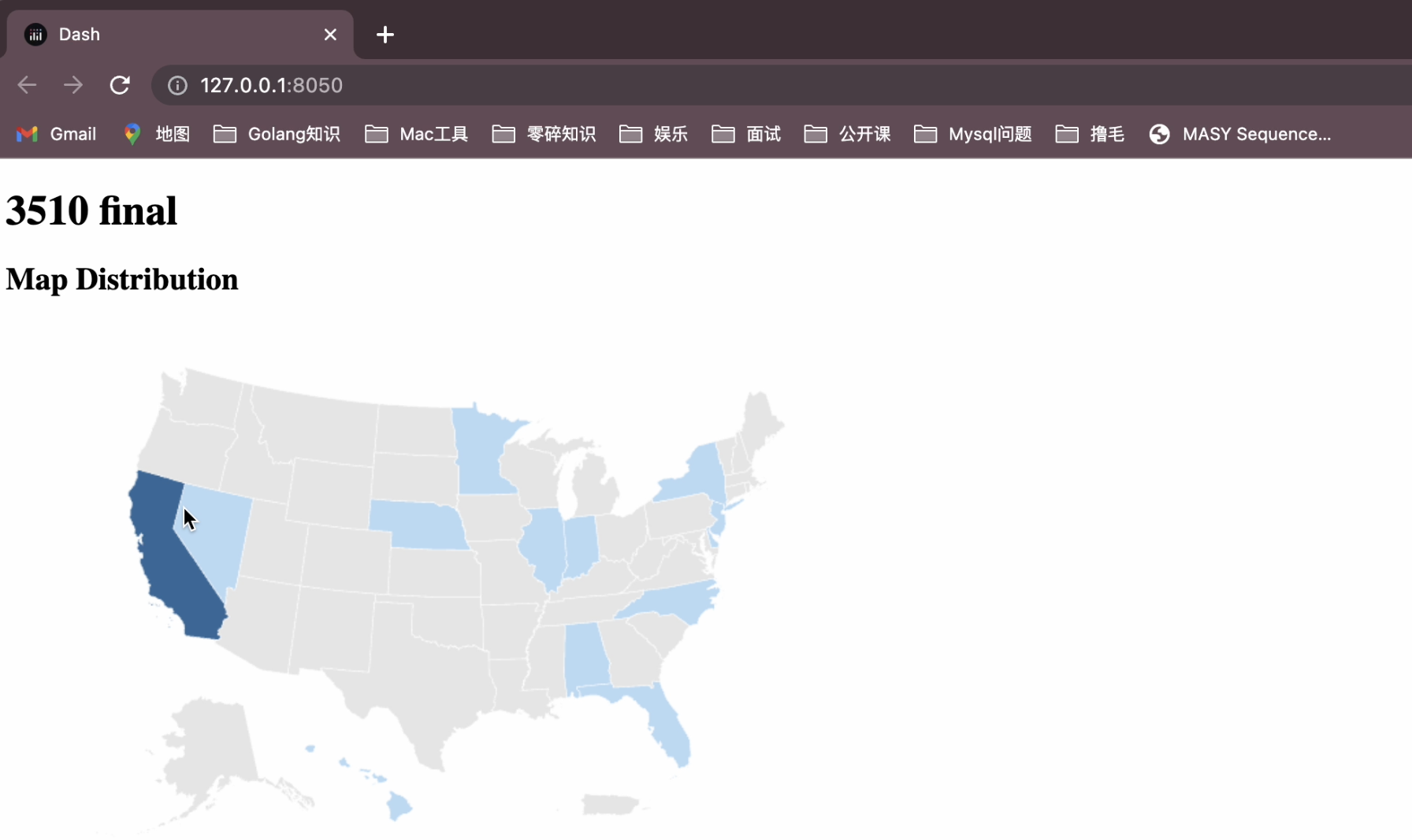
4. Analysis of the impact of recent transactions on sales.

5. Analyze the impact of positive feedback, store score and collection rate on sales.

**Data model**

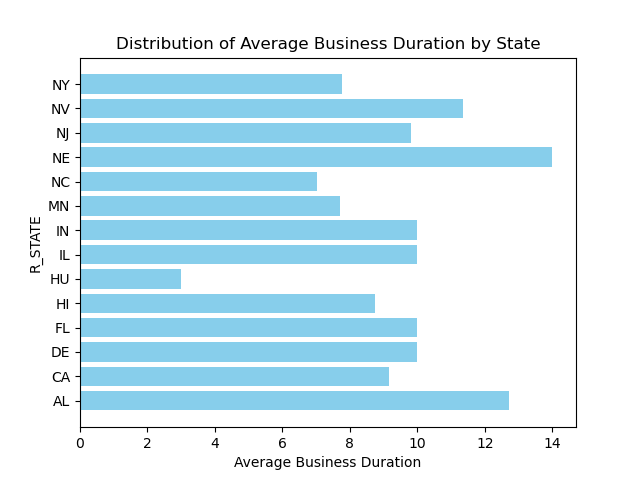
****

**Map Distribution**



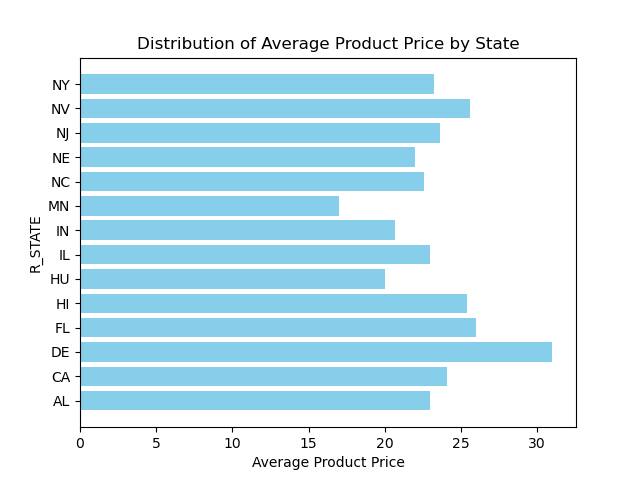
This graph shows us a visual representation of the distribution of restaurants that have partnered with Uber Eats across North America. The shading on the map indicates the concentration of Uber Eats partners, with darker regions indicating a higher number of restaurants that work with the popular food delivery service. California has the most restaurants partnering with Uber Eats.The company should increase the number of restaurants covering the central and northern regions.

**Average Business Hours**



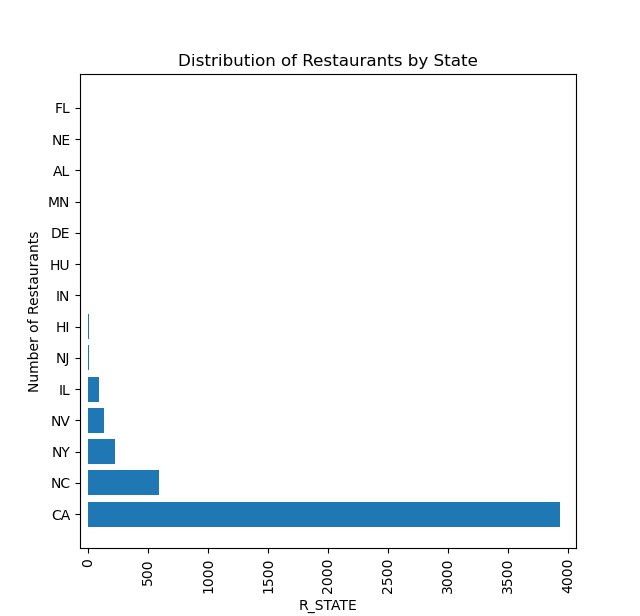
This graph shows the average business duration by state. We can see that NE has the longest business duration and HU has the shortest business duration. According to the data in the graph, restaurants in most areas are open for 10 hours or more. If new restaurants want to stay competitive, they should keep their business hours around 10 hours.

**Average Price**



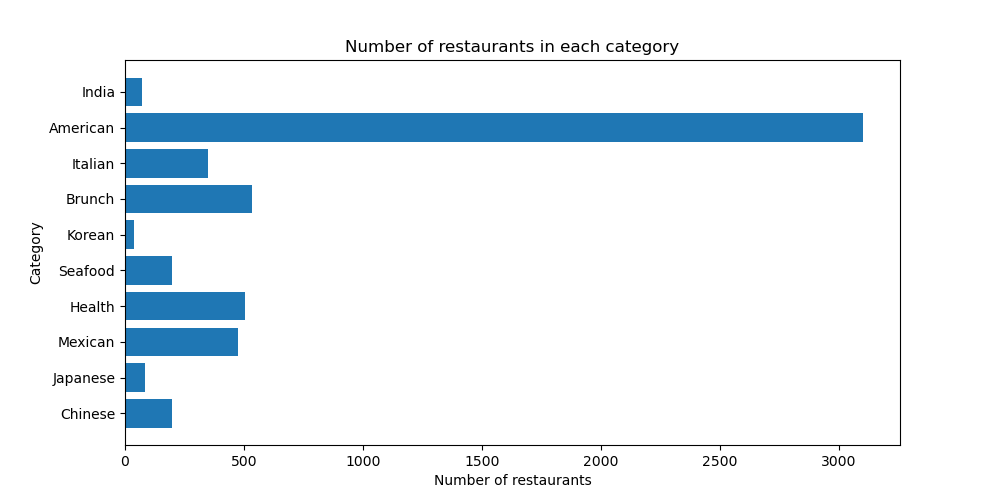
This chart shows the average product price by state. We can see that DE has the highest product price and MN has the lowest product price. On the whole, the avg price is mainly concentrated at 22. To open a restaurant in the east or west, the price should be in the range of 20-25 to be competitive.

**Distribution of Restaurant by State**



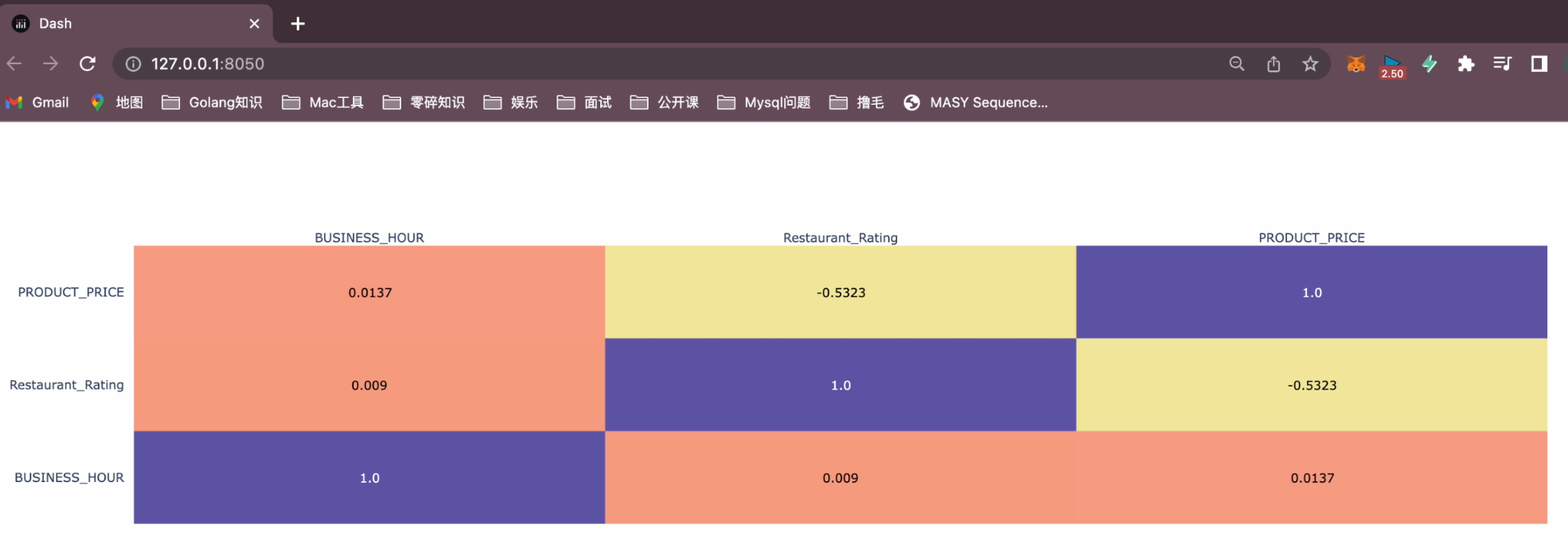
As shown above, we picked the top 5 states with the most restaurants as our further analysis. ( CA, NC, NY, NV, IL) In the later stage of development, Uber Eats should strengthen cooperation with restaurants in the central and northern regions, so as to achieve nationwide coverage of cooperative restaurants.

**Category Type**



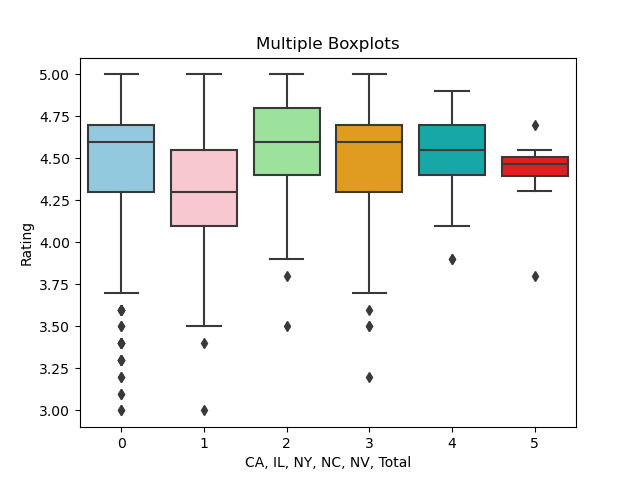
This graph shows the number of restaurants in each category. We can see that American restaurants have the largest number and widest coverage, followed by Brunch restaurants. The least number and least coverage are Korean restaurants.

**Correlation between rating,business\_hour and product price**

****

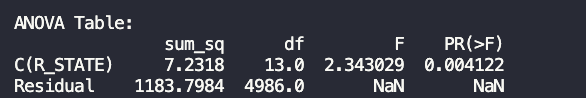
This heat map represents the correlation of the three variables business\_hour,restaurant\_rating, and product\_price.From the graph, we can conclude that the absolute value of the correlation coefficient between business\_hour and restaurant\_rating, business\_hour and product\_price is less than 0.3, and the correlation is very weak, and we can almost consider that they are not correlated. product\_price and The absolute value of the correlation coefficient between restaurant\_rating is 0.5323, when 0.5<|correlation coefficient|<0.8, it means that product\_price has a significant linear relationship with restaurant\_rating. Since product\_price and restaurant\_rating are negatively correlated, we can learn that the lower the product\_price, the higher the restaurant\_rating.

**BOXPLOT ANALYSIS**

****

Medians vary widely, with nc and ca having the highest medians. The degree of dispersion varies, where nv, as a high rating top5 state, has a more concentrated rating, compared to il, which has the most discrete data. By looking at the following outliers, it can be seen that ca has the most outliers and has more anomalies. The outliers are concentrated on the side of smaller values, the distribution is left-skewed, and the median is skewed toward the upper quartile, indicating that there is a problem of pulling down the outliers. We used the overall rating as a reference value and selected a median of 4.47 as the criterion for a high rating, we therefore conjectured that location was correlated with rating and therefore performed a follow-up.

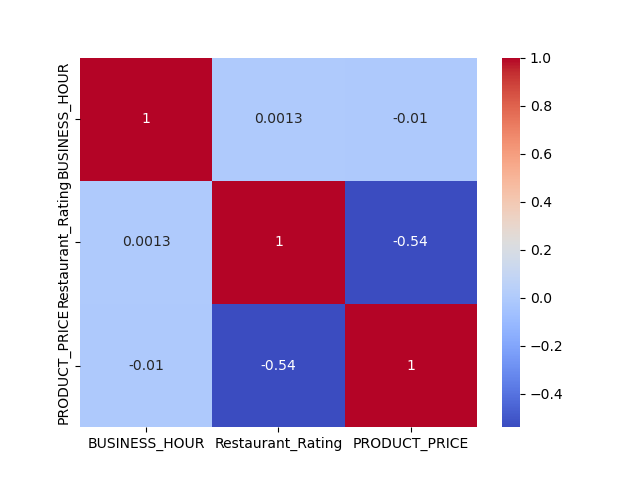
**Anova Table**



The F of the ANOVA table and the factor position can be obtained by the program as 2.343029. The pr(>f) in the ANOVA table shows that the p-value is 0.004122. From the anova analysis, it is known that the p-value obtained is less than 0.05, which means that the effect of the factor position is significant. If it is less than 0.01, it means that the effect of factor position is highly significant. Based on the p-value obtained, it is known that state is highly significant than restaurant rating.

This analysis can prove that restaurant location is related to high ratings, so if uber eats wants to improve customer satisfaction, then it should deploy restaurants in developed areas.

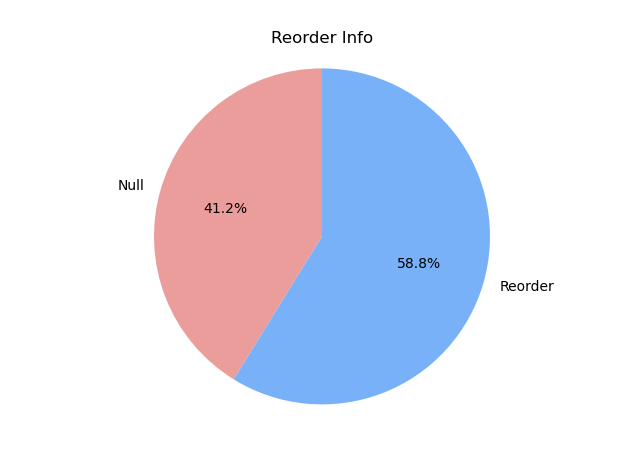
**Correlation between rating, business\_hour and product price**



This graph is the correlation graph. The absolute value of the correlation coefficient between product\_price and restaurant rating can be found in the picture is 0.54, when 0.5<|correlation coefficient|<0.8, it means that product\_price has a significant linear relationship with restaurant rating. According to the negative correlation, we can conclude that the lower the price, the higher the rating.

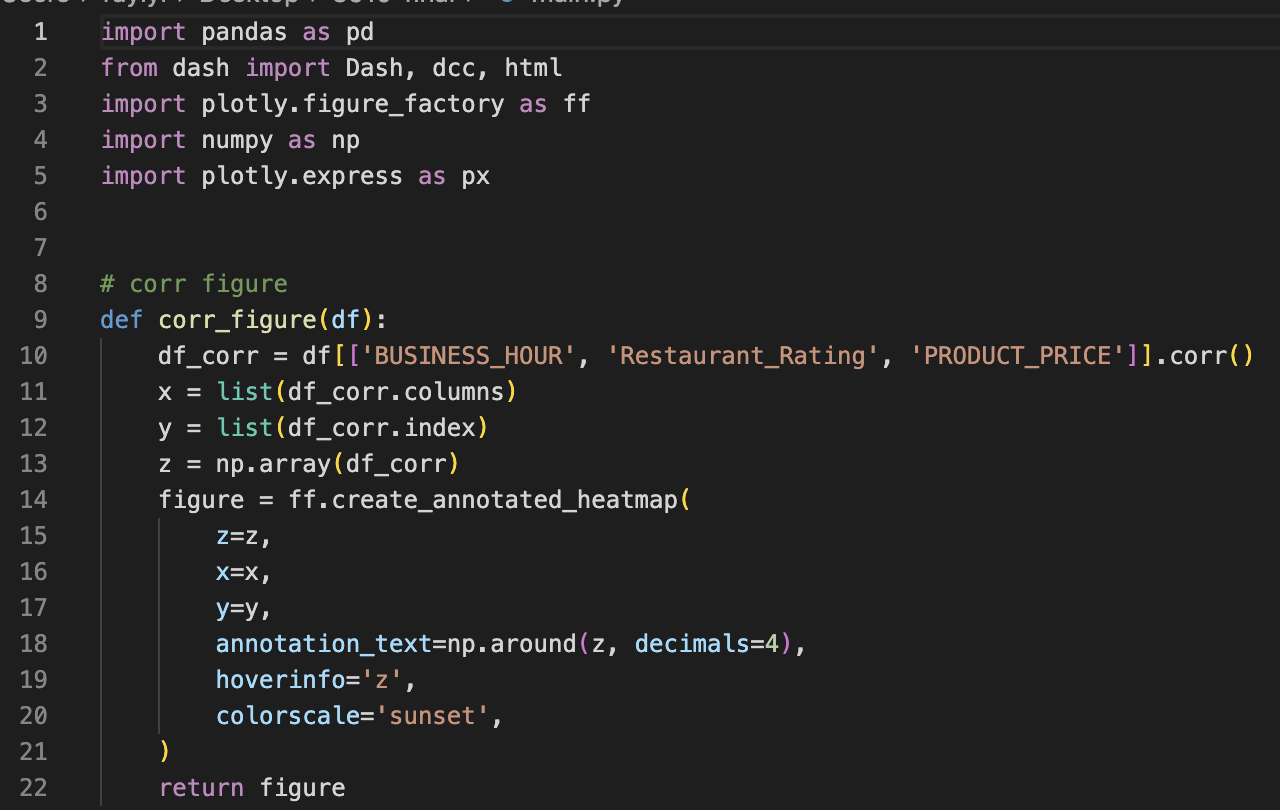
Therefore, to get high satisfaction, you can reduce the product price appropriately.

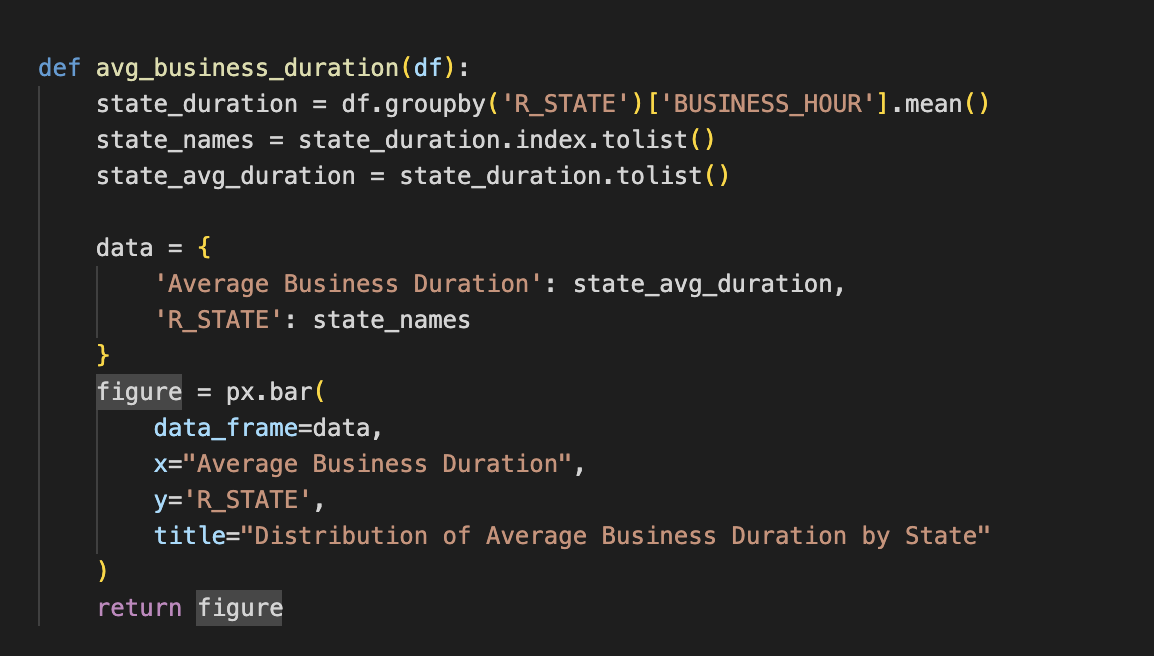
**REORDER\_RATE**

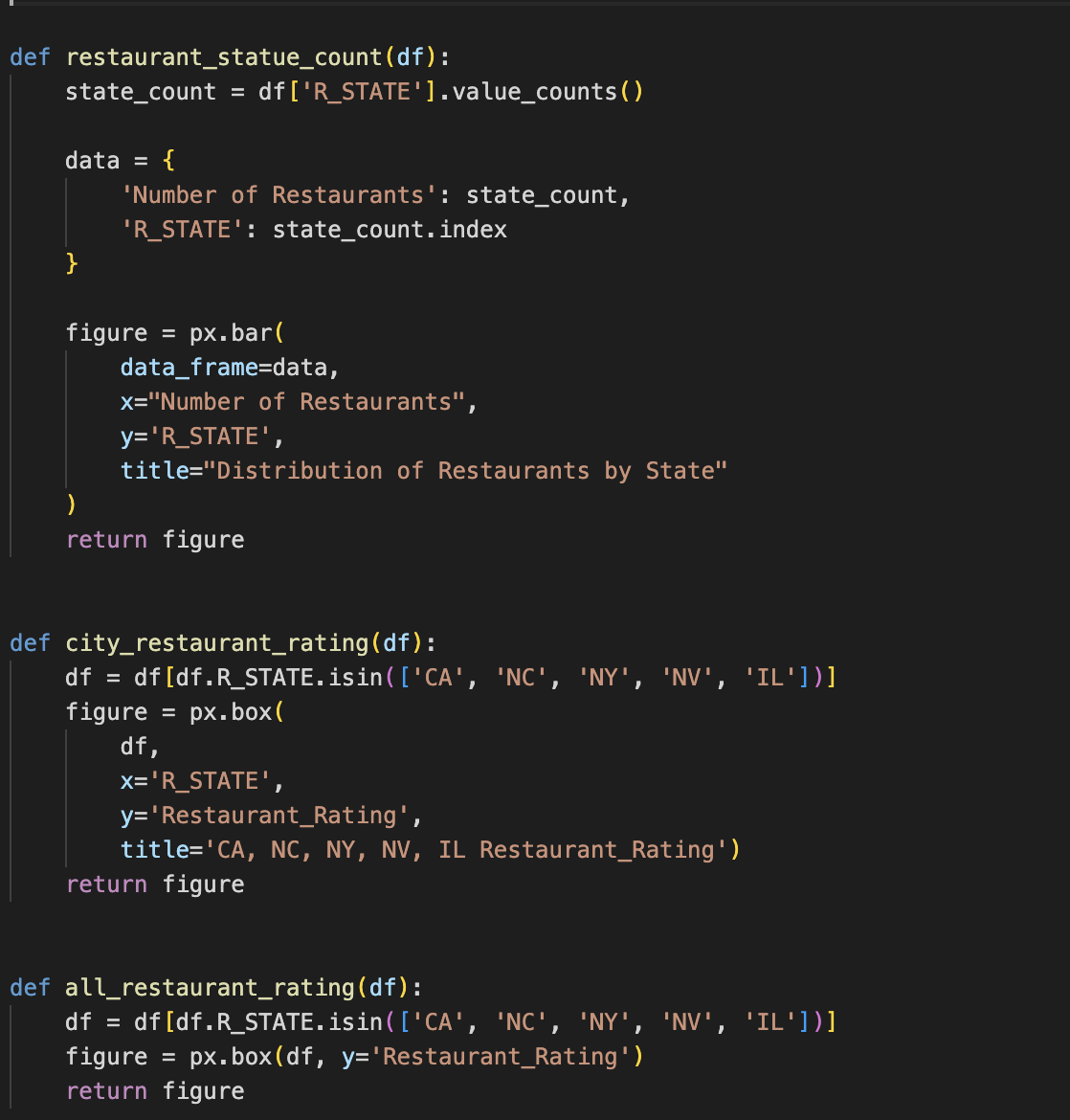


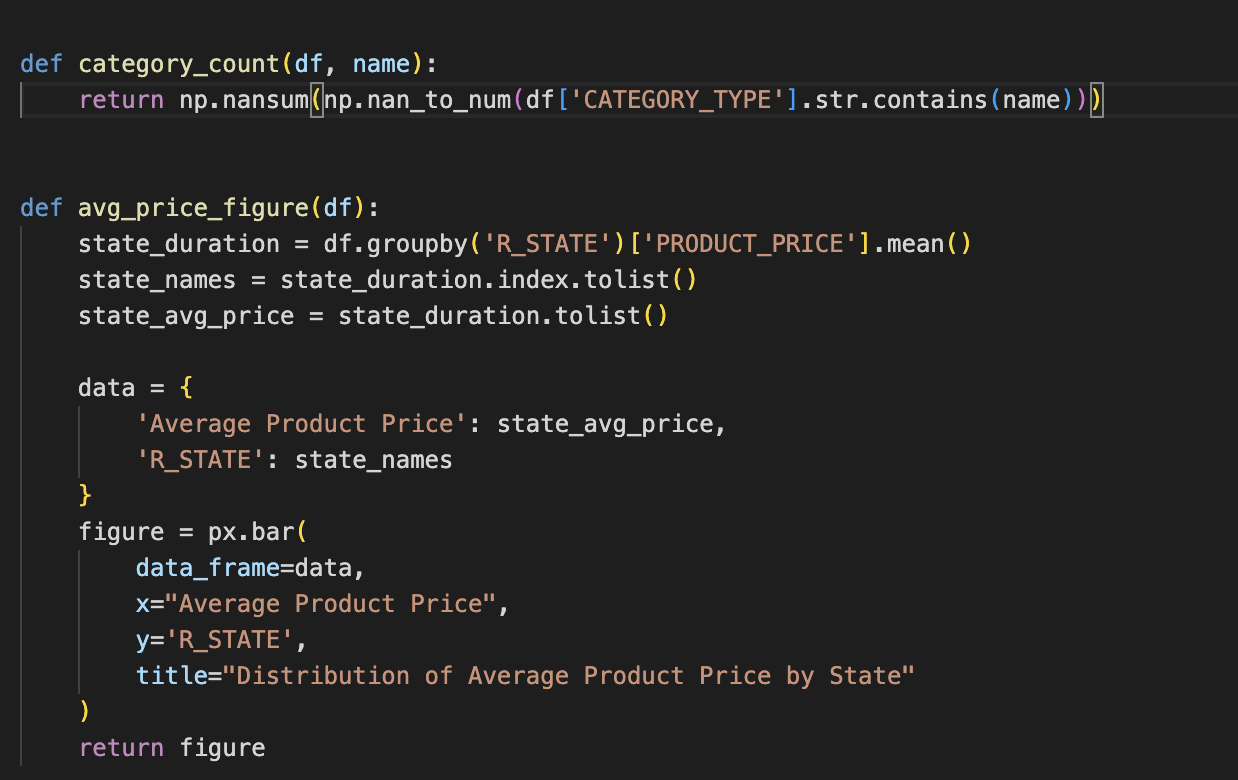
This graph shows the reorder information. This graph shows what percentage of people are willing to order again with Uber Eats. We can see that 58.8% of people like to reorder with Uber Eats. Although the proportion of reorders is relatively high, the company should strengthen customer relationship management, maintain close contact with customers, understand their needs and feedback, solve problems in a timely manner, and improve customer satisfaction in the future development.

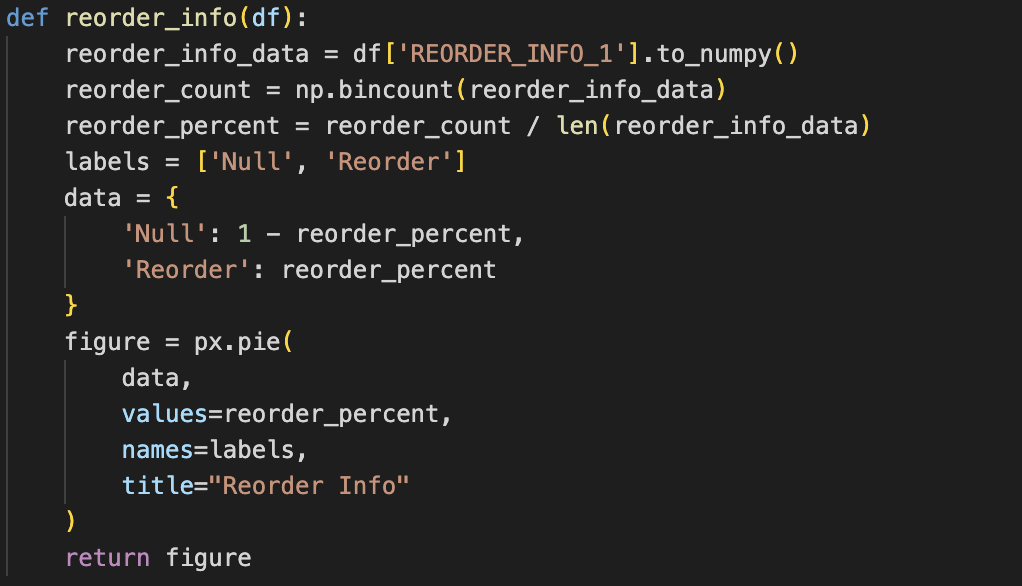
**Dashboard code:**

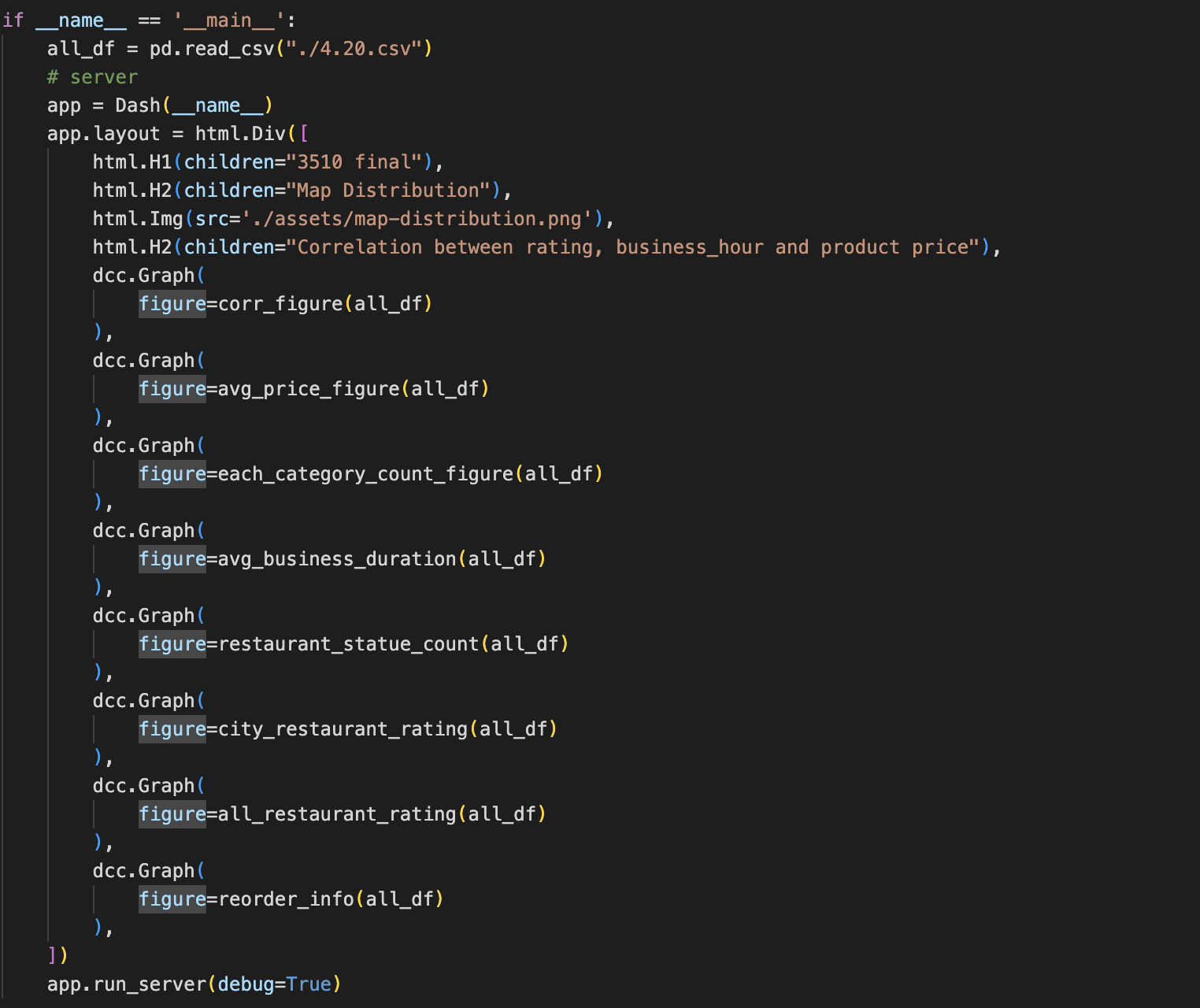
****

****

****

****

****

****

Analysis Code：

import pandas as pd

import matplotlib.pyplot as plt

df = pd.read\_csv('4.20.csv')

# 所有的州的avg

avg\_by\_state = round(df.groupby('R\_STATE')['Restaurant\_Rating'].mean(),1)

sorted\_df = avg\_by\_state.sort\_values(*ascending*=False)

print(sorted\_df)

#top 5 ：MN, NY, CA, NC, HV

# # ## 证明大于95%的概率可以说明 高评分与地区有关系 证明到底评分的分类正确

from statsmodels.formula.api import ols

from statsmodels.stats.anova import anova\_lm

grouped = df.groupby('R\_STATE')

means = grouped['boo'].mean()

# ANOVA检验

import statsmodels.api as sm

from statsmodels.formula.api import ols

from statsmodels.stats.anova import anova\_lm

formula = 'boo ~ C(R\_STATE)'

model = ols(formula, *data*=df).fit()

aov\_table = anova\_lm(model, *typ*=2)

print('Means:')

print(means)

print('')

print('ANOVA Table:')

print(aov\_table)

#相关系数corr

import seaborn as sns

import matplotlib.pyplot as plt

corr = df[['BUSINESS\_HOUR', 'Restaurant\_Rating', 'PRODUCT\_PRICE']].corr()

sns.heatmap(corr, *annot*=True, *cmap*='coolwarm')

plt.show()

# 散点图

plt.scatter(df['NY\_STATE'], df['Restaurant\_Rating'])

plt.title('Restaurant\_Rating vs. NY\_STATE')

plt.xlabel('NY\_STATE')

plt.ylabel('Restaurant\_Rating')

plt.show()

import csv

import seaborn as sns

import matplotlib.pyplot as plt

rating1 = []

rating2 = []

rating3 = []

rating4 = []

rating5 = []

rating6 = []

with open('4.20.csv', 'r') as f:

reader = csv.reader(f)

for col in reader:

if col[7] == 'CA':

rating\_str = col[2]

if rating\_str != '':

rating1.append(float(rating\_str))

elif col[7] == 'IL':

rating\_str = col[2]

if rating\_str != '':

rating2.append(float(rating\_str))

elif col[7] == 'NY':

rating\_str = col[2]

if rating\_str != '':

rating3.append(float(rating\_str))

elif col[7] == 'NC':

rating\_str = col[2]

if rating\_str != '':

rating4.append(float(rating\_str))

elif col[7] == 'NV':

rating\_str = col[2]

if rating\_str != '':

rating5.append(float(rating\_str))

grouped = df.groupby('R\_STATE')['Restaurant\_Rating'].mean()

# total

fig, ax = plt.subplots()

sns.boxplot(*data*=[rating1, rating2, rating3, rating4, rating5, rating6], *ax*=ax, *palette*=['skyblue', 'pink', 'lightgreen', 'orange', 'c','r'])

ax.set(*title*='Multiple Boxplots', *xlabel*='CA, IL, NY, NC, NV',*ylabel*='Rating')

plt.show()

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

data = pd.read\_csv('4.20.csv')

grouped = data.groupby('R\_STATE')['Restaurant\_Rating'].mean()

fig, ax = plt.subplots()

sns.boxplot(*data*=[data[data['R\_STATE'] == 'CA']['Restaurant\_Rating'].dropna(),

data[data['R\_STATE'] == 'IL']['Restaurant\_Rating'].dropna(),

data[data['R\_STATE'] == 'NY']['Restaurant\_Rating'].dropna(),

data[data['R\_STATE'] == 'NC']['Restaurant\_Rating'].dropna(),

data[data['R\_STATE'] == 'NV']['Restaurant\_Rating'].dropna(),

grouped.values],

*ax*=ax,

*palette*=['skyblue', 'pink', 'lightgreen', 'orange', 'c', 'r'])

ax.set(*title*='Multiple Boxplots', *xlabel*='CA, IL, NY, NC, NV, Total', *ylabel*='Rating')

plt.show()

# COUNT CATEGORY

import numpy as np

chinese = np.nansum(np.nan\_to\_num(df['CATEGORY\_TYPE'].str.contains('Chinese')))

japanese = np.nansum(np.nan\_to\_num(df['CATEGORY\_TYPE'].str.contains('Japanese')))

mexican = np.nansum(np.nan\_to\_num(df['CATEGORY\_TYPE'].str.contains('Mexican')))

health = np.nansum(np.nan\_to\_num(df['CATEGORY\_TYPE'].str.contains('Health')))

seafood = np.nansum(np.nan\_to\_num(df['CATEGORY\_TYPE'].str.contains('Seafood')))

korean = np.nansum(np.nan\_to\_num(df['CATEGORY\_TYPE'].str.contains('Korean')))

brunch = np.nansum(np.nan\_to\_num(df['CATEGORY\_TYPE'].str.contains('Brunch')))

italian = np.nansum(np.nan\_to\_num(df['CATEGORY\_TYPE'].str.contains('Italian')))

american = np.nansum(np.nan\_to\_num(df['CATEGORY\_TYPE'].str.contains('American')))

india = np.nansum(np.nan\_to\_num(df['CATEGORY\_TYPE'].str.contains('India')))

print(f"chinese: {chinese}")

print(f"japanese: {japanese}")

print(f"mexican: {mexican}")

print(f"health: {health}")

print(f"seafood: {seafood}")

print(f"korean: {korean}")

print(f"brunch: {brunch}")

print(f"italian: {italian}")

print(f"american: {american}")

print(f"india: {india}")

import matplotlib.pyplot as plt

counts = [chinese, japanese, mexican, health, seafood, korean, brunch, italian, american, india]

categories = ['Chinese', 'Japanese', 'Mexican', 'Health', 'Seafood', 'Korean', 'Brunch', 'Italian', 'American', 'India']

fig, ax = plt.subplots(*figsize*=(10,5))

ax.barh(categories, counts)

ax.set\_xlabel('Number of restaurants')

ax.set\_ylabel('Category')

ax.set\_title('Number of restaurants in each category')

plt.show()

# reorder info rate

import numpy as np

import matplotlib.pyplot as plt

reorder\_info = df['REORDER\_INFO\_1'].to\_numpy()

reorder\_count = np.bincount(reorder\_info)

reorder\_percent = reorder\_count / len(reorder\_info)

labels = ['Null', 'Reorder']

colors = ['#ff9999', '#66b3ff']

fig1, ax1 = plt.subplots()

ax1.pie(reorder\_percent, *colors*=colors, *labels*=labels, *autopct*='%1.1f%%', *startangle*=90)

ax1.axis('equal')

plt.title('Reorder Info')

plt.show()

#不同州business的直方图：

import matplotlib.pyplot as plt

state\_duration = df.groupby('R\_STATE')['BUSINESS\_HOUR'].mean()

state\_names = state\_duration.index.tolist()

state\_avg\_duration = state\_duration.tolist()

plt.barh(state\_names, state\_avg\_duration, *color*='skyblue')

plt.xlabel('Average Business Duration')

plt.ylabel('R\_STATE')

plt.title('Distribution of Average Business Duration by State')

plt.show()

不同州的avg price

import matplotlib.pyplot as plt

state\_duration = df.groupby('R\_STATE')['PRODUCT\_PRICE'].mean()

state\_names = state\_duration.index.tolist()

state\_avg\_duration = state\_duration.tolist()

plt.barh(state\_names, state\_avg\_duration, *color*='skyblue')

plt.xlabel('Average Product Price')

plt.ylabel('R\_STATE')

plt.title('Distribution of Average Product Price by State')

plt.show()

# COUNT STATE

import matplotlib.pyplot as plt

state\_count = df['R\_STATE'].value\_counts()

plt.barh(state\_count.index, state\_count.values)

plt.xlabel('R\_STATE')

plt.ylabel('Number of Restaurants')

plt.title('Distribution of Restaurants by State')

plt.xticks(*rotation*=90)

plt.show()

Model：

-- Generated by Oracle SQL Developer Data Modeler 22.2.0.165.1149

-- at: 2023-04-10 01:27:30 EDT

-- site: Oracle Database 11g

-- type: Oracle Database 11g

-- predefined type, no DDL - MDSYS.SDO\_GEOMETRY

-- predefined type, no DDL - XMLTYPE

CREATE TABLE category (

category\_id NUMBER(8) NOT NULL,

category\_type VARCHAR2(20)

);

ALTER TABLE category ADD CONSTRAINT category\_pk PRIMARY KEY ( category\_id );

CREATE TABLE customer (

customer\_id NUMBER(8) NOT NULL,

customer\_name VARCHAR2(20) NOT NULL,

customer\_phone NUMBER(10) NOT NULL,

customer\_email VARCHAR2(20) NOT NULL

);

ALTER TABLE customer ADD CONSTRAINT customer\_pk PRIMARY KEY ( customer\_id );

CREATE TABLE employee (

employee\_id NUMBER(4) NOT NULL,

employee\_name VARCHAR2(30) NOT NULL,

employee\_phone NUMBER(10) NOT NULL,

employee\_address VARCHAR2(50),

employee\_email VARCHAR2(30),

restaurant\_restaurant\_id NUMBER(8)

);

ALTER TABLE employee ADD CONSTRAINT employee\_pk PRIMARY KEY ( employee\_id );

CREATE TABLE "Order" (

order\_id NUMBER(8) NOT NULL,

reorder\_info VARCHAR2(20)

);

ALTER TABLE "Order" ADD CONSTRAINT order\_pk PRIMARY KEY ( order\_id );

CREATE TABLE order\_address (

order\_address\_id NUMBER(8) NOT NULL,

o\_state VARCHAR2(20) NOT NULL,

o\_city VARCHAR2(20),

o\_zip\_code VARCHAR2(20)

);

ALTER TABLE order\_address ADD CONSTRAINT order\_address\_pk PRIMARY KEY ( order\_address\_id );

CREATE TABLE order\_fact (

order\_fact\_id NUMBER(8) NOT NULL,

time\_id NUMBER(8) NOT NULL,

product\_id NUMBER(8) NOT NULL,

order\_order\_id NUMBER(8) NOT NULL,

customer\_customer\_id NUMBER(8) NOT NULL,

order\_address\_order\_address\_id NUMBER(8) NOT NULL

);

ALTER TABLE order\_fact ADD CONSTRAINT order\_fact\_pk PRIMARY KEY ( order\_fact\_id );

CREATE TABLE product (

product\_id NUMBER(8) NOT NULL,

product\_name VARCHAR2(20) NOT NULL

);

ALTER TABLE product ADD CONSTRAINT product\_pk PRIMARY KEY ( product\_id );

CREATE TABLE restaurant (

restaurant\_id NUMBER(8) NOT NULL,

restaurant\_name VARCHAR2(30) NOT NULL,

restaurant\_rating FLOAT(2) NOT NULL

);

ALTER TABLE restaurant ADD CONSTRAINT restaurant\_pk PRIMARY KEY ( restaurant\_id );

CREATE TABLE restaurant\_business\_time (

restaurant\_business\_time\_id NUMBER(8) NOT NULL,

open\_time VARCHAR2(10) NOT NULL,

close\_time VARCHAR2(10) NOT NULL

);

ALTER TABLE restaurant\_business\_time ADD CONSTRAINT restaurant\_business\_time\_pk PRIMARY KEY ( restaurant\_business\_time\_id );

CREATE TABLE resturant\_address (

rest\_address\_id NUMBER(8) NOT NULL,

o\_state VARCHAR2(20) NOT NULL,

o\_city VARCHAR2(20),

o\_zip\_code NUMBER(5)

);

ALTER TABLE resturant\_address ADD CONSTRAINT resturant\_address\_pk PRIMARY KEY ( rest\_address\_id );

CREATE TABLE resturant\_fact (

rest\_fact\_id NUMBER(8) NOT NULL,

category\_id NUMBER(8) NOT NULL,

resturant\_address\_id NUMBER(8) NOT NULL,

restaurant\_bus\_time\_id NUMBER(8) NOT NULL,

time\_id NUMBER(8) NOT NULL,

product\_id NUMBER(8) NOT NULL,

restaurant\_id NUMBER(8) NOT NULL

);

ALTER TABLE resturant\_fact ADD CONSTRAINT resturant\_fact\_pk PRIMARY KEY ( rest\_fact\_id );

CREATE TABLE time (

time\_id NUMBER(8) NOT NULL,

oder\_year VARCHAR2(10) NOT NULL,

order\_month VARCHAR2(10) NOT NULL,

order\_day VARCHAR2(6) NOT NULL,

order\_hour NUMBER(2) NOT NULL,

order\_min NUMBER(2)

);

ALTER TABLE time ADD CONSTRAINT time\_pk PRIMARY KEY ( time\_id );

ALTER TABLE employee

ADD CONSTRAINT employee\_restaurant\_fk FOREIGN KEY ( restaurant\_restaurant\_id )

REFERENCES restaurant ( restaurant\_id );

ALTER TABLE order\_fact

ADD CONSTRAINT order\_fact\_customer\_fk FOREIGN KEY ( customer\_customer\_id )

REFERENCES customer ( customer\_id );

ALTER TABLE order\_fact

ADD CONSTRAINT order\_fact\_order\_address\_fk FOREIGN KEY ( order\_address\_order\_address\_id )

REFERENCES order\_address ( order\_address\_id );

ALTER TABLE order\_fact

ADD CONSTRAINT order\_fact\_order\_fk FOREIGN KEY ( order\_order\_id )

REFERENCES "Order" ( order\_id );

ALTER TABLE order\_fact

ADD CONSTRAINT order\_fact\_product\_fk FOREIGN KEY ( product\_id )

REFERENCES product ( product\_id );

ALTER TABLE order\_fact

ADD CONSTRAINT order\_fact\_time\_fk FOREIGN KEY ( time\_id )

REFERENCES time ( time\_id );

ALTER TABLE resturant\_fact

ADD CONSTRAINT rest\_fact\_rest\_address\_fk FOREIGN KEY ( resturant\_address\_id )

REFERENCES resturant\_address ( rest\_address\_id );

ALTER TABLE resturant\_fact

ADD CONSTRAINT resturant\_fact\_category\_fk FOREIGN KEY ( category\_id )

REFERENCES category ( category\_id );

ALTER TABLE resturant\_fact

ADD CONSTRAINT resturant\_fact\_fk FOREIGN KEY ( restaurant\_id )

REFERENCES restaurant ( restaurant\_id );

ALTER TABLE resturant\_fact

ADD CONSTRAINT resturant\_fact\_product\_fk FOREIGN KEY ( product\_id )

REFERENCES product ( product\_id );

ALTER TABLE resturant\_fact

ADD CONSTRAINT resturant\_fact\_time\_fk FOREIGN KEY ( time\_id )

REFERENCES time ( time\_id );

ALTER TABLE resturant\_fact

ADD CONSTRAINT resturant\_fact\_time\_fkv1 FOREIGN KEY ( restaurant\_bus\_time\_id )

REFERENCES restaurant\_business\_time ( restaurant\_business\_time\_id );

-- Oracle SQL Developer Data Modeler Summary Report:

--

-- CREATE TABLE 12

-- CREATE INDEX 0

-- ALTER TABLE 24

-- CREATE VIEW 0

-- ALTER VIEW 0

-- CREATE PACKAGE 0

-- CREATE PACKAGE BODY 0

-- CREATE PROCEDURE 0

-- CREATE FUNCTION 0

-- CREATE TRIGGER 0

-- ALTER TRIGGER 0

-- CREATE COLLECTION TYPE 0

-- CREATE STRUCTURED TYPE 0

-- CREATE STRUCTURED TYPE BODY 0

-- CREATE CLUSTER 0

-- CREATE CONTEXT 0

-- CREATE DATABASE 0

-- CREATE DIMENSION 0

-- CREATE DIRECTORY 0

-- CREATE DISK GROUP 0

-- CREATE ROLE 0

-- CREATE ROLLBACK SEGMENT 0

-- CREATE SEQUENCE 0

-- CREATE MATERIALIZED VIEW 0

-- CREATE MATERIALIZED VIEW LOG 0

-- CREATE SYNONYM 0

-- CREATE TABLESPACE 0

-- CREATE USER 0

--

-- DROP TABLESPACE 0

-- DROP DATABASE 0

--

-- REDACTION POLICY 0

--

-- ORDS DROP SCHEMA 0

-- ORDS ENABLE SCHEMA 0

-- ORDS ENABLE OBJECT 0

--

-- ERRORS 0

-- WARNINGS 0