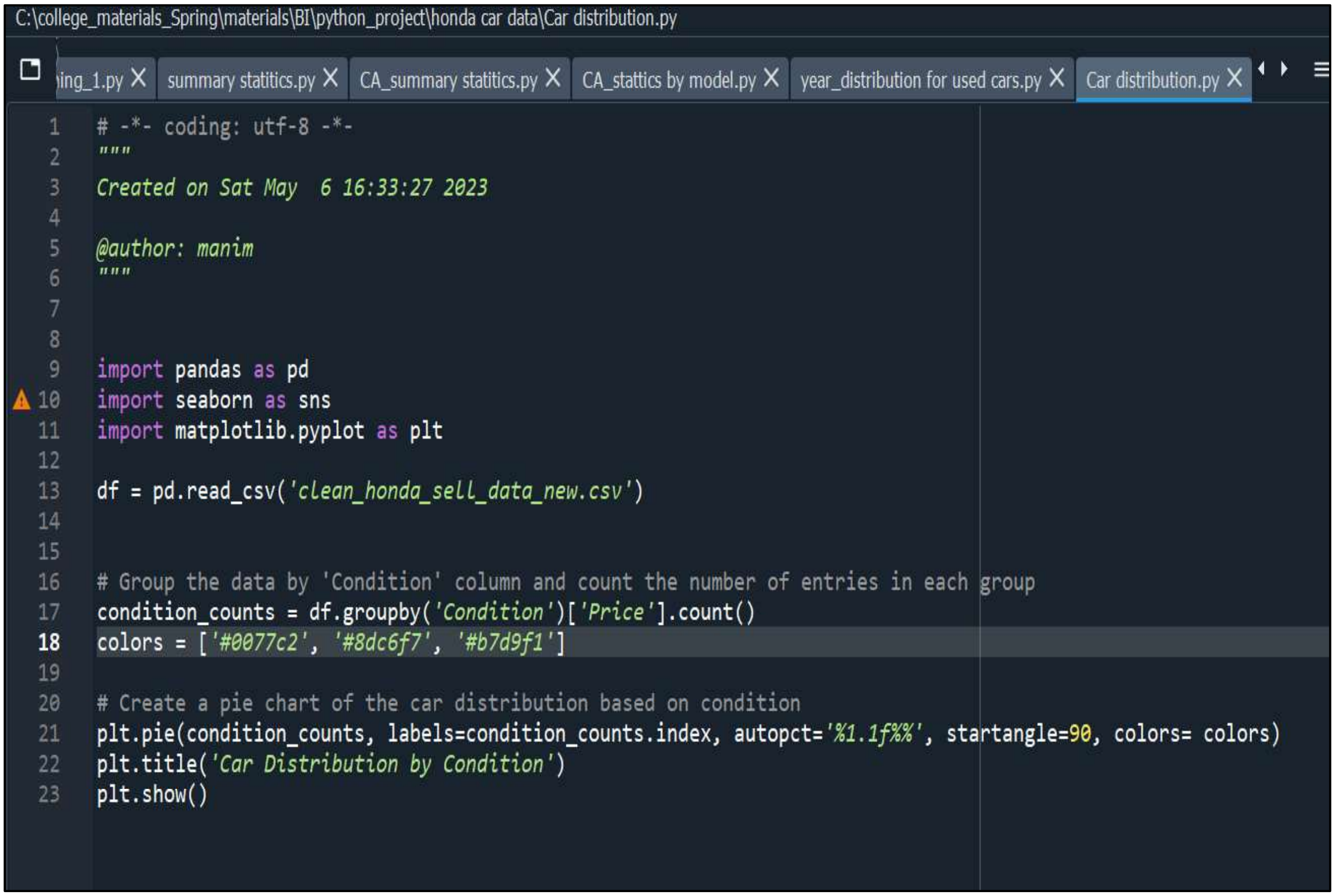
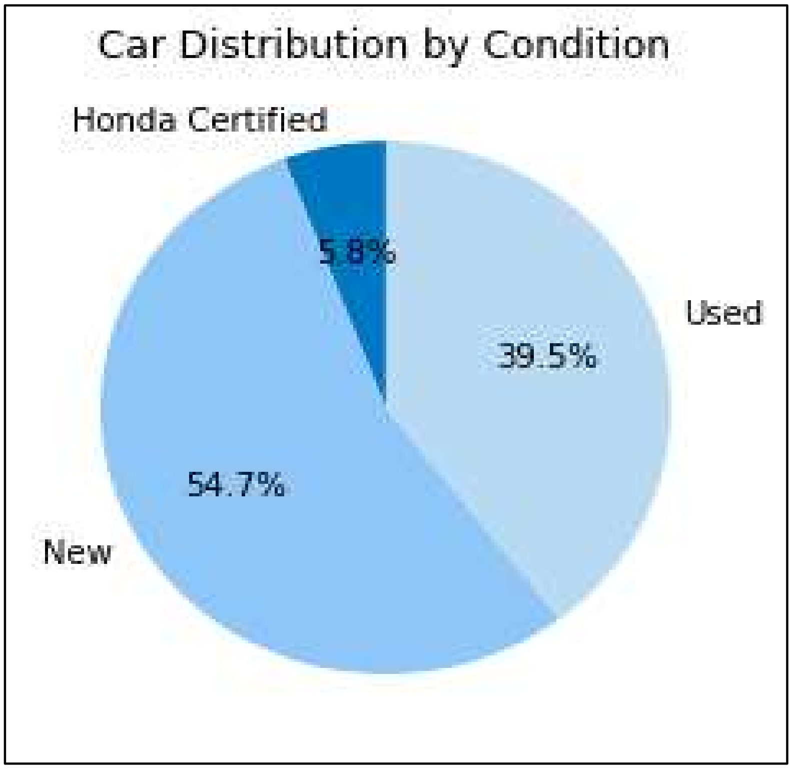
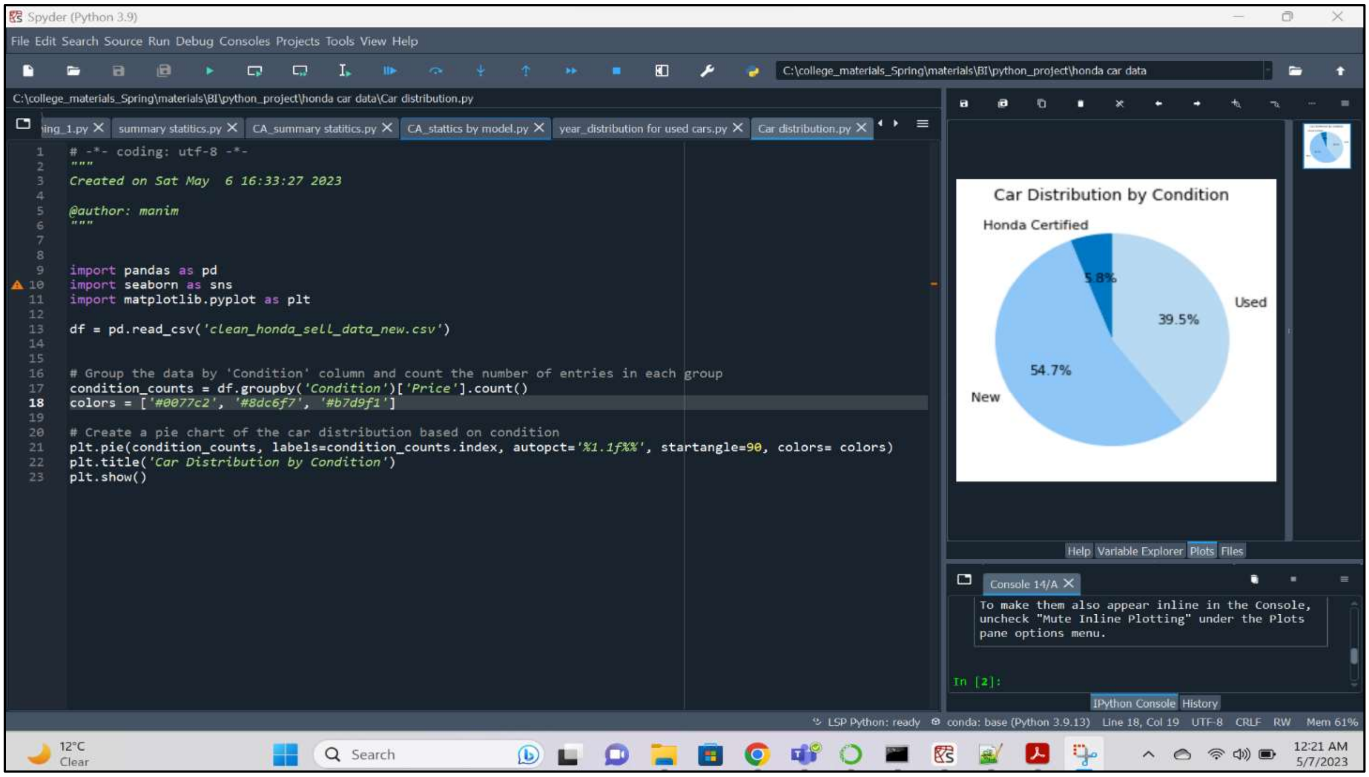
E. Visualizations

1. Show the car distribution based on condition (New, used, certified)


C:\college_materials_Spring\materials\BI\python_project\honda car data\Car distribution.py
ing_1.py
summary statitics.py
CA_summary statitics.py
CA_stattics by model.py
year_distribution for used cars.py
Car distribution.py

```python
# -*- coding: utf-8 -*-
"""
Created on Sat May  6 16:33:27 2023

@author: manim
"""


import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

df = pd.read_csv('clean_honda_sell_data_new.csv')


# Group the data by 'Condition' column and count the number of entries in each group
condition_counts = df.groupby('Condition')['Price'].count()
colors = ['#0077c2', '#8dc6f7', '#b7d9f1']

# Create a pie chart of the car distribution based on condition
plt.pie(condition_counts, labels=condition_counts.index, autopct='%1.1f%%', startangle=90, colors= colors)
plt.title('Car Distribution by Condition')
plt.show()
```




Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\college_materials_Spring\materials\BI\python_project\honda car data
C:\college_materials_Spring\materials\BI\python_project\honda car data\Car distribution.py
ing_1.py
summary statitics.py
CA_summary statitics.py
CA_stattics by model.py
year_distribution for used cars.py
Car distribution.py
Car Distribution by Condition
Honda Certified
5.8%
Used
39.5%
54.7%
New
Help Variable Explorer Plots Files
Console 14/A
IPython Console History
LSP Python: ready conda: base (Python 3.9.13) Line 18, Col 19 UTF-8 CRLF RW Mem 61%
12°C Clear
Search
12:21 AM 5/7/2023

```python
# -*- coding: utf-8 -*-
"""
Created on Sat May  6 16:33:27 2023

@author: manim
"""


import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

df = pd.read_csv('clean_honda_sell_data_new.csv')


# Group the data by 'Condition' column and count the number of entries in each group
condition_counts = df.groupby('Condition')['Price'].count()
colors = ['#0077c2', '#8dc6f7', '#b7d9f1']

# Create a pie chart of the car distribution based on condition
plt.pie(condition_counts, labels=condition_counts.index, autopct='%1.1f%%', startangle=90, colors= colors)
plt.title('Car Distribution by Condition')
plt.show()
```

```
To make them also appear inline in the Console,
uncheck "Mute Inline Plotting" under the Plots
pane options menu.

In [2]:
```

Insights:

The analysis provides a distribution of cars based on their condition. The three conditions being considered are 'New', 'Used', and 'Honda Certified'.

Most of the cars in the dataset are in the 'New' condition, accounting for 54.7% of the total cars. 'Used' cars account for 39.5%, and 'Honda Certified ' cars account for 5.8%.

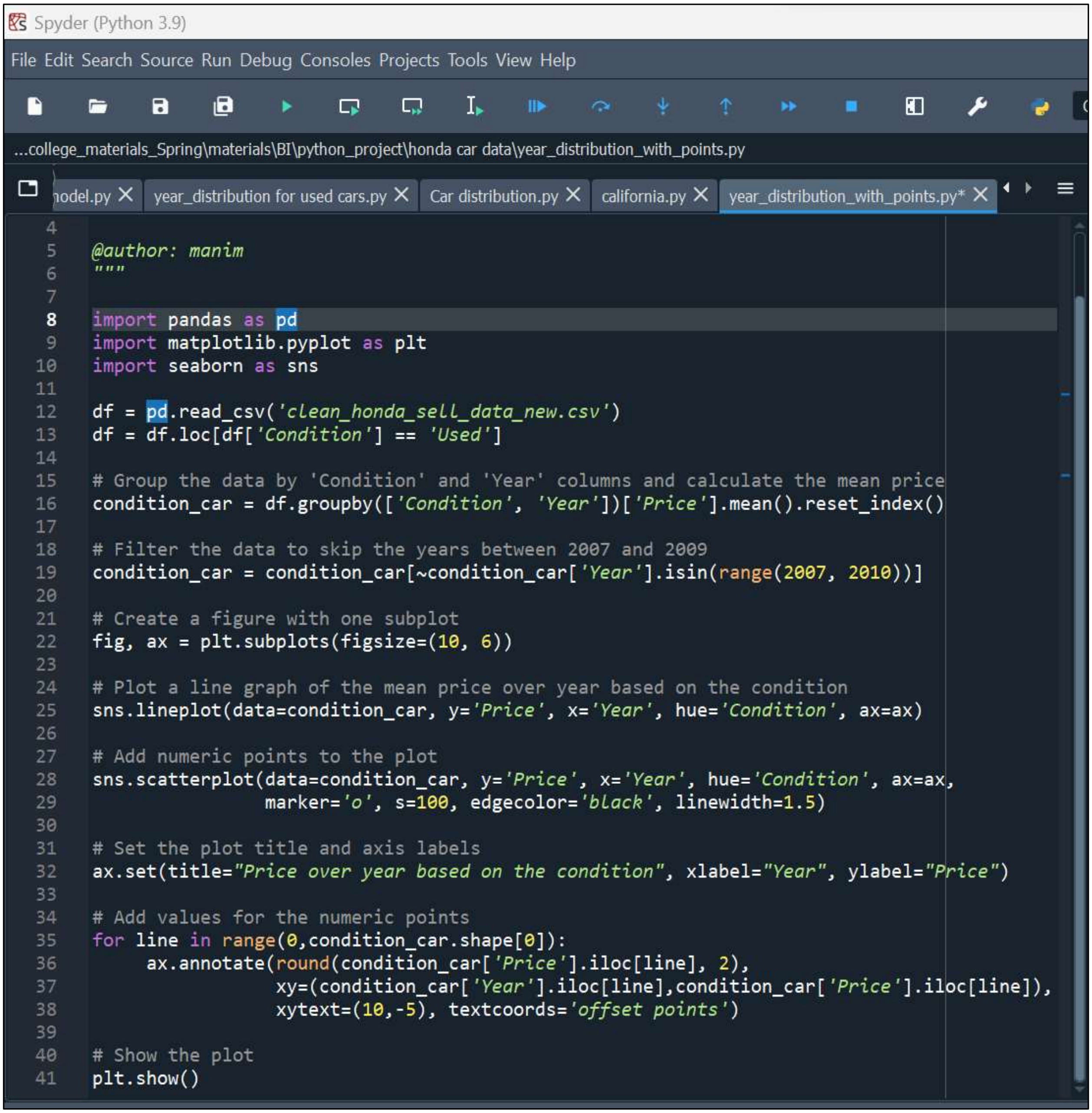
The visualization of the distribution of cars based on condition is done using a pie chart. The color palette used in the pie chart is shades of blue, with the darkest shade being used for the 'Used' condition, followed by 'New' and 'Certified Pre-Owned'.

As part of the analysis, we can conclude that most of the cars in the dataset are in the 'New' condition, accounting for 54.7% of the total cars. This suggests that Honda cars are popular among buyers who are interested in purchasing a new car.

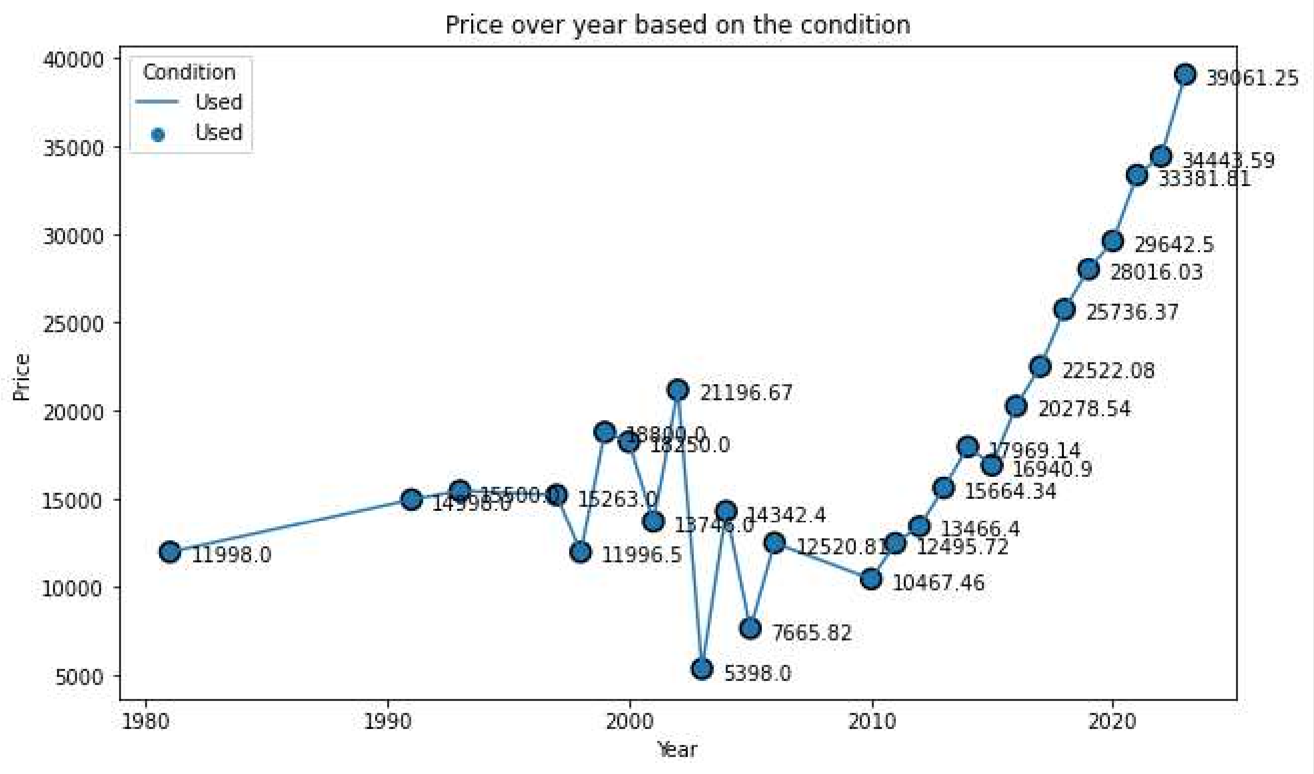
The 'Used' cars account for 39.5%, which shows that there is still a significant demand for used Honda cars. This indicates that Honda cars have a good resale value and are considered reliable vehicles in the market.

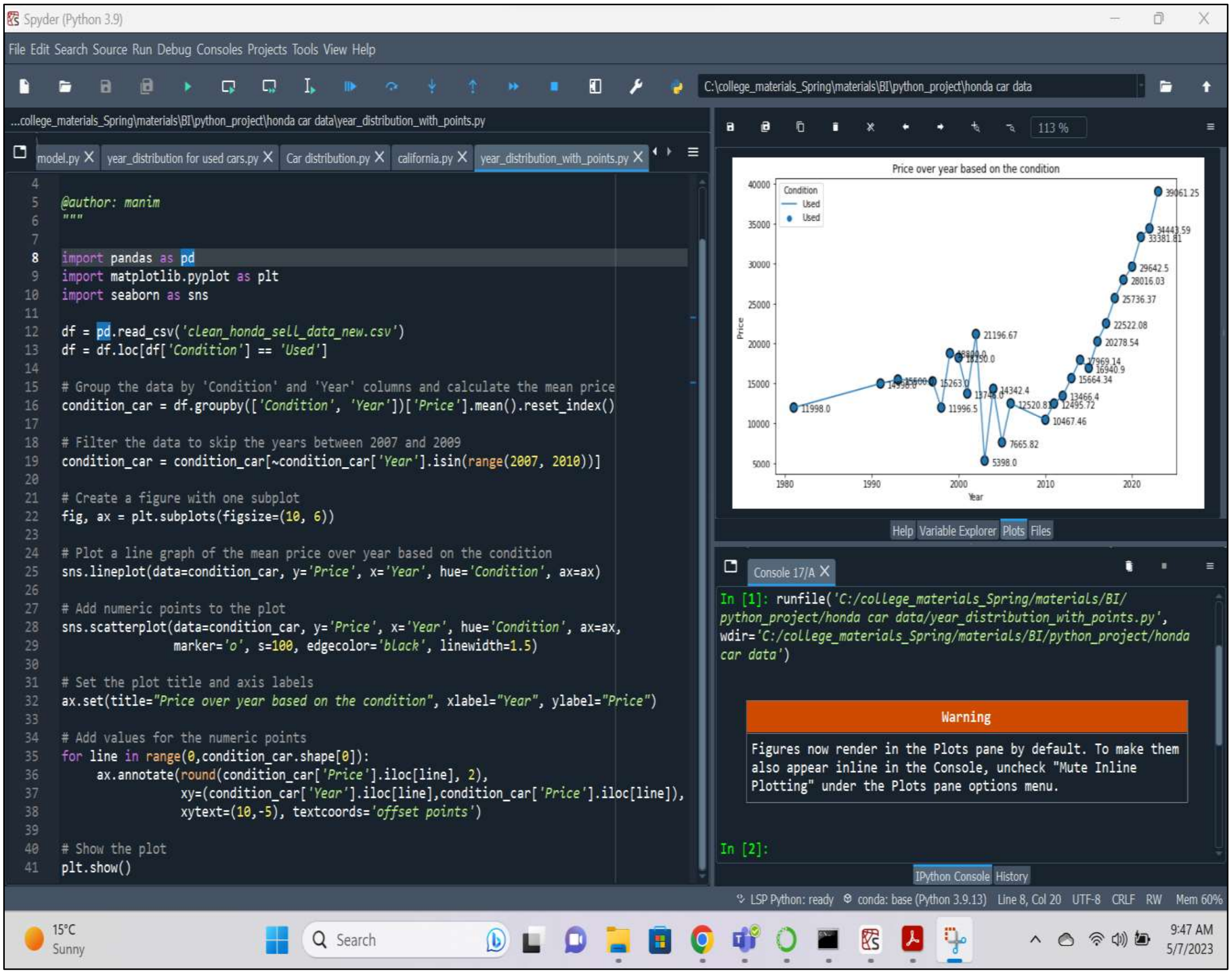
The 'Honda Certified' cars account for only 5.8%, which suggests that buyers may not be as interested in certified pre-owned Honda cars. However, this could also indicate that the certification process may not be well-known or trusted among buyers.

2. How has the price of Honda model car varied by year?


Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
...college_materials_Spring\materials\BI\python_project\honda car data\year_distribution_with_points.py
nodel.py
year_distribution for used cars.py
Car distribution.py
california.py
year_distribution_with_points.py*

```python

@author: manim
"""

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

df = pd.read_csv('clean_honda_sell_data_new.csv')
df = df.loc[df['Condition'] == 'Used']

# Group the data by 'Condition' and 'Year' columns and calculate the mean price
condition_car = df.groupby(['Condition', 'Year'])['Price'].mean().reset_index()

# Filter the data to skip the years between 2007 and 2009
condition_car = condition_car[~condition_car['Year'].isin(range(2007, 2010))]

# Create a figure with one subplot
fig, ax = plt.subplots(figsize=(10, 6))

# Plot a line graph of the mean price over year based on the condition
sns.lineplot(data=condition_car, y='Price', x='Year', hue='Condition', ax=ax)

# Add numeric points to the plot
sns.scatterplot(data=condition_car, y='Price', x='Year', hue='Condition', ax=ax,
                marker='o', s=100, edgecolor='black', linewidth=1.5)

# Set the plot title and axis labels
ax.set(title="Price over year based on the condition", xlabel="Year", ylabel="Price")

# Add values for the numeric points
for line in range(0,condition_car.shape[0]):
     ax.annotate(round(condition_car['Price'].iloc[line], 2),
                 xy=(condition_car['Year'].iloc[line],condition_car['Price'].iloc[line]),
                 xytext=(10,-5), textcoords='offset points')

# Show the plot
plt.show()
```


Price over year based on the condition
Condition
Used
Used
40000
35000
30000
25000
20000
15000
10000
5000
Price
1980
1990
2000
2010
2020
Year
11998.0
14998.0
15500.0
15263.0
11996.5
18800.0
18250.0
13746.0
21196.67
5398.0
14342.4
7665.82
12520.81
10467.46
12495.72
13466.4
15664.34
17969.14
16940.9
20278.54
22522.08
25736.37
28016.03
29642.5
33381.81
34443.59
39061.25



```
@author: manim
"""

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

df = pd.read_csv('clean_honda_sell_data_new.csv')
df = df.loc[df['Condition'] == 'Used']

# Group the data by 'Condition' and 'Year' columns and calculate the mean price
condition_car = df.groupby(['Condition', 'Year'])['Price'].mean().reset_index()

# Filter the data to skip the years between 2007 and 2009
condition_car = condition_car[~condition_car['Year'].isin(range(2007, 2010))]

# Create a figure with one subplot
fig, ax = plt.subplots(figsize=(10, 6))

# Plot a line graph of the mean price over year based on the condition
sns.lineplot(data=condition_car, y='Price', x='Year', hue='Condition', ax=ax)

# Add numeric points to the plot
sns.scatterplot(data=condition_car, y='Price', x='Year', hue='Condition', ax=ax,
                marker='o', s=100, edgecolor='black', linewidth=1.5)

# Set the plot title and axis labels
ax.set(title="Price over year based on the condition", xlabel="Year", ylabel="Price")

# Add values for the numeric points
for line in range(0,condition_car.shape[0]):
     ax.annotate(round(condition_car['Price'].iloc[line], 2),
                 xy=(condition_car['Year'].iloc[line],condition_car['Price'].iloc[line]),
                 xytext=(10,-5), textcoords='offset points')

# Show the plot
plt.show()
```

```
In [1]: runfile('C:/college_materials_Spring/materials/BI/
python_project/honda car data/year_distribution_with_points.py',
wdir='C:/college_materials_Spring/materials/BI/python_project/honda
car data')

Warning

Figures now render in the Plots pane by default. To make them
also appear inline in the Console, uncheck "Mute Inline
Plotting" under the Plots pane options menu.

In [2]:
```

Insights:

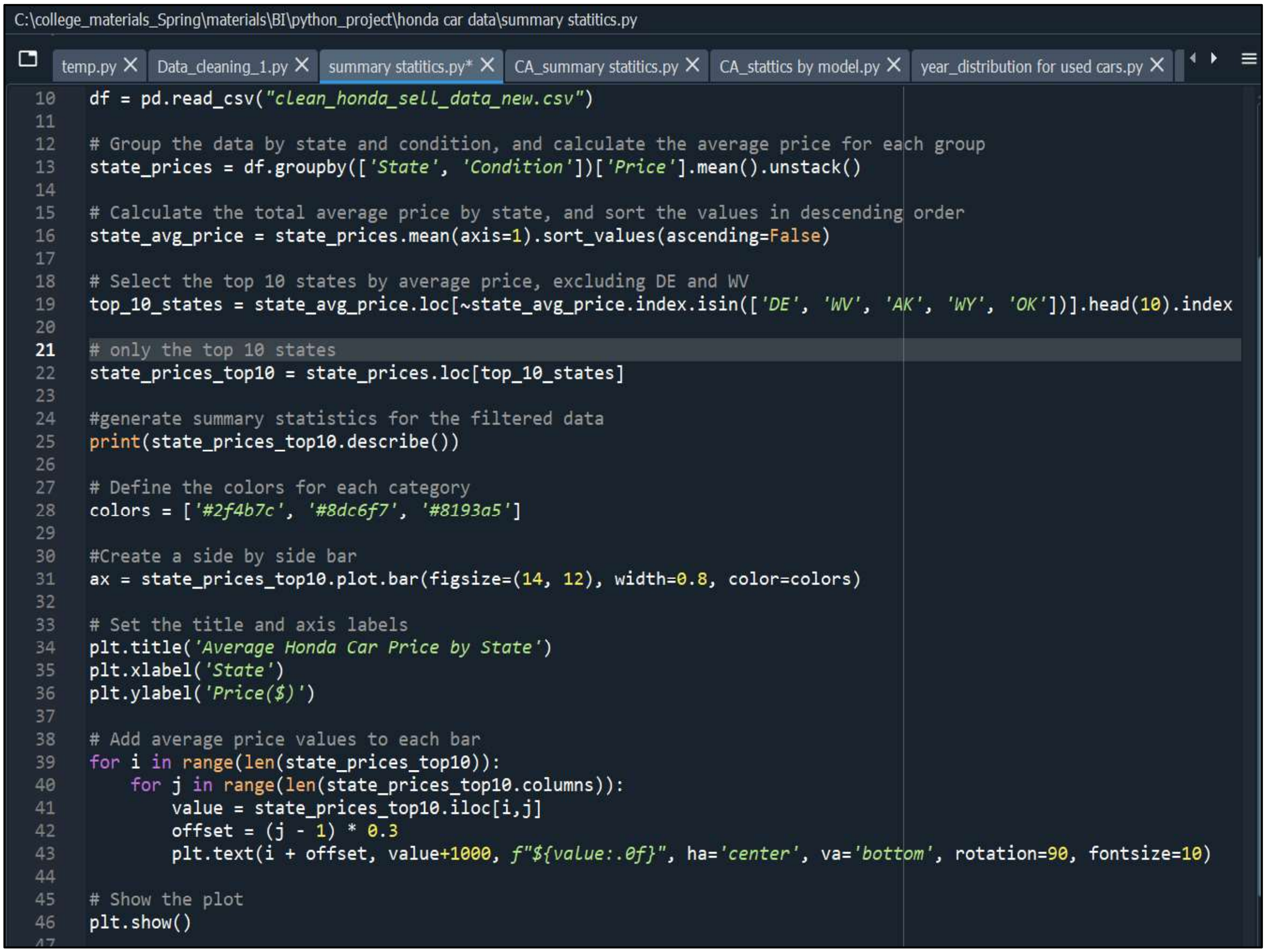
The data is filtered for only used cars as the analysis is focused on the price trends for used cars.

The data is grouped by condition and year, and the mean price of the used cars is calculated. This is done to indicate that the analysis is focused on the price trends of used cars. The resulting line graph shows how the average price of used Honda cars has fluctuated over time.

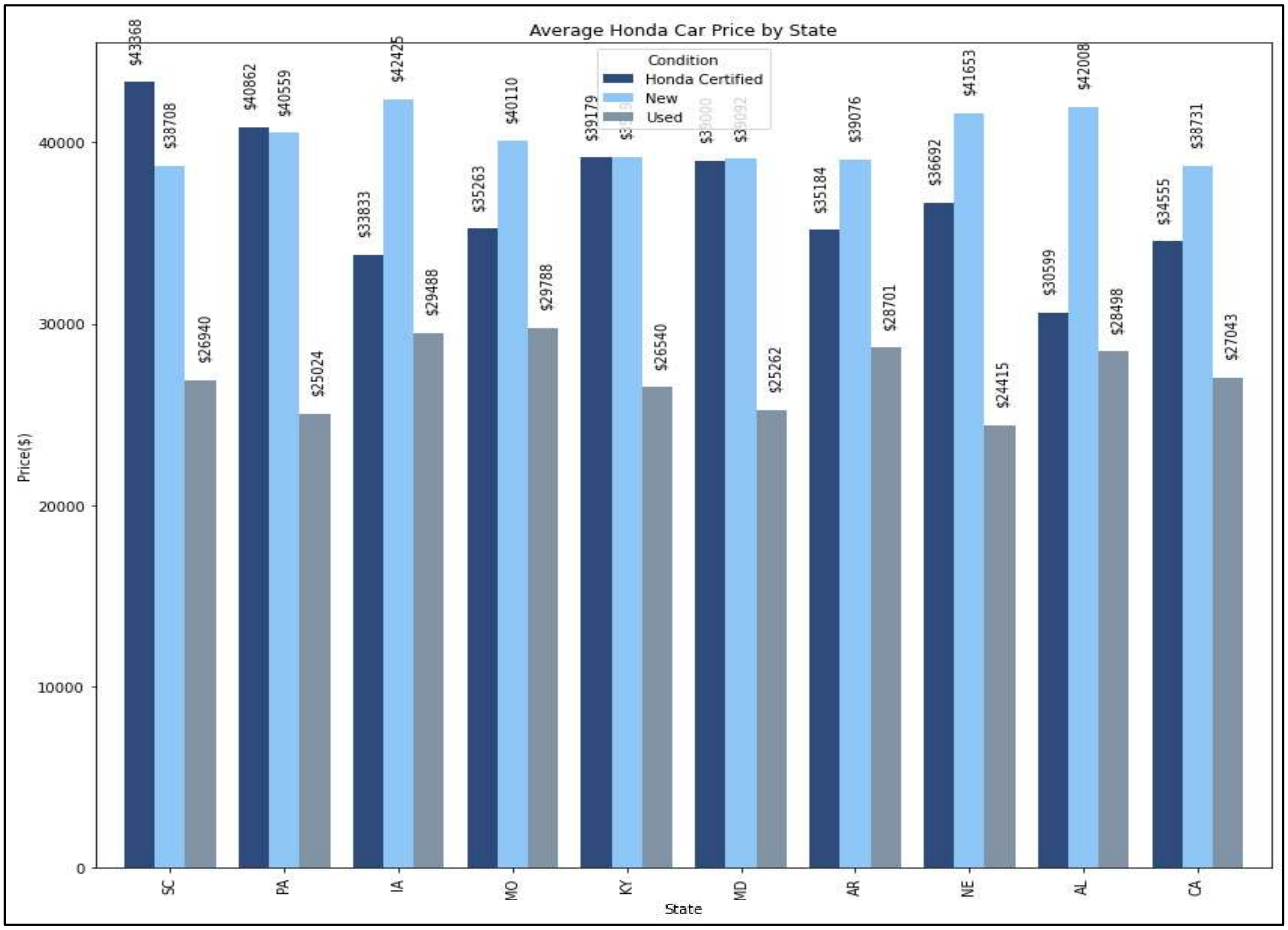
Through the line plot we visualize the mean price of the used cars over the years based on their condition. The x-axis shows the Year, and the Y-axis shows the Average Price. This visualization helps in understanding the trend in price changes for used cars over time.

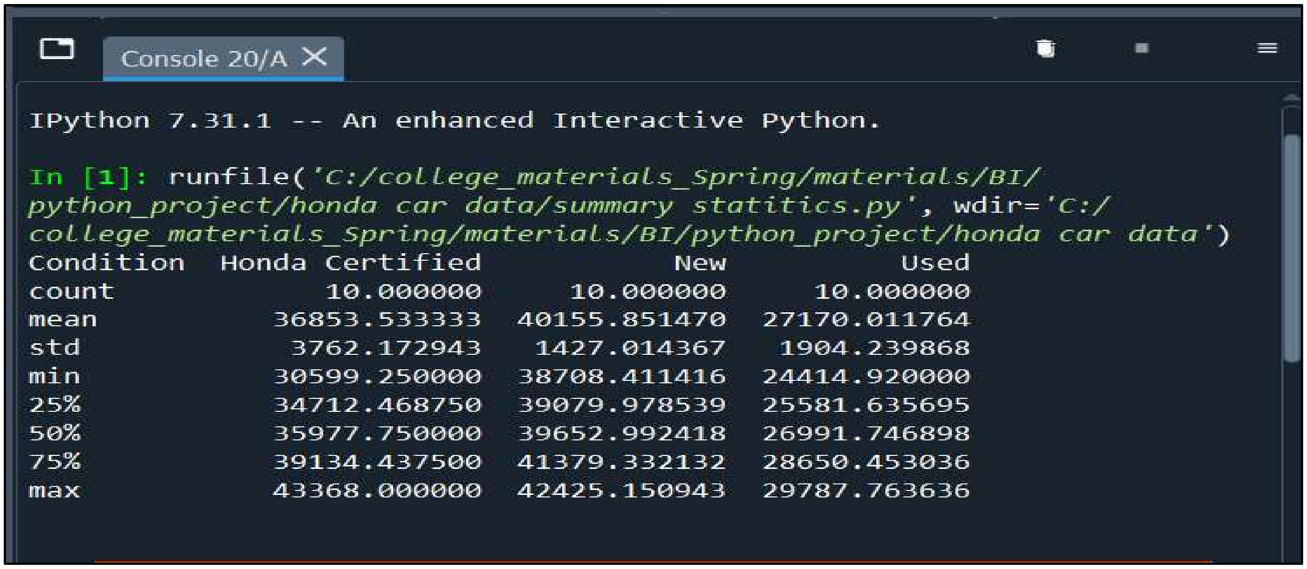
The plot indicates that the price of used cars has increased over the years. We can see that after the year 2000 till 2010 year, the price of used cars dropped drastically and after 2010, the average price of used cars shoots up very high. This indicates that the demand for used cars has increased over time, leading to an increase in their prices.

3. Show the Average Honda cars Price for top 10 states

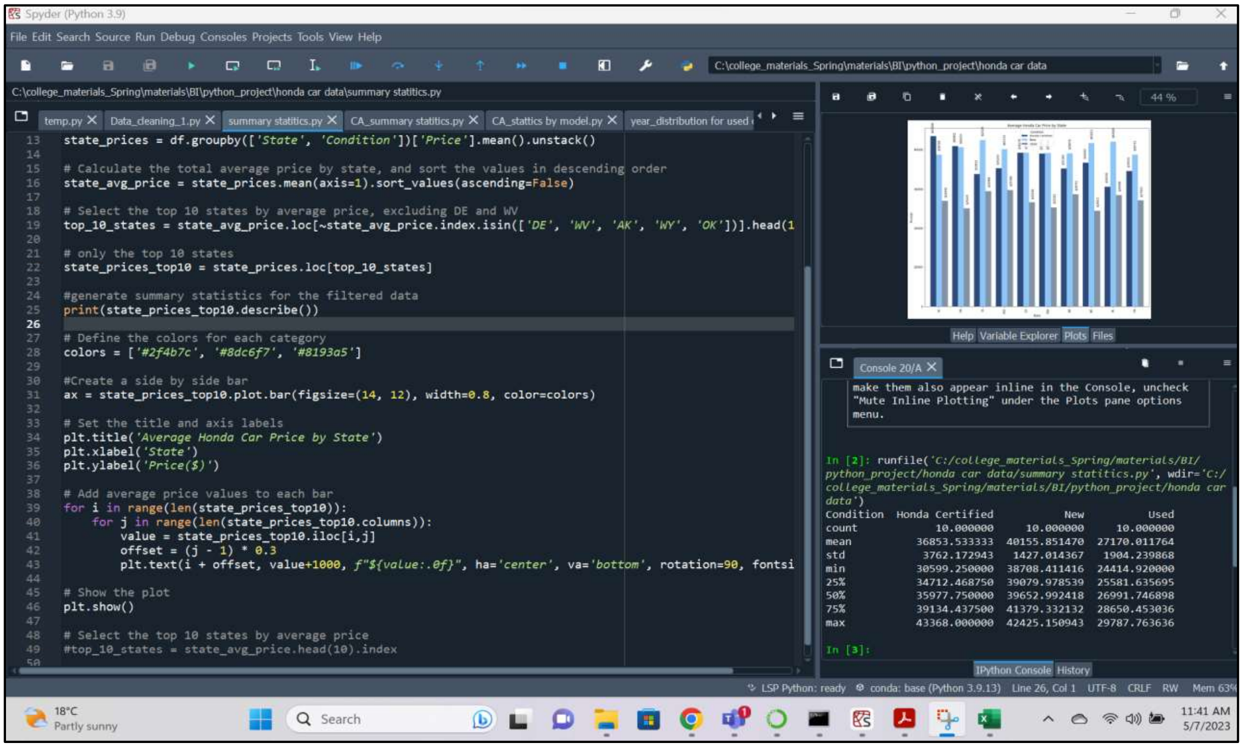

C:\college_materials_Spring\materials\BI\python_project\honda car data\summary statitics.py
temp.py
Data_cleaning_1.py
summary statitics.py*
CA_summary statitics.py
CA_stattics by model.py
year_distribution for used cars.py

```python
df = pd.read_csv("clean_honda_sell_data_new.csv")

# Group the data by state and condition, and calculate the average price for each group
state_prices = df.groupby(['State', 'Condition'])['Price'].mean().unstack()

# Calculate the total average price by state, and sort the values in descending order
state_avg_price = state_prices.mean(axis=1).sort_values(ascending=False)

# Select the top 10 states by average price, excluding DE and WV
top_10_states = state_avg_price.loc[~state_avg_price.index.isin(['DE', 'WV', 'AK', 'WY', 'OK'])].head(10).index

# only the top 10 states
state_prices_top10 = state_prices.loc[top_10_states]

#generate summary statistics for the filtered data
print(state_prices_top10.describe())

# Define the colors for each category
colors = ['#2f4b7c', '#8dc6f7', '#8193a5']

#Create a side by side bar
ax = state_prices_top10.plot.bar(figsize=(14, 12), width=0.8, color=colors)

# Set the title and axis labels
plt.title('Average Honda Car Price by State')
plt.xlabel('State')
plt.ylabel('Price($)')

# Add average price values to each bar
for i in range(len(state_prices_top10)):
    for j in range(len(state_prices_top10.columns)):
        value = state_prices_top10.iloc[i,j]
        offset = (j - 1) * 0.3
        plt.text(i + offset, value+1000, f"${value:.0f}", ha='center', va='bottom', rotation=90, fontsize=10)

# Show the plot
plt.show()
```





```
IPython 7.31.1 -- An enhanced Interactive Python.

In [1]: runfile('C:/college_materials_Spring/materials/BI/
python_project/honda car data/summary statitics.py', wdir='C:/
college_materials_Spring/materials/BI/python_project/honda car data')
Condition   Honda Certified           New          Used
count             10.000000     10.000000     10.000000
mean           36853.533333  40155.851470  27170.011764
std             3762.172943   1427.014367   1904.239868
min            30599.250000  38708.411416  24414.920000
25%            34712.468750  39079.978539  25581.635695
50%            35977.750000  39652.992418  26991.746898
75%            39134.437500  41379.332132  28650.453036
max            43368.000000  42425.150943  29787.763636
```



Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\college_materials_Spring\materials\BI\python_project\honda car data
C:\college_materials_Spring\materials\BI\python_project\honda car data\summary statitics.py
temp.py
Data_cleaning_1.py
summary statitics.py
CA_summary statitics.py
CA_stattics by model.py
year_distribution for used
44 %
Help
Variable Explorer
Plots
Files
Console 20/A
IPython Console
History
LSP Python: ready
conda: base (Python 3.9.13)
Line 26, Col 1
UTF-8
CRLF
RW
Mem 63%
18°C
Partly sunny
Search
11:41 AM
5/7/2023

```python
state_prices = df.groupby(['State', 'Condition'])['Price'].mean().unstack()

# Calculate the total average price by state, and sort the values in descending order
state_avg_price = state_prices.mean(axis=1).sort_values(ascending=False)

# Select the top 10 states by average price, excluding DE and WV
top_10_states = state_avg_price.loc[~state_avg_price.index.isin(['DE', 'WV', 'AK', 'WY', 'OK'])].head(1

# only the top 10 states
state_prices_top10 = state_prices.loc[top_10_states]

#generate summary statistics for the filtered data
print(state_prices_top10.describe())

# Define the colors for each category
colors = ['#2f4b7c', '#8dc6f7', '#8193a5']

#Create a side by side bar
ax = state_prices_top10.plot.bar(figsize=(14, 12), width=0.8, color=colors)

# Set the title and axis labels
plt.title('Average Honda Car Price by State')
plt.xlabel('State')
plt.ylabel('Price($)')

# Add average price values to each bar
for i in range(len(state_prices_top10)):
    for j in range(len(state_prices_top10.columns)):
        value = state_prices_top10.iloc[i,j]
        offset = (j - 1) * 0.3
        plt.text(i + offset, value+1000, f"${value:.0f}", ha='center', va='bottom', rotation=90, fontsi

# Show the plot
plt.show()

# Select the top 10 states by average price
#top_10_states = state_avg_price.head(10).index
```

```
make them also appear inline in the Console, uncheck
"Mute Inline Plotting" under the Plots pane options
menu.


In [2]: runfile('C:/college_materials_Spring/materials/BI/
python_project/honda car data/summary statitics.py', wdir='C:/
college_materials_Spring/materials/BI/python_project/honda car
data')
Condition  Honda Certified           New          Used
count            10.000000     10.000000     10.000000
mean          36853.533333  40155.851470  27170.011764
std            3762.172943   1427.014367   1904.239868
min           30599.250000  38708.411416  24414.920000
25%           34712.468750  39079.978539  25581.635695
50%           35977.750000  39652.992418  26991.746898
75%           39134.437500  41379.332132  28650.453036
max           43368.000000  42425.150943  29787.763636

In [3]:
```

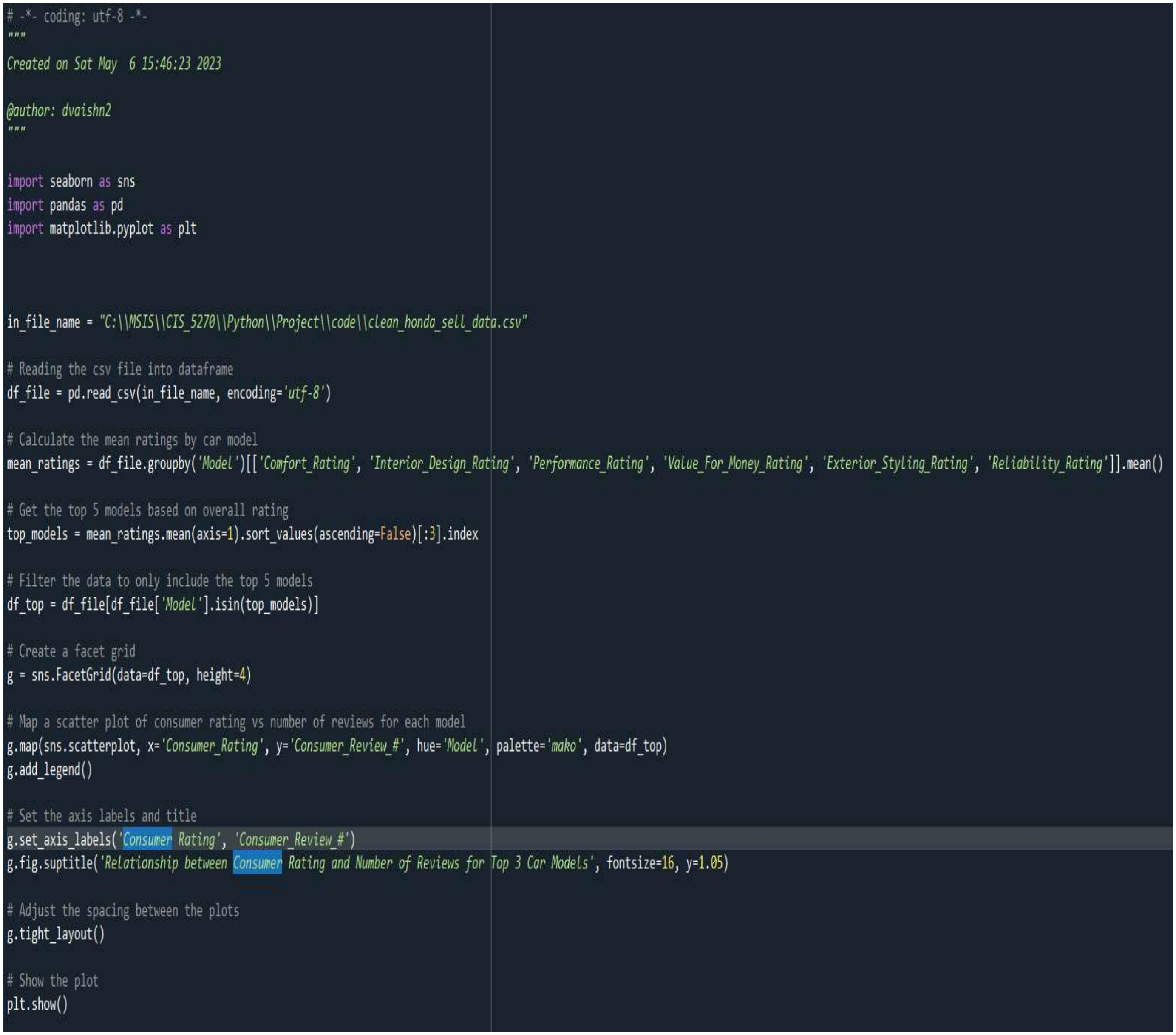
Insights:

The code groups the data by state and condition, calculates the average price for each group, and creates a side-by-side bar chart showing the average Honda car price by state. Only the data for these top 10 states is used to generate the bar chart. The bar chart showed us that the average price of Honda cars varied widely by state and condition, with some states having much higher prices than others.

Through this analysis we can understand that the state SC (South Carolina) has highest price for Honda certified cars and in the state IA (IOWA) Average New Honda car price is $42,425. The Used Honda car has the highest price in the state of MO (Missouri). In the State PY(Pennsylvania), MD(Maryland), KY(Kentucky) has not much difference between the price of Honda Certified cars and new cars. Overall, there is not much significant difference between the Average price of NEW cars across the states.

Overall, this analysis provides useful insights for anyone interested in buying or selling Honda cars in the United States, as it highlights the importance of considering regional variations in price.

4. Display the correlation between the number of reviews and consumer ratings for the top 3 car models, which are determined based on their consumer ratings

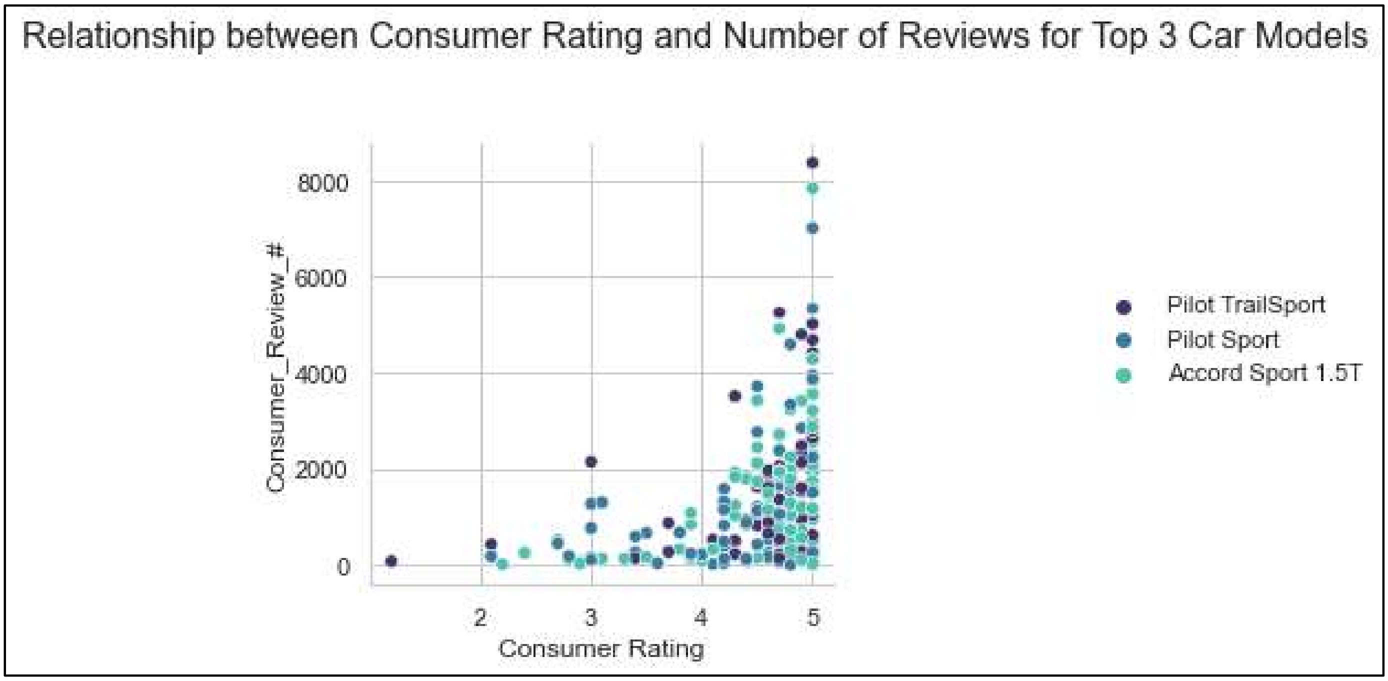


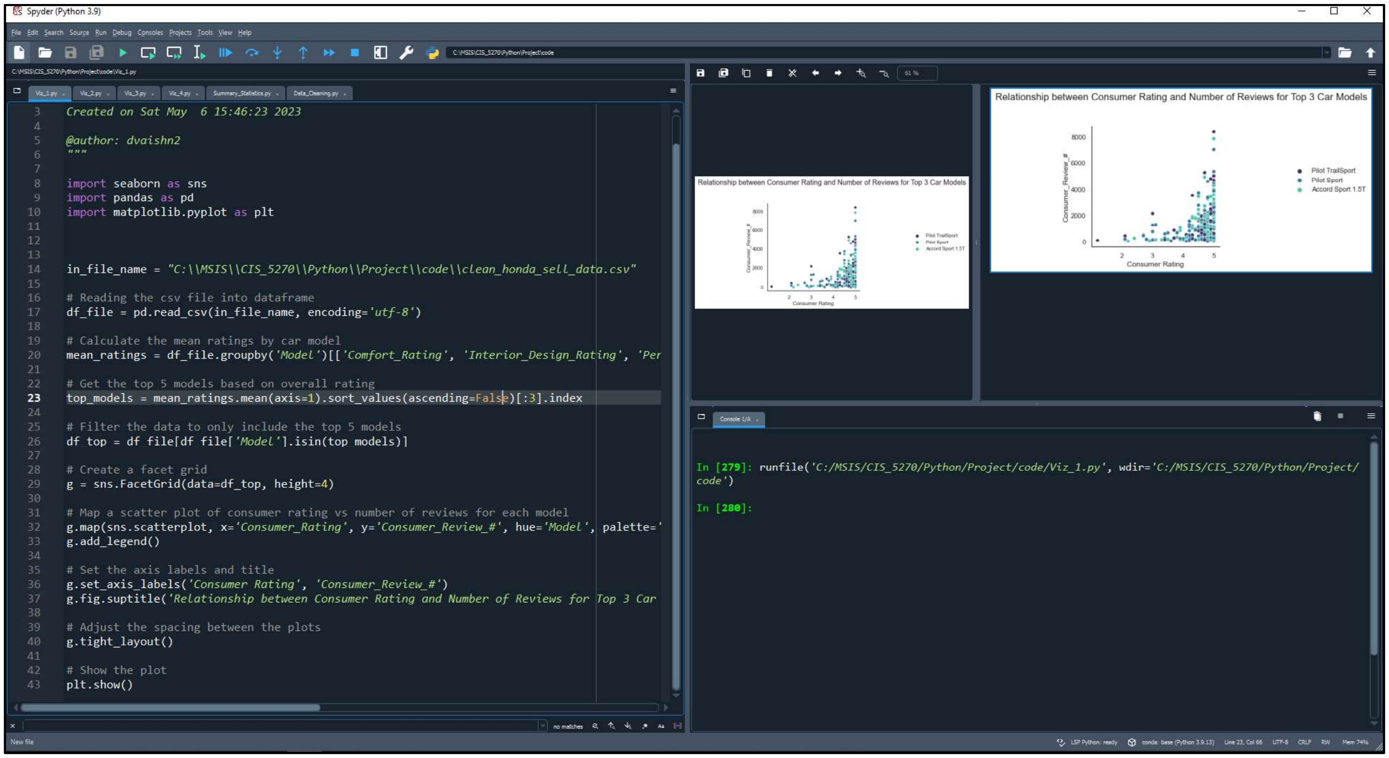
```
# -*- coding: utf-8 -*-
"""
Created on Sat May  6 15:46:23 2023

@author: dvaishn2
"""

import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt



in_file_name = "C:\\MSIS\\CIS_5270\\Python\\Project\\code\\clean_honda_sell_data.csv"

# Reading the csv file into dataframe
df_file = pd.read_csv(in_file_name, encoding='utf-8')

# Calculate the mean ratings by car model
mean_ratings = df_file.groupby('Model')[['Comfort_Rating', 'Interior_Design_Rating', 'Performance_Rating', 'Value_For_Money_Rating', 'Exterior_Styling_Rating', 'Reliability_Rating']].mean()

# Get the top 5 models based on overall rating
top_models = mean_ratings.mean(axis=1).sort_values(ascending=False)[:3].index

# Filter the data to only include the top 5 models
df_top = df_file[df_file['Model'].isin(top_models)]

# Create a facet grid
g = sns.FacetGrid(data=df_top, height=4)

# Map a scatter plot of consumer rating vs number of reviews for each model
g.map(sns.scatterplot, x='Consumer_Rating', y='Consumer_Review_#', hue='Model', palette='mako', data=df_top)
g.add_legend()

# Set the axis labels and title
g.set_axis_labels('Consumer Rating', 'Consumer_Review_#')
g.fig.suptitle('Relationship between Consumer Rating and Number of Reviews for Top 3 Car Models', fontsize=16, y=1.05)

# Adjust the spacing between the plots
g.tight_layout()

# Show the plot
plt.show()
```

```
# -*- coding: utf-8 -*-
"""
Created on Sat May  6 15:46:23 2023
@author: dvaishn2
"""
```

```
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt

in_file_name = "C:\\MSIS\\CIS_5270\\Python\\Project\\code\\clean_honda_sell_data.csv"

# Reading the csv file into dataframe
df_file = pd.read_csv(in_file_name, encoding='utf-8')

# Calculate the mean ratings by car model
mean_ratings = df_file.groupby('Model')[['Comfort_Rating', 'Interior_Design_Rating',
'Performance_Rating', 'Value_For_Money_Rating', 'Exterior_Styling_Rating',
'Reliability_Rating']].mean()

# Get the top 5 models based on overall rating
top_models = mean_ratings.mean(axis=1).sort_values(ascending=False)[:3].index

# Filter the data to only include the top 5 models
df_top = df_file[df_file['Model'].isin(top_models)]

# Create a facet grid
g = sns.FacetGrid(data=df_top, height=4)

# Map a scatter plot of consumer rating vs number of reviews for each model
g.map(sns.scatterplot, x='Consumer_Rating', y='Consumer_Review_#', hue='Model',
palette='mako', data=df_top)
g.add_legend()

# Set the axis labels and title
```

```
g.set_axis_labels('Consumer Rating', 'Consumer_Review_#')

g.fig.suptitle('Relationship between Consumer Rating and Number of Reviews for Top 3 Car Models', fontsize=16, y=1.05)


# Adjust the spacing between the plots

g.tight_layout()


# Show the plot

plt.show()
```




Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\MSIS\CIS_5270\Python\Project\code
C:\MSIS\CIS_5270\Python\Project\code\Viz_1.py
Viz_1.py
Viz_2.py
Viz_3.py
Viz_4.py
Summary_Statistics.py
Data_Cleaning.py
Relationship between Consumer Rating and Number of Reviews for Top 3 Car Models
Consumer_Review_#
8000
6000
4000
2000
0
2
3
4
5
Consumer Rating
Pilot TrailSport
Pilot Sport
Accord Sport 1.5T
Console 1/A
no matches
New file

```python
Created on Sat May  6 15:46:23 2023

@author: dvaishn2
"""


import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt



in_file_name = "C:\\MSIS\\CIS_5270\\Python\\Project\\code\\clean_honda_sell_data.csv"

# Reading the csv file into dataframe
df_file = pd.read_csv(in_file_name, encoding='utf-8')

# Calculate the mean ratings by car model
mean_ratings = df_file.groupby('Model')[['Comfort_Rating', 'Interior_Design_Rating', 'Per

# Get the top 5 models based on overall rating
top_models = mean_ratings.mean(axis=1).sort_values(ascending=False)[:3].index

# Filter the data to only include the top 5 models
df_top = df_file[df_file['Model'].isin(top_models)]

# Create a facet grid
g = sns.FacetGrid(data=df_top, height=4)

# Map a scatter plot of consumer rating vs number of reviews for each model
g.map(sns.scatterplot, x='Consumer_Rating', y='Consumer_Review_#', hue='Model', palette='
g.add_legend()

# Set the axis labels and title
g.set_axis_labels('Consumer Rating', 'Consumer_Review_#')
g.fig.suptitle('Relationship between Consumer Rating and Number of Reviews for Top 3 Car

# Adjust the spacing between the plots
g.tight_layout()

# Show the plot
plt.show()
```

```
In [279]: runfile('C:/MSIS/CIS_5270/Python/Project/code/Viz_1.py', wdir='C:/MSIS/CIS_5270/Python/Project/
code')

In [280]:
```

Insights:

This visualization shows the relationship between consumer ratings and the number of reviews for the top three car models based on their overall rating.

A Facet Grid is created, which is a grid of plots showing the same relationship conditioned on different levels of a variable. In this case, Facet Grid contains scatterplots of consumer ratings vs. the number of reviews for each of the top 3 car models.

The scatterplots show the relationship between consumer rating and the number of reviews, with each dot representing a different review.

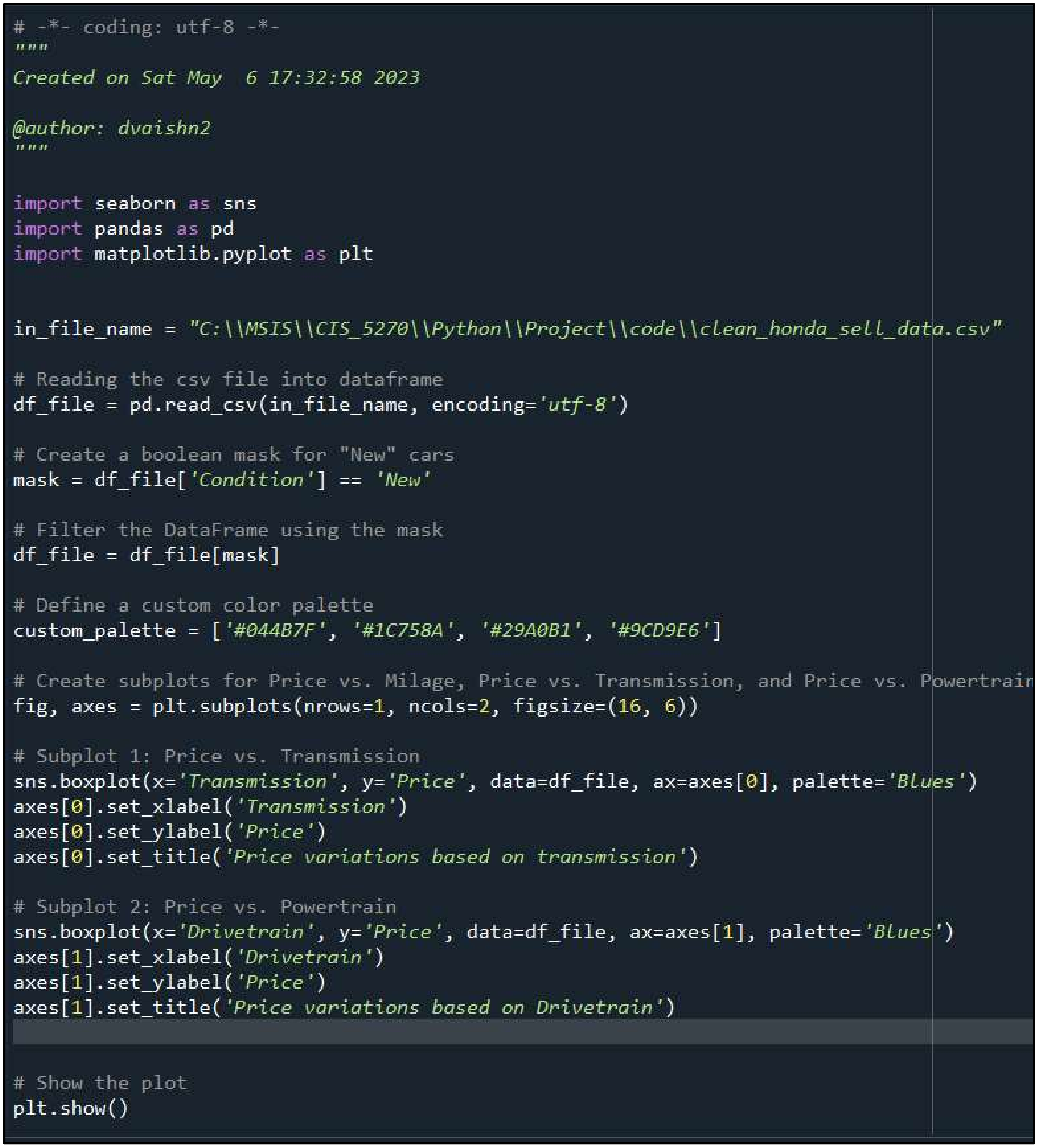
The x-axis of the scatter plot represents the consumer rating while the y-axis represents the number of reviews. The color of each data point represents the car model, and a legend is added to the plot for clarity.

The plot shows that there is a positive relationship between consumer ratings and the number of reviews for each of the top three car models. As the number of reviews increases, so does the consumer rating. However, the magnitude of the relationship differs across the three car models.

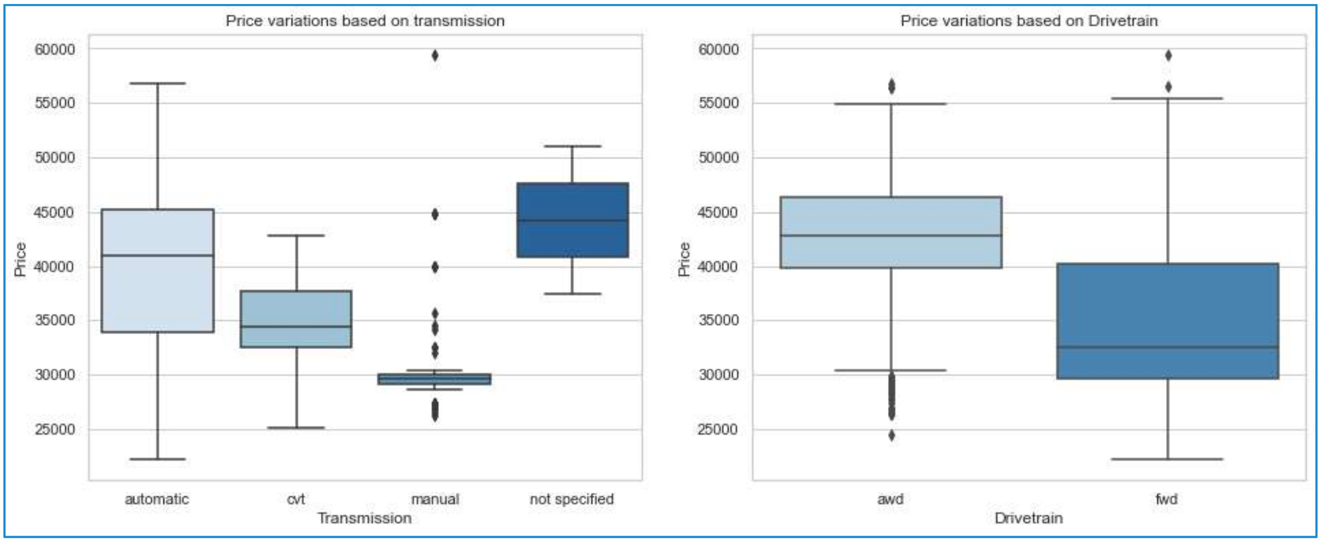
The graph helps in understanding the popularity of the top three car models and how consumer ratings and reviews affect their overall rating.

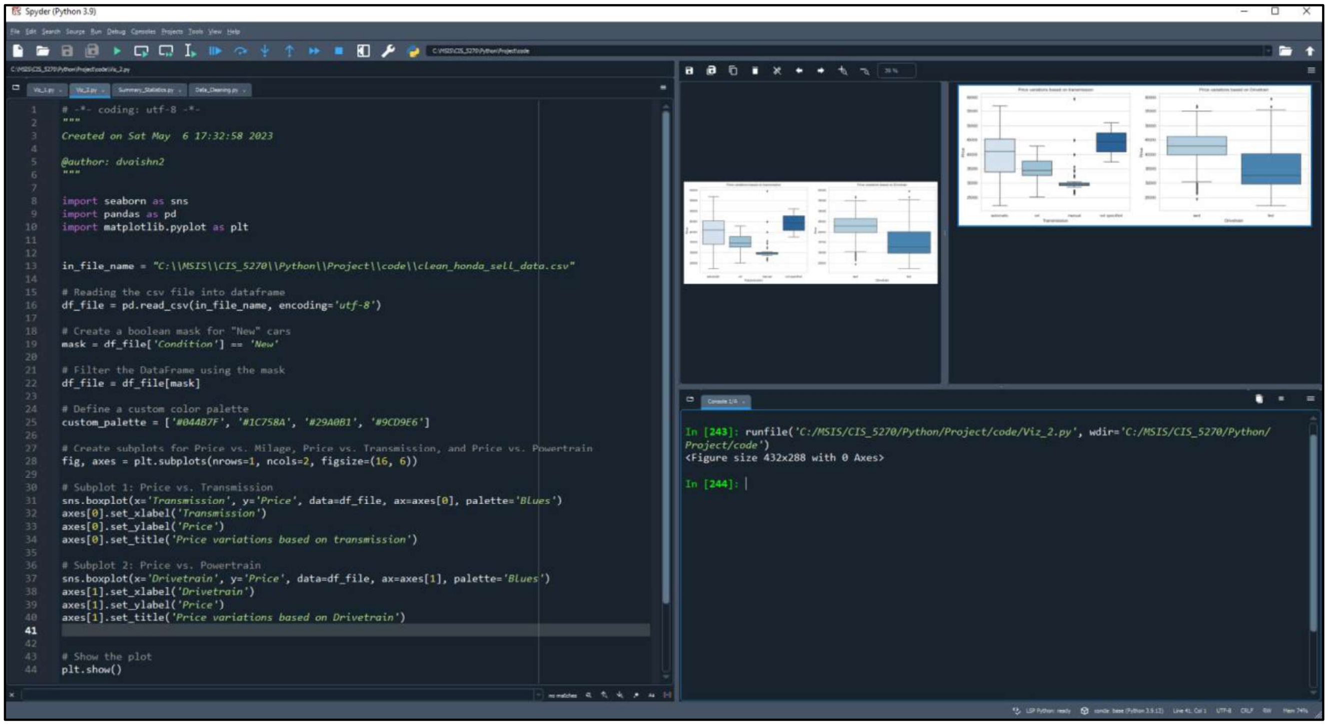
It also provides insights into which car models have a stronger relationship between consumer ratings and the number of reviews.

5. How do the price variations based on transmission compare to the price variations based on powertrain?



```
# -*- coding: utf-8 -*-
"""
Created on Sat May  6 17:32:58 2023

@author: dvaishn2
"""

import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt


in_file_name = "C:\\MSIS\\CIS_5270\\Python\\Project\\code\\clean_honda_sell_data.csv"

# Reading the csv file into dataframe
df_file = pd.read_csv(in_file_name, encoding='utf-8')

# Create a boolean mask for "New" cars
mask = df_file['Condition'] == 'New'

# Filter the DataFrame using the mask
df_file = df_file[mask]

# Define a custom color palette
custom_palette = ['#044B7F', '#1C758A', '#29A0B1', '#9CD9E6']

# Create subplots for Price vs. Milage, Price vs. Transmission, and Price vs. Powertrain
fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(16, 6))

# Subplot 1: Price vs. Transmission
sns.boxplot(x='Transmission', y='Price', data=df_file, ax=axes[0], palette='Blues')
axes[0].set_xlabel('Transmission')
axes[0].set_ylabel('Price')
axes[0].set_title('Price variations based on transmission')

# Subplot 2: Price vs. Powertrain
sns.boxplot(x='Drivetrain', y='Price', data=df_file, ax=axes[1], palette='Blues')
axes[1].set_xlabel('Drivetrain')
axes[1].set_ylabel('Price')
axes[1].set_title('Price variations based on Drivetrain')


# Show the plot
plt.show()
```





```python
# -*- coding: utf-8 -*-
"""
Created on Sat May  6 17:32:58 2023

@author: dvaishn2
"""

import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt


in_file_name = "C:\\MSIS\\CIS_5270\\Python\\Project\\code\\clean_honda_sell_data.csv"

# Reading the csv file into dataframe
df_file = pd.read_csv(in_file_name, encoding='utf-8')

# Create a boolean mask for "New" cars
mask = df_file['Condition'] == 'New'

# Filter the DataFrame using the mask
df_file = df_file[mask]

# Define a custom color palette
custom_palette = ['#044B7F', '#1C758A', '#29A0B1', '#9CD9E6']

# Create subplots for Price vs. Milage, Price vs. Transmission, and Price vs. Powertrain
fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(16, 6))

# Subplot 1: Price vs. Transmission
sns.boxplot(x='Transmission', y='Price', data=df_file, ax=axes[0], palette='Blues')
axes[0].set_xlabel('Transmission')
axes[0].set_ylabel('Price')
axes[0].set_title('Price variations based on transmission')

# Subplot 2: Price vs. Powertrain
sns.boxplot(x='Drivetrain', y='Price', data=df_file, ax=axes[1], palette='Blues')
axes[1].set_xlabel('Drivetrain')
axes[1].set_ylabel('Price')
axes[1].set_title('Price variations based on Drivetrain')


# Show the plot
plt.show()
```

```
In [243]: runfile('C:/MSIS/CIS_5270/Python/Project/code/Viz_2.py', wdir='C:/MSIS/CIS_5270/Python/
Project/code')
<Figure size 432x288 with 0 Axes>

In [244]: 
```

Insights:

This visualization has two boxplots to show the variations in price of new cars based on their transmission type and drivetrain.

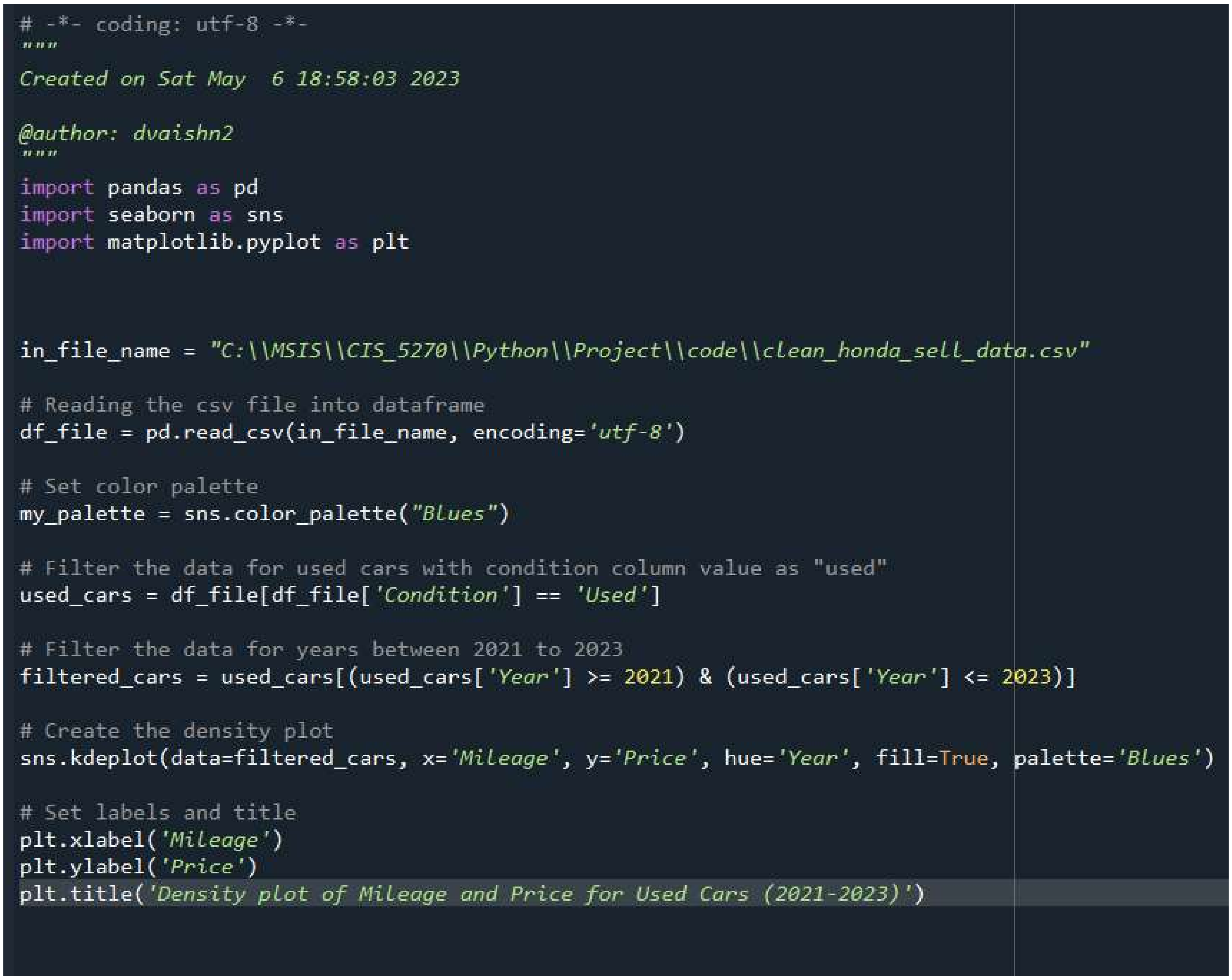
The left plot shows the relationship between price and transmission, while the right plot shows the relationship between price and drivetrain.

The boxplots display the distribution of prices for each category and highlight the median (the line in the box), the interquartile range (the box itself), and the range of values (the whiskers). The plot helps to identify whether there are significant differences in prices between different categories of transmission and drivetrain.

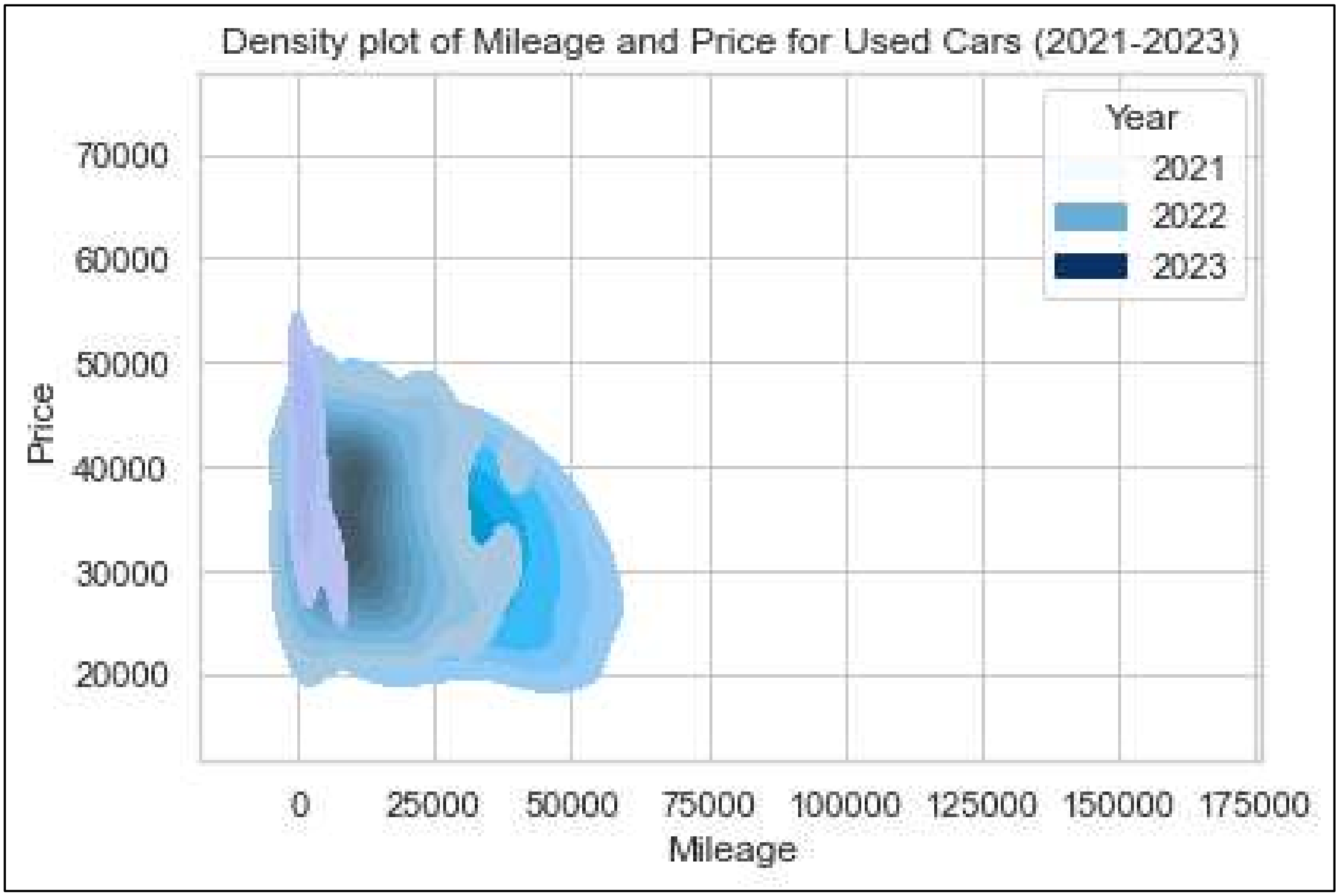
From the visualization, we can see that automatic transmission and all-wheel drivetrain cars tend to have higher prices compared to other categories (except for the not specified ones).

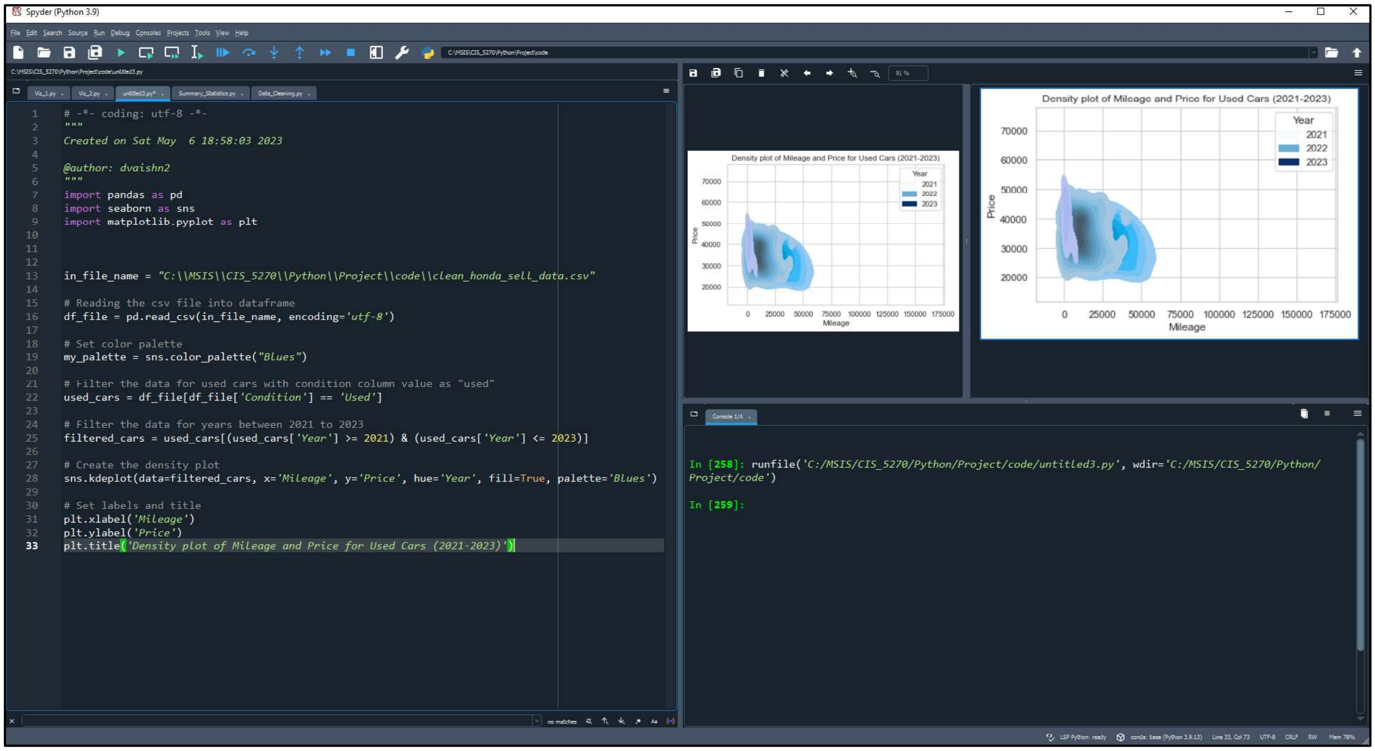
Additionally, the boxplots also show the presence of outliers, which are the individual data points outside the whiskers of the boxplots. These outliers are cars with prices that are significantly higher or lower than the typical prices for a given category.

6. Show the relationship between the MILEAGE values and PRICES for USED cars over a span of three years.



```
# -*- coding: utf-8 -*-
"""
Created on Sat May  6 18:58:03 2023

@author: dvaishn2
"""
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt



in_file_name = "C:\\MSIS\\CIS_5270\\Python\\Project\\code\\clean_honda_sell_data.csv"

# Reading the csv file into dataframe
df_file = pd.read_csv(in_file_name, encoding='utf-8')

# Set color palette
my_palette = sns.color_palette("Blues")

# Filter the data for used cars with condition column value as "used"
used_cars = df_file[df_file['Condition'] == 'Used']

# Filter the data for years between 2021 to 2023
filtered_cars = used_cars[(used_cars['Year'] >= 2021) & (used_cars['Year'] <= 2023)]

# Create the density plot
sns.kdeplot(data=filtered_cars, x='Mileage', y='Price', hue='Year', fill=True, palette='Blues')

# Set labels and title
plt.xlabel('Mileage')
plt.ylabel('Price')
plt.title('Density plot of Mileage and Price for Used Cars (2021-2023)')
```




Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\MSIS\CIS_5270\Python\Project\code
C:\MSIS\CIS_5270\Python\Project\code\untitled3.py
Viz_1.py
Viz_2.py
untitled3.py*
Summary_Statistics.py
Data_Cleaning.py
Density plot of Mileage and Price for Used Cars (2021-2023)
Year
2021
2022
2023
Price
Mileage
Console 1/A

```python
# -*- coding: utf-8 -*-
"""
Created on Sat May  6 18:58:03 2023

@author: dvaishn2
"""
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt



in_file_name = "C:\\MSIS\\CIS_5270\\Python\\Project\\code\\clean_honda_sell_data.csv"

# Reading the csv file into dataframe
df_file = pd.read_csv(in_file_name, encoding='utf-8')

# Set color palette
my_palette = sns.color_palette("Blues")

# Filter the data for used cars with condition column value as "used"
used_cars = df_file[df_file['Condition'] == 'Used']

# Filter the data for years between 2021 to 2023
filtered_cars = used_cars[(used_cars['Year'] >= 2021) & (used_cars['Year'] <= 2023)]

# Create the density plot
sns.kdeplot(data=filtered_cars, x='Mileage', y='Price', hue='Year', fill=True, palette='Blues')

# Set labels and title
plt.xlabel('Mileage')
plt.ylabel('Price')
plt.title('Density plot of Mileage and Price for Used Cars (2021-2023)')
```

```
In [258]: runfile('C:/MSIS/CIS_5270/Python/Project/code/untitled3.py', wdir='C:/MSIS/CIS_5270/Python/
Project/code')

In [259]:
```

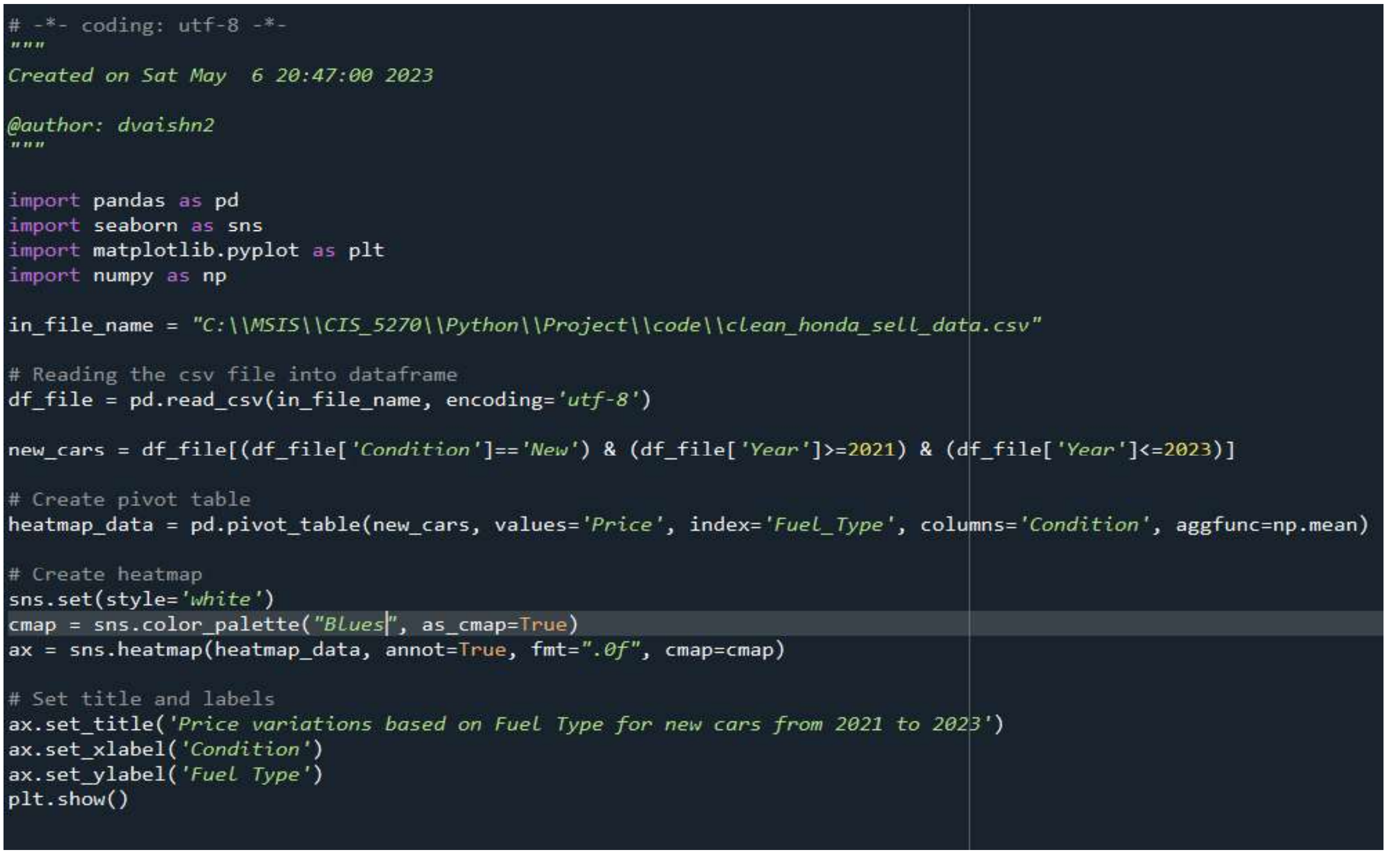
Insights:

The density plot visualizes the spread of used car prices based on their mileage and year of manufacture. The x-axis displays the mileage, the y-axis represents the year, and the color of the plot denotes the density of the price distribution.

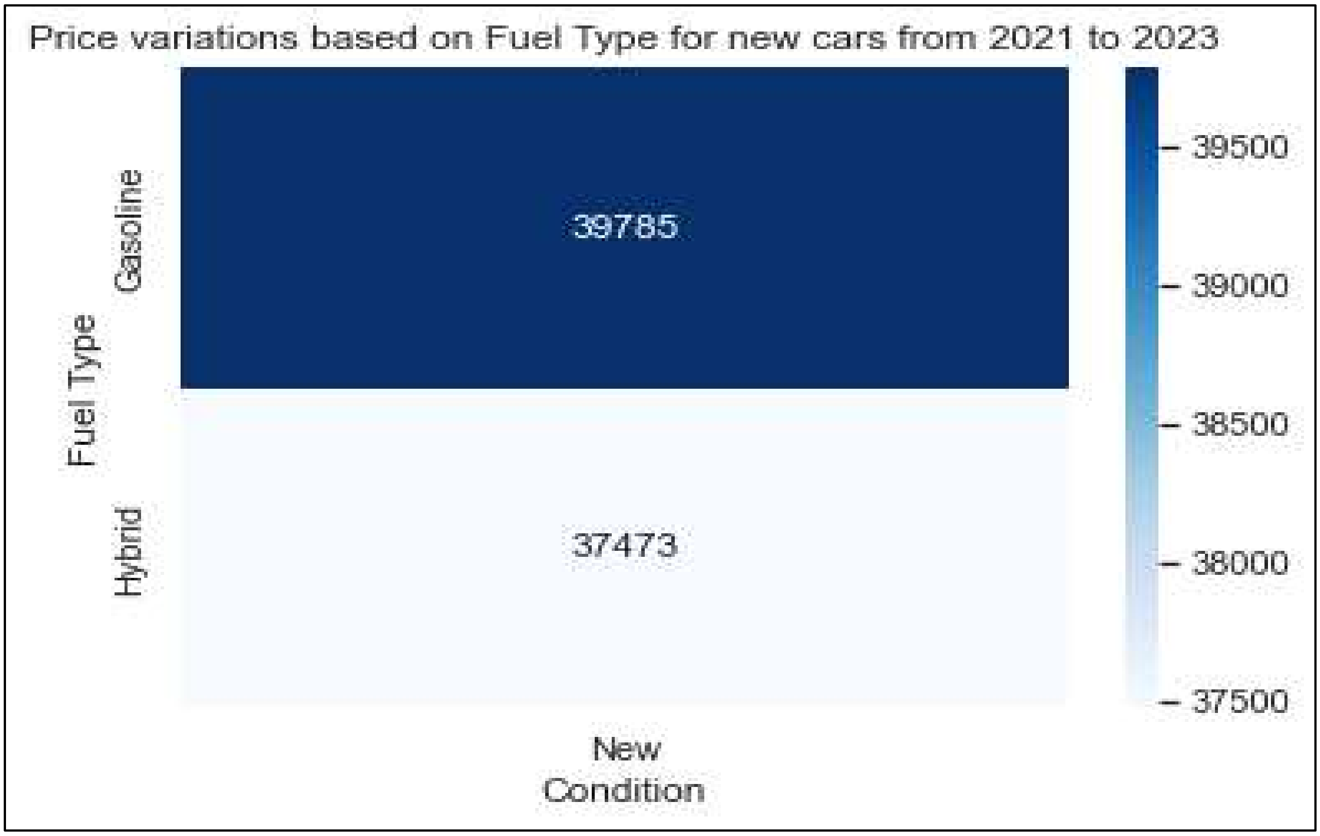
Darker regions on the plot indicate that there are more used cars with similar prices, while lighter regions show a lower density of prices. We can observe that most used car prices are clustered in the range of 0 to 60,000 miles (about 96560.64 km) and for cars manufactured in 2021. As the mileage and year of the car increase, the density of prices decreases.

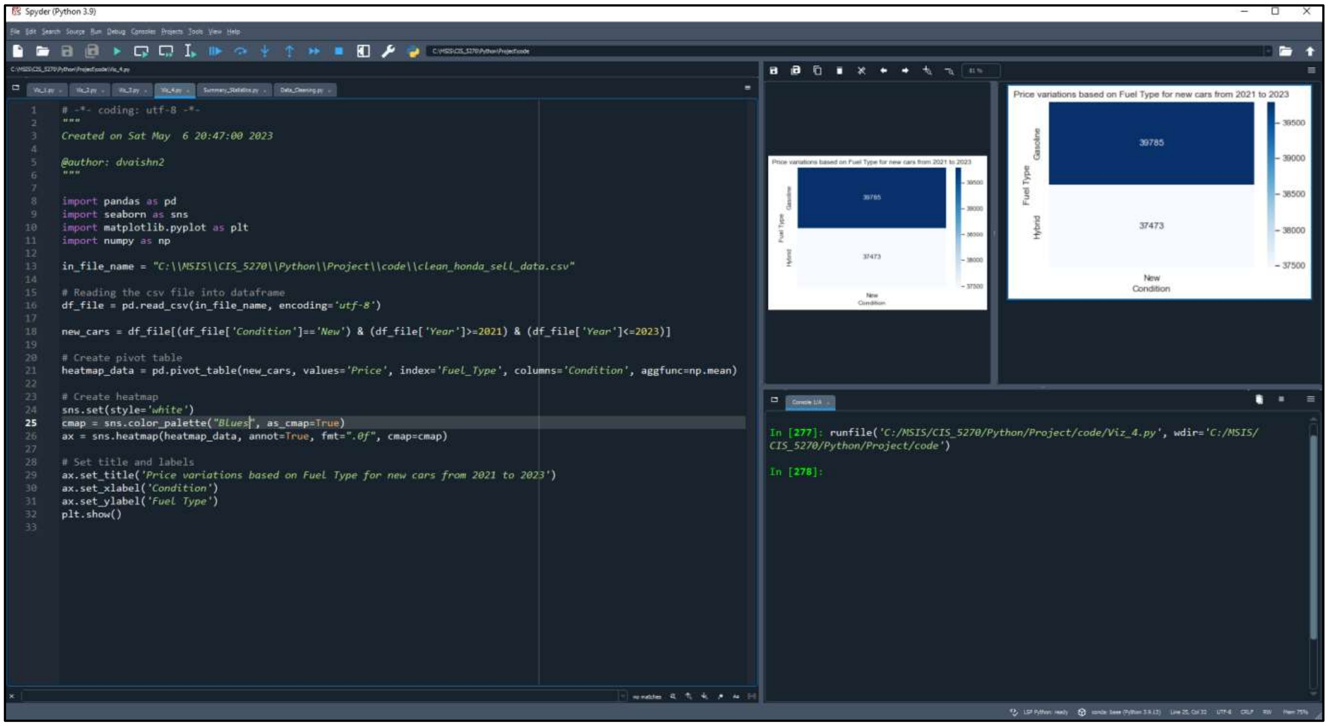
This plot can be useful in spotting patterns in the used car market, like the relationship between the year and mileage of a car and its price. It can also aid in identifying potential outliers, i.e., cars that are priced significantly higher or lower than the typical price range for a particular mileage and year combination.

7. Display how the prices are spread across various FUEL TYPES in the market for NEW cars for the last three years.



```
# -*- coding: utf-8 -*-
"""
Created on Sat May  6 20:47:00 2023

@author: dvaishn2
"""

import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np

in_file_name = "C:\\MSIS\\CIS_5270\\Python\\Project\\code\\clean_honda_sell_data.csv"

# Reading the csv file into dataframe
df_file = pd.read_csv(in_file_name, encoding='utf-8')

new_cars = df_file[(df_file['Condition']=='New') & (df_file['Year']>=2021) & (df_file['Year']<=2023)]

# Create pivot table
heatmap_data = pd.pivot_table(new_cars, values='Price', index='Fuel_Type', columns='Condition', aggfunc=np.mean)

# Create heatmap
sns.set(style='white')
cmap = sns.color_palette("Blues", as_cmap=True)
ax = sns.heatmap(heatmap_data, annot=True, fmt=".0f", cmap=cmap)

# Set title and labels
ax.set_title('Price variations based on Fuel Type for new cars from 2021 to 2023')
ax.set_xlabel('Condition')
ax.set_ylabel('Fuel Type')
plt.show()
```




Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\MSIS\CIS_5270\Python\Project\code
Viz_1.py
Viz_2.py
Viz_3.py
Viz_4.py
Summary_Statistics.py
Data_Cleaning.py
Console 1/A

```python
# -*- coding: utf-8 -*-
"""
Created on Sat May  6 20:47:00 2023

@author: dvaishn2
"""

import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np

in_file_name = "C:\\MSIS\\CIS_5270\\Python\\Project\\code\\clean_honda_sell_data.csv"

# Reading the csv file into dataframe
df_file = pd.read_csv(in_file_name, encoding='utf-8')

new_cars = df_file[(df_file['Condition']=='New') & (df_file['Year']>=2021) & (df_file['Year']<=2023)]

# Create pivot table
heatmap_data = pd.pivot_table(new_cars, values='Price', index='Fuel_Type', columns='Condition', aggfunc=np.mean)

# Create heatmap
sns.set(style='white')
cmap = sns.color_palette("Blues", as_cmap=True)
ax = sns.heatmap(heatmap_data, annot=True, fmt=".0f", cmap=cmap)

# Set title and labels
ax.set_title('Price variations based on Fuel Type for new cars from 2021 to 2023')
ax.set_xlabel('Condition')
ax.set_ylabel('Fuel Type')
plt.show()
```

```
In [277]: runfile('C:/MSIS/CIS_5270/Python/Project/code/Viz_4.py', wdir='C:/MSIS/CIS_5270/Python/Project/code')

In [278]:
```

Insights:

The heatmap plot in the graph showcases the average price of new cars for different fuel types and conditions, thereby providing a quick and easy way to compare the prices across various categories.

The heat map is color-coded, with the lighter shades indicating lower values and the darker shades indicating higher values.

By analyzing the heatmap, we can get insights into which fuel type is preferred for new cars and its influence on price.

For instance, the graph could reveal that hybrid cars tend to have a higher price point than gasoline cars in the new car market.

Additionally, we can observe the price variations among different conditions for a particular fuel type.

Overall, this graph provides an effective way to understand the relationship between the average price of new cars and fuel types under different conditions, giving us valuable insights into market trends and consumer preferences.