CTEC 298 Final Project

By: Saniya Isaac, Andrea Redman

CTEC 298

Introduction

- This presentation covers what we learned and completed in CTEC 128 and CTEC 298.
- We'll begin with a quick overview of both courses and how they connect.
- You'll see summaries of our CTEC 128 projects, including the data we used and key findings.
- Then we'll walk through the materials we submitted for CTEC 298.
- We'll also describe the Python and Tableau plots we created and explain what they show.
- Finally, we'll end with a summary of what we learned and the skills we gained from both courses.

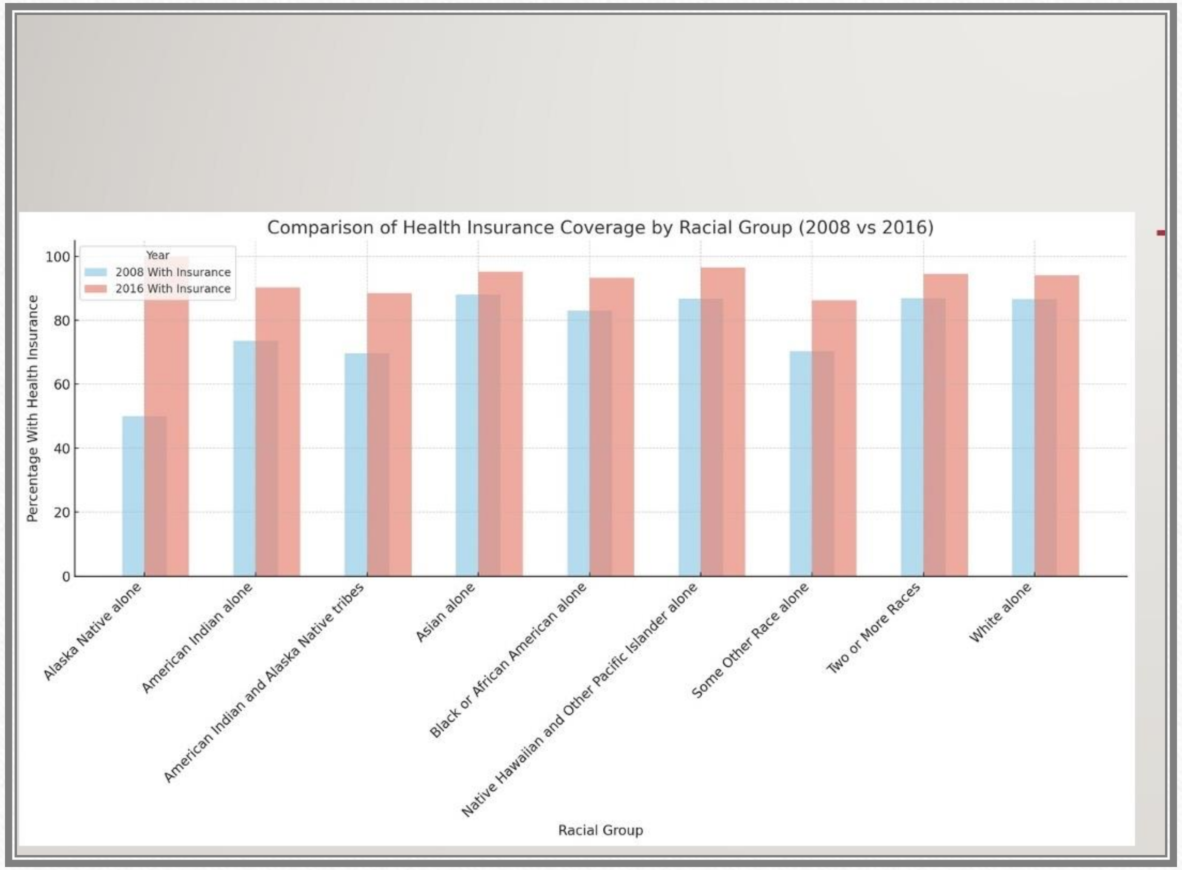
CTEC 128 (Group Leader)

- Studied whether Black drivers in Maryland are more likely than White drivers to be pulled over for expired license plates.
- Used traffic stop data from Maryland's 2017 Race-Based Traffic Stop Data Dashboard.
- Focused on key features: race/ethnicity, stop reasons, and population demographics.
- Worked with raw data from official PDF reports and converted the figures into structured data tables.
- Performed Exploratory Data Analysis (EDA) to explore differences in stop percentages by race.
- Found that Black drivers made up 29% of Maryland's population but accounted for 40% of traffic stops, while White drivers were 51% of the population but only 44% of stops.
- Results suggest that Black drivers are stopped more often than expected based on their population share, pointing to possible racial bias in traffic enforcement.

CTEC 128 (Member 1)

- Studied the impact of the Affordable Care Act (ACA) on access to health coverage in California.
- Used health data from 2008 (pre-ACA) and 2016 (post-ACA) to compare changes over time.
- Focused on key features: income level, race/ethnicity, age group (18–64), and health insurance status.
- Worked with raw data from the American Community Survey and cleaned it for analysis.
- Applied data wrangling (fixing missing values, combining features, categorizing columns).
- Performed Exploratory Data Analysis (EDA) to find trends and patterns in the data.
- Created visualizations (bar charts, age comparisons, race vs coverage graphs) to highlight findings.
- Found that health insurance rates increased after the ACA, especially for low-income and minority groups.

Health Insurance Coverage by Race (2008 vs. 2016)



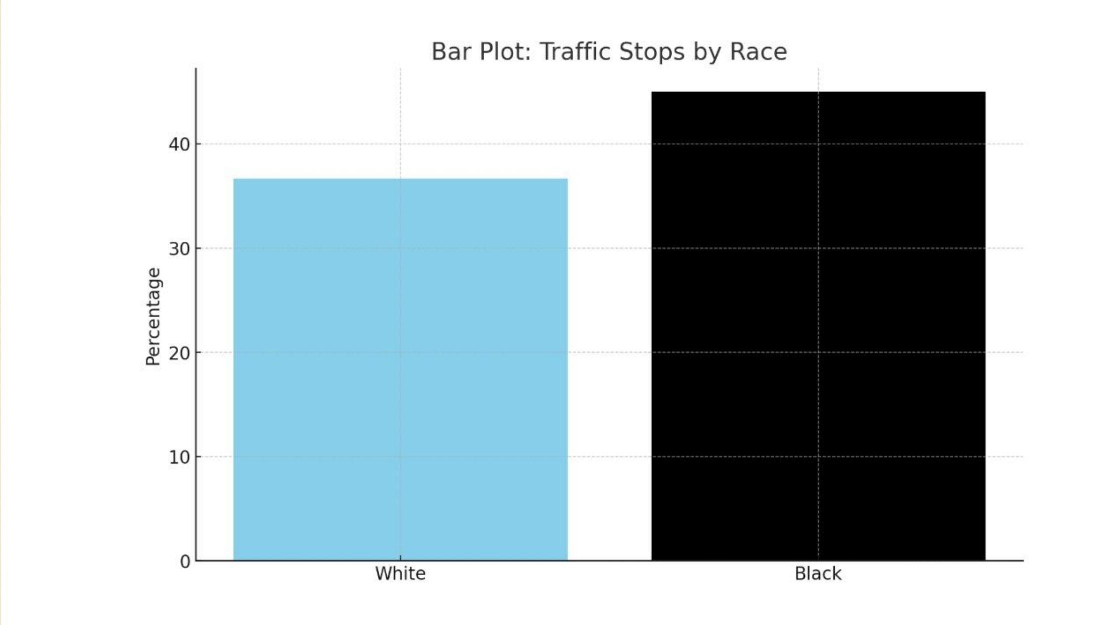
CTEC 298 Material Submitted (Group Leader)

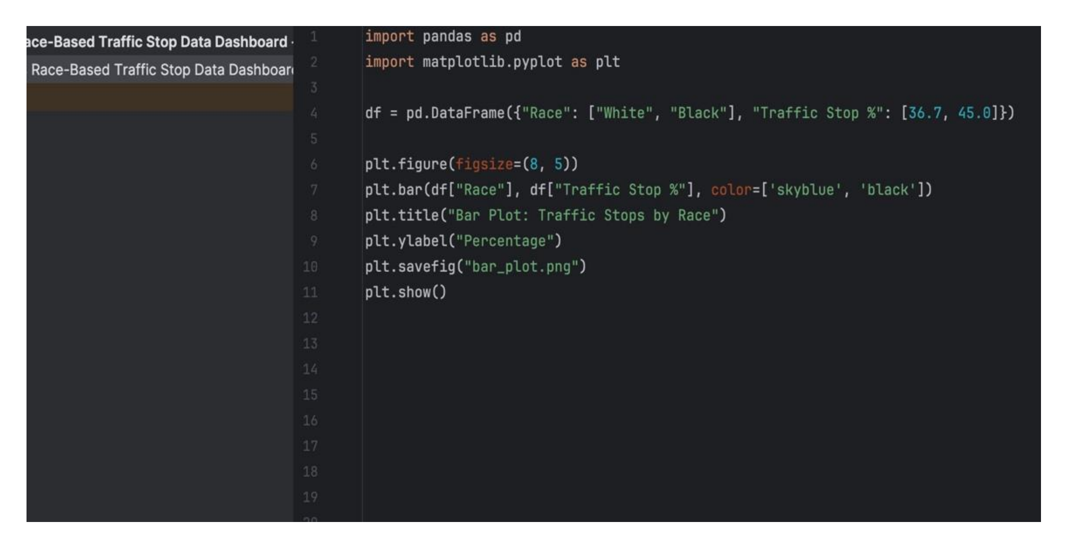
- Completed multiple assignments focused on Python programming, data analysis, and data visualization.
- Practiced basic Python functions through the LearnPython tutorials.
- Used Jupyter Notebook to run and test Python code step by step, which helped organize and understand the code better.
- Completed the Matplotlib assignment, creating six different types of plots.
- Designed and presented a bar plot presentation explaining how to create bar plots and why they are useful in data analysis.
- Learned to use Tableau to build two visual charts for presenting data.
- Used GitHub and Google Drive to store, organize, and back up all assignments and code.
- Gained hands-on experience with real data and professional tools for data visualization and analysis.

CTEC 298 Material Submitted (Member 1)

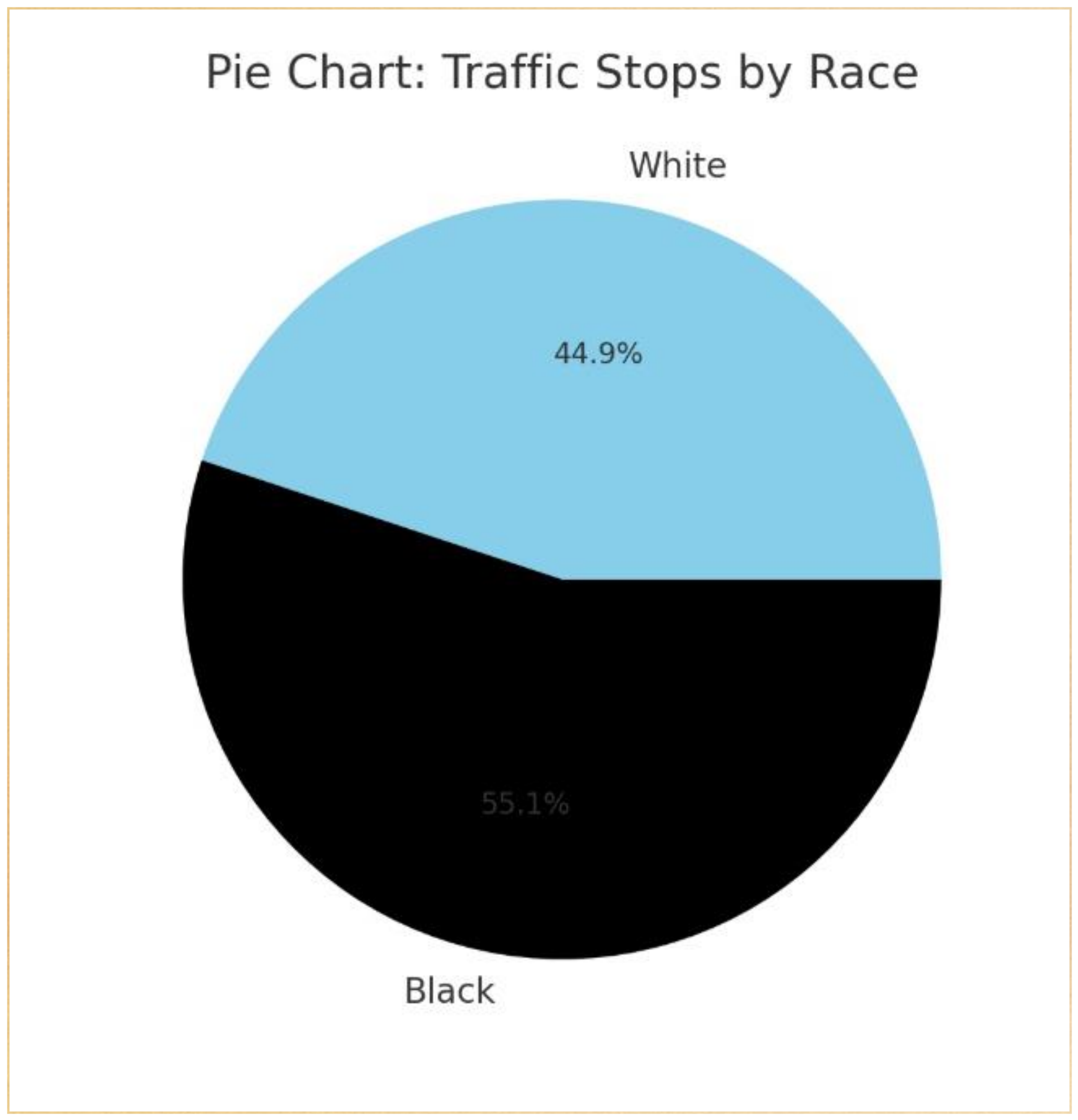
- Python Basics (LearnPython1 & 2):
  Completed beginner lessons to learn how to use Python for reading files, using variables, loops, and writing simple code.
- Jupyter Notebook Coding:
  Practiced writing and testing Python code in a step-by-step format using Jupyter Notebooks, which made it easier to organize and run code blocks.
- Matplotlib (6 Plots):
  Created six different chart types using Matplotlib: bar plot, pie chart, histogram, scatter plot, stack plot, and multiplot, based on real health data.
- Scatter Plot Presentation:
  Made a presentation that explained what a scatter plot is, how it shows relationships between variables, and why it's useful in data science.
- Pandas Assignment (DataFrame):
  Used Pandas to load the 2008 and 2016 health data, turned it into a DataFrame, and explored it using functions like .head() and .info().
- Tableau Plots (Bar Chart & Circle Chart):
  Built two visualizations in Tableau one comparing racial group counts, and another showing health insurance coverage in a simple, clear way.
- GitHub Used to Store Work:
  Uploaded all assignments to GitHub to organize, share, and back up all my project files and code.

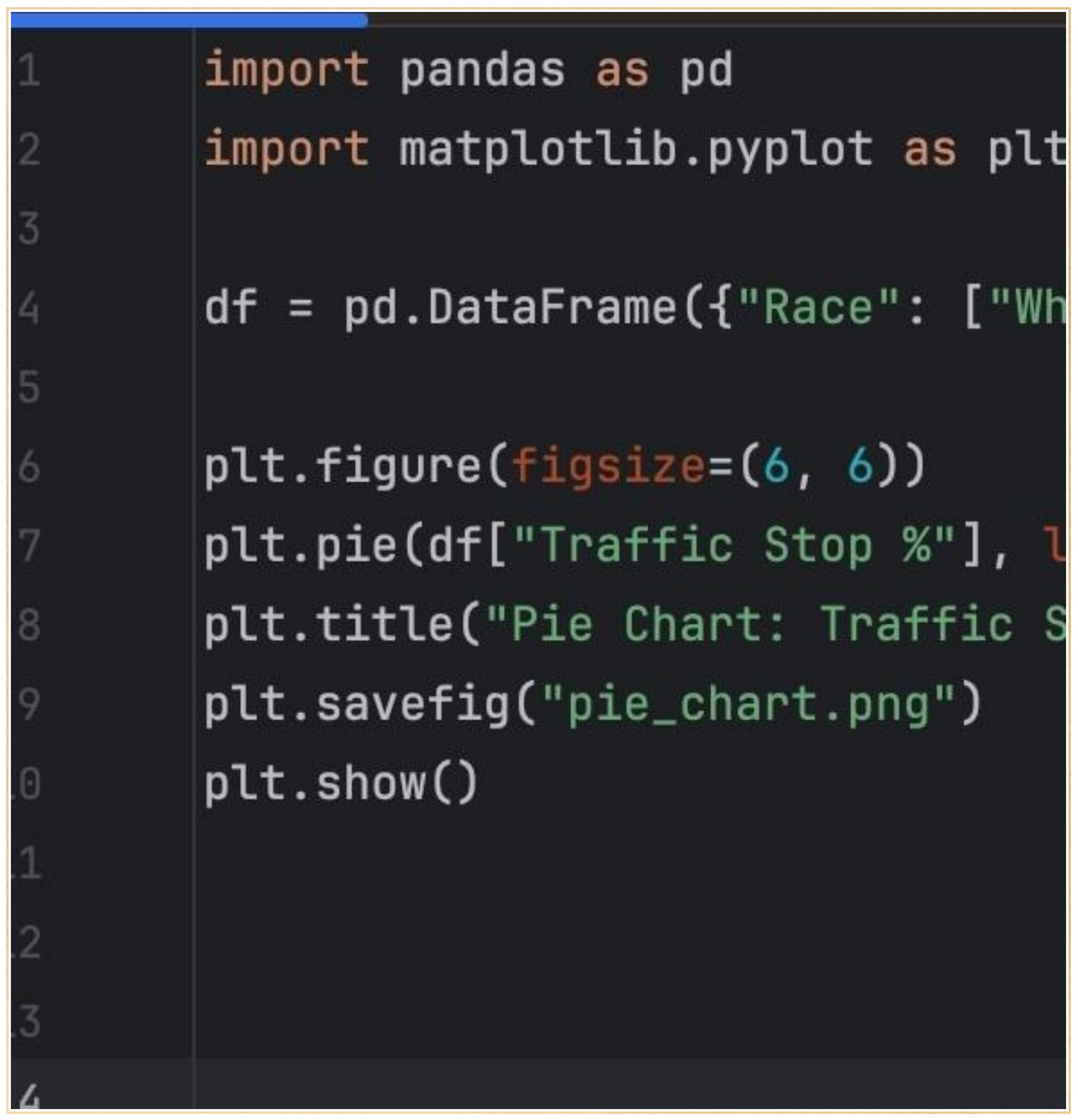
Plot Deliverables (Group Leader)



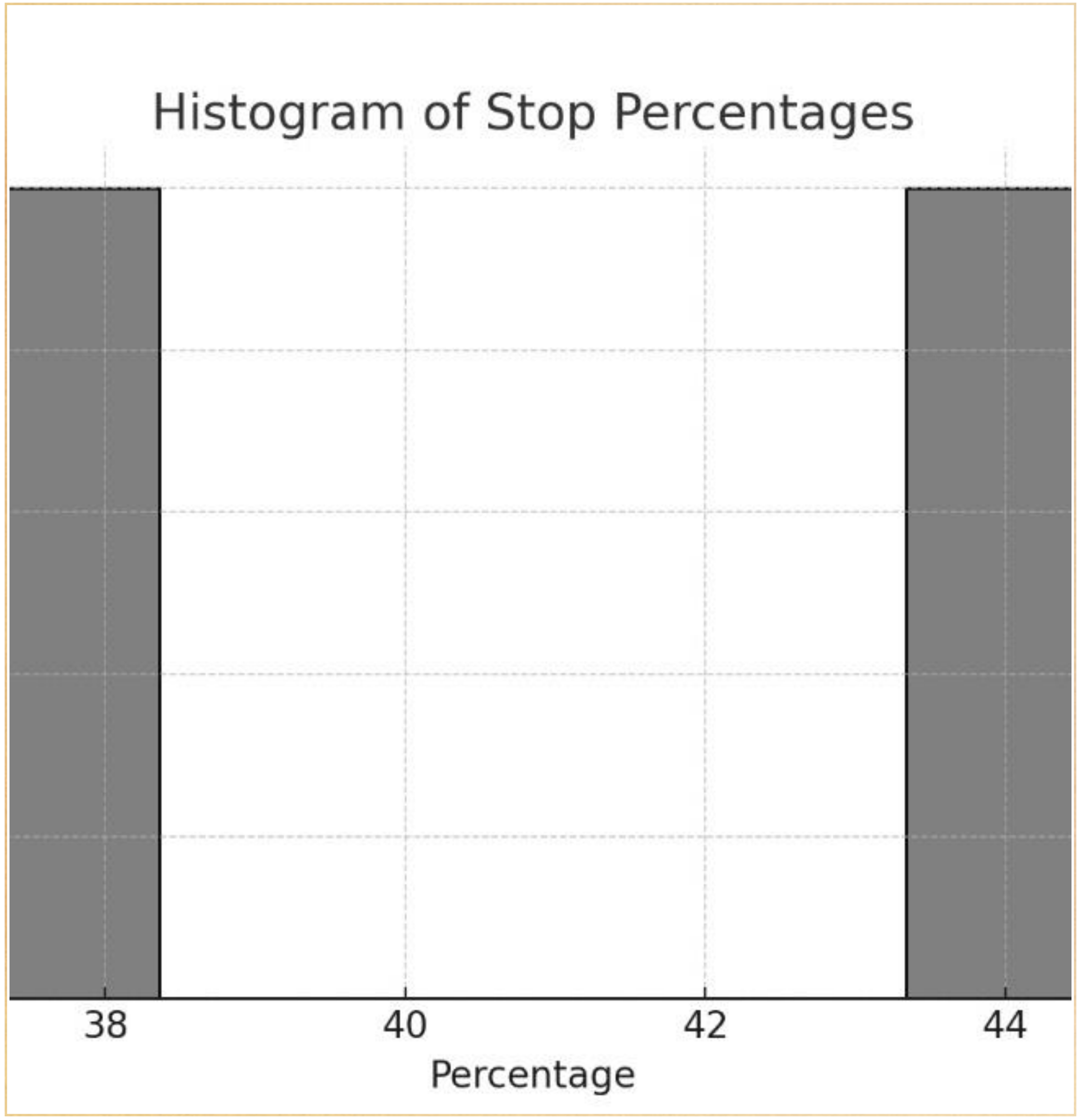

ace-Based Traffic Stop Data Dashboard
Race-Based Traffic Stop Data Dashboar

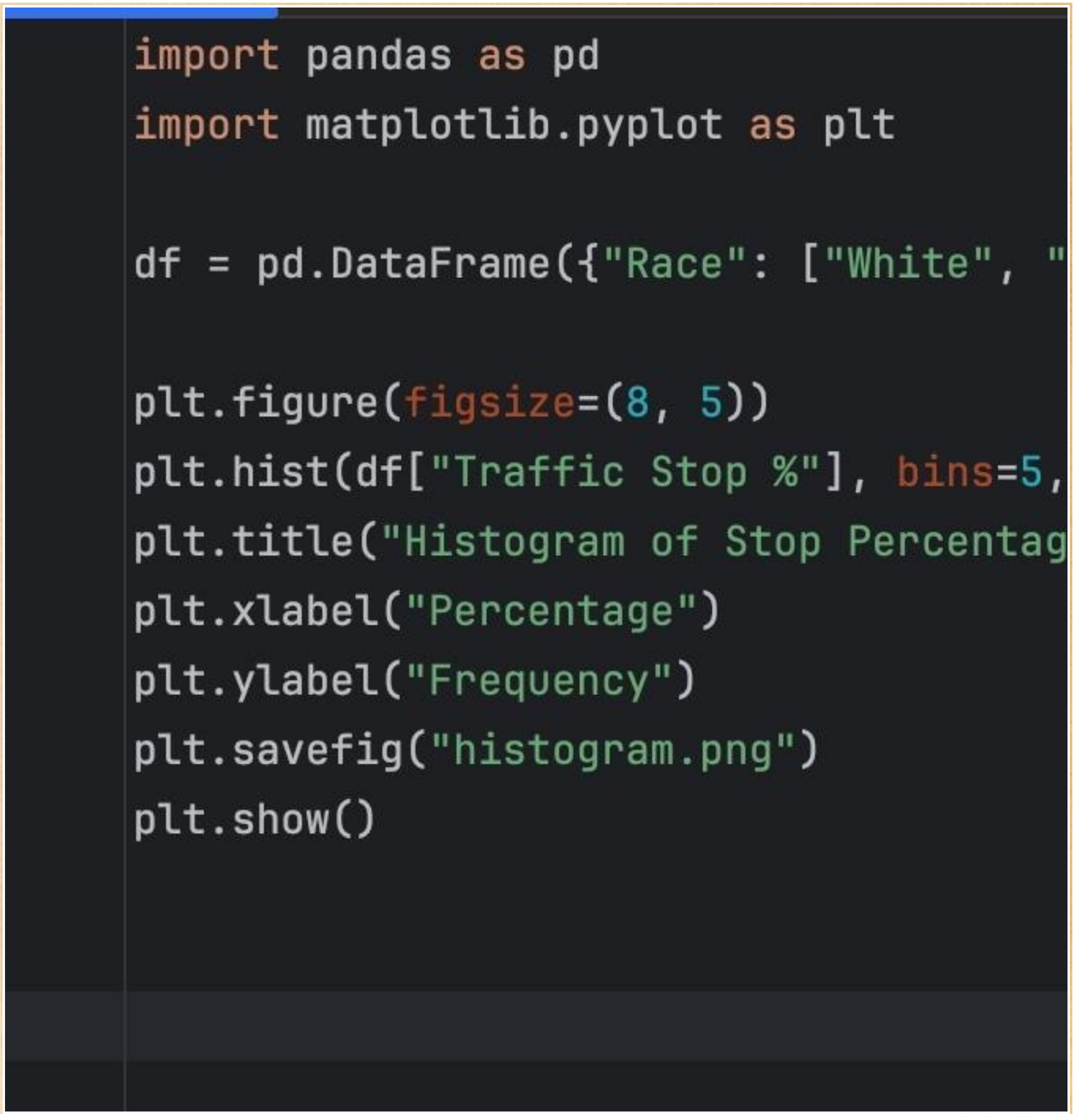
```python
import pandas as pd
import matplotlib.pyplot as plt

df = pd.DataFrame({"Race": ["White", "Black"], "Traffic Stop %": [36.7, 45.0]})

plt.figure(figsize=(8, 5))
plt.bar(df["Race"], df["Traffic Stop %"], color=['skyblue', 'black'])
plt.title("Bar Plot: Traffic Stops by Race")
plt.ylabel("Percentage")
plt.savefig("bar_plot.png")
plt.show()
```



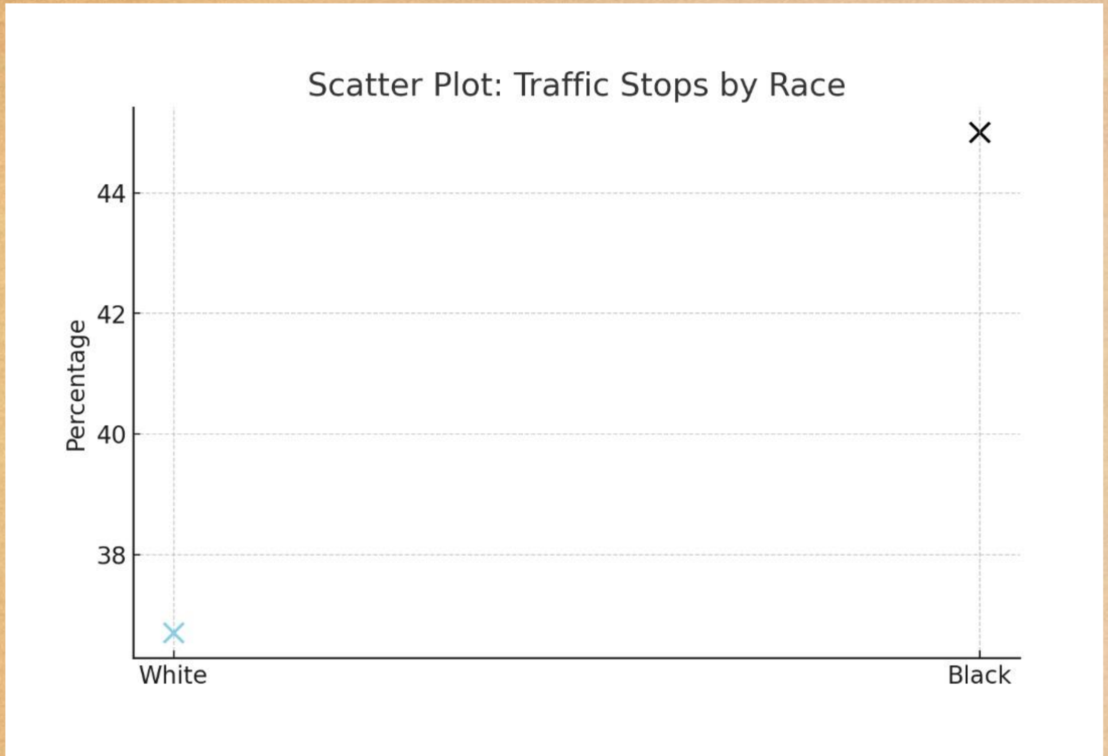


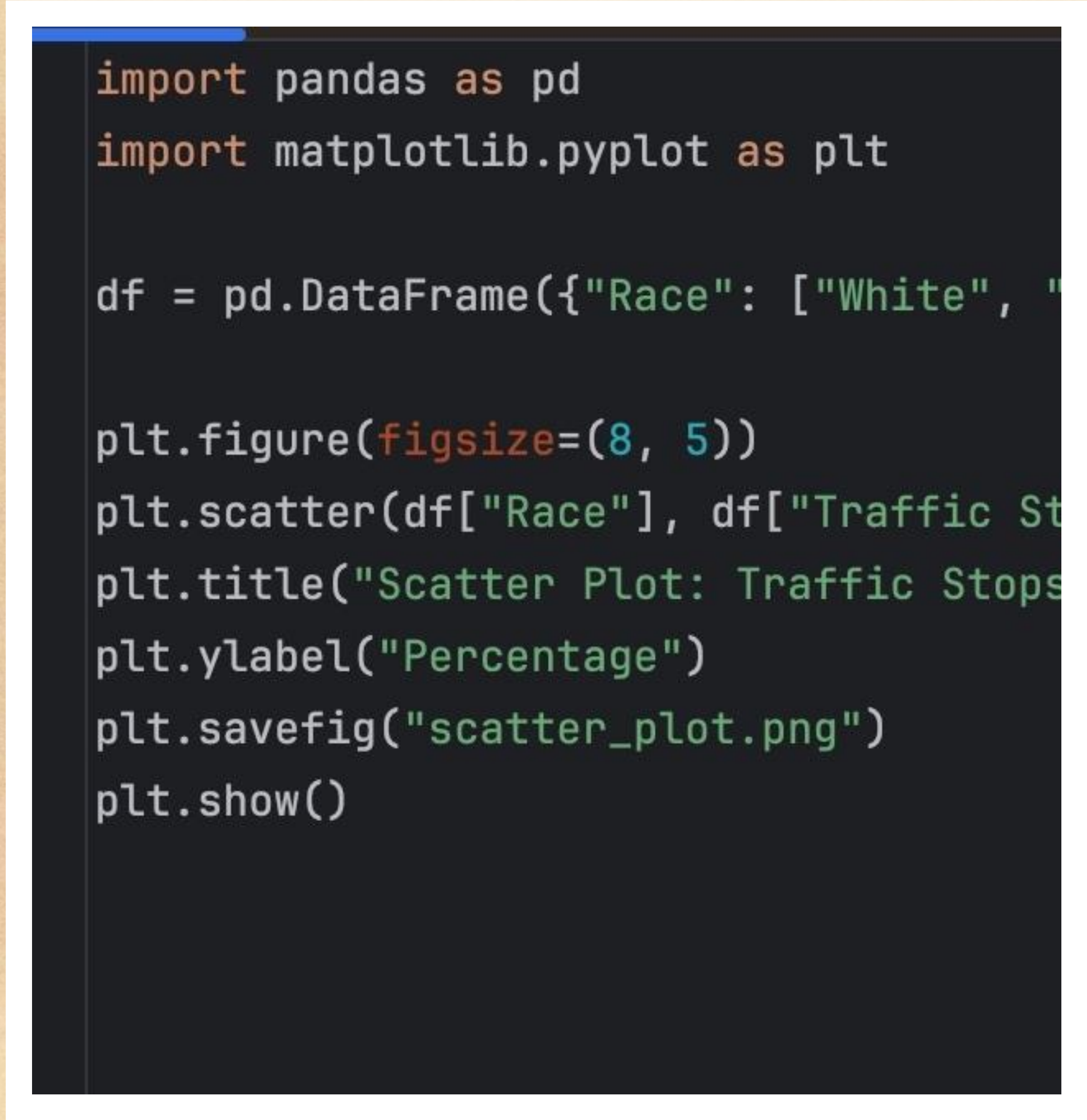
```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.DataFrame({"Race": ["Wh

plt.figure(figsize=(6, 6))
plt.pie(df["Traffic Stop %"], l
plt.title("Pie Chart: Traffic S
plt.savefig("pie_chart.png")
plt.show()
```



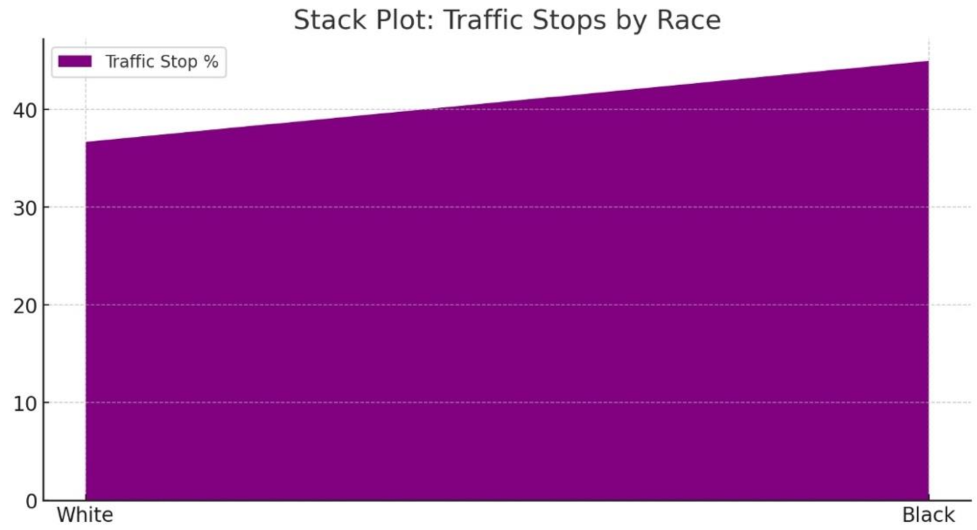


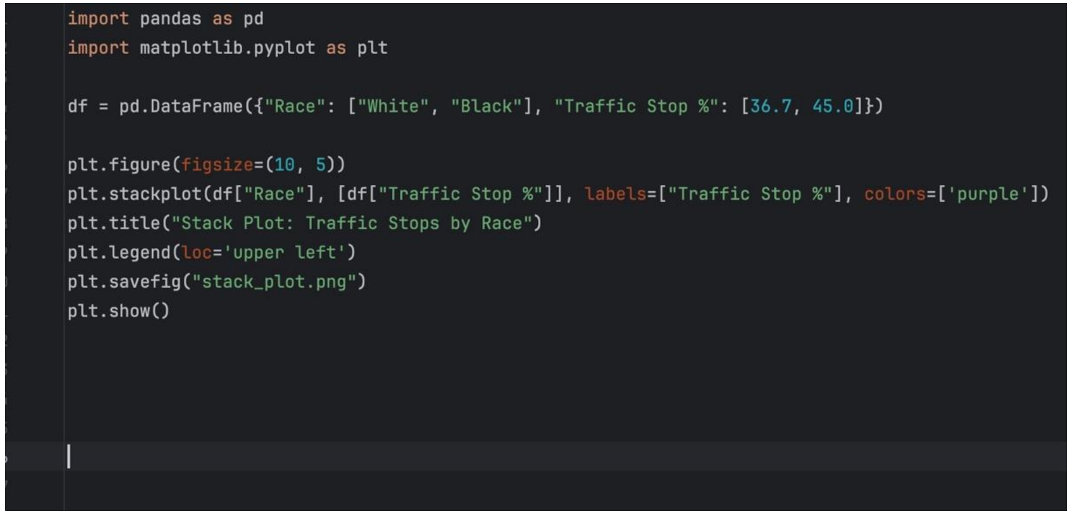
```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.DataFrame({"Race": ["White", "

plt.figure(figsize=(8, 5))
plt.hist(df["Traffic Stop %"], bins=5,
plt.title("Histogram of Stop Percentag
plt.xlabel("Percentage")
plt.ylabel("Frequency")
plt.savefig("histogram.png")
plt.show()
```

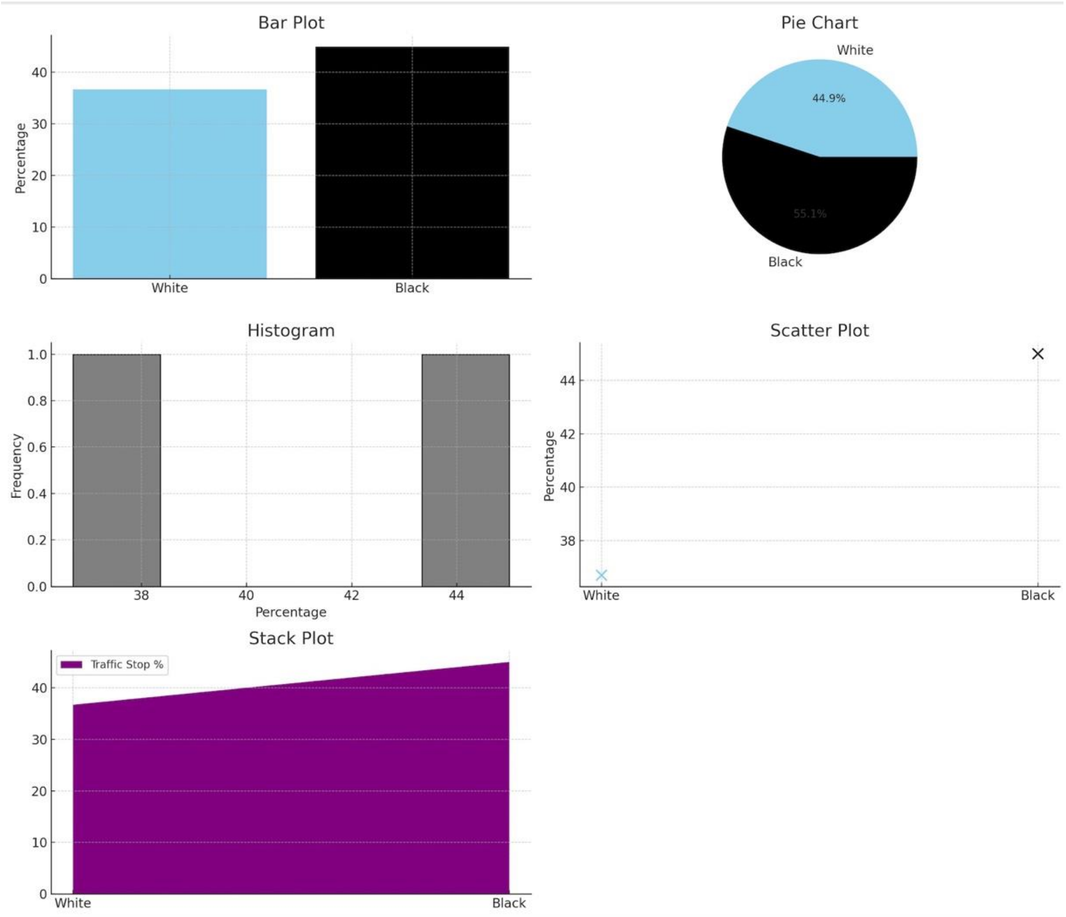


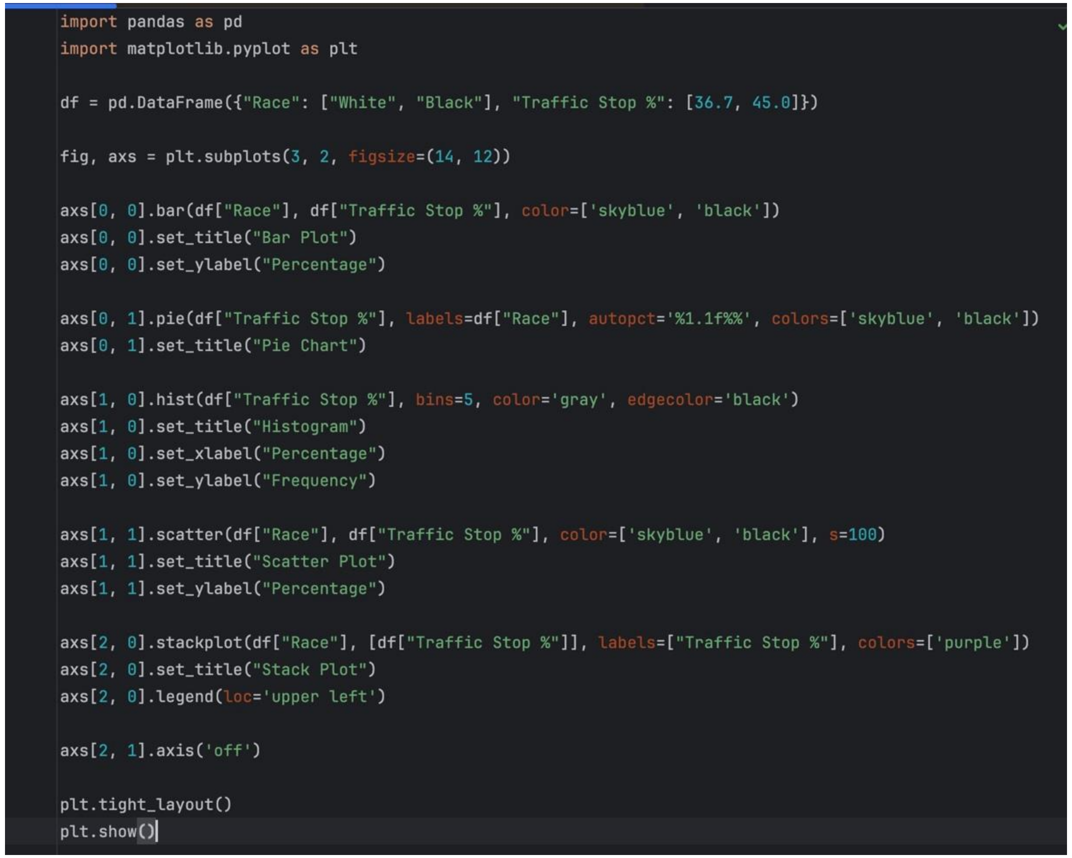


```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.DataFrame({"Race": ["White", "

plt.figure(figsize=(8, 5))
plt.scatter(df["Race"], df["Traffic St
plt.title("Scatter Plot: Traffic Stops
plt.ylabel("Percentage")
plt.savefig("scatter_plot.png")
plt.show()
```



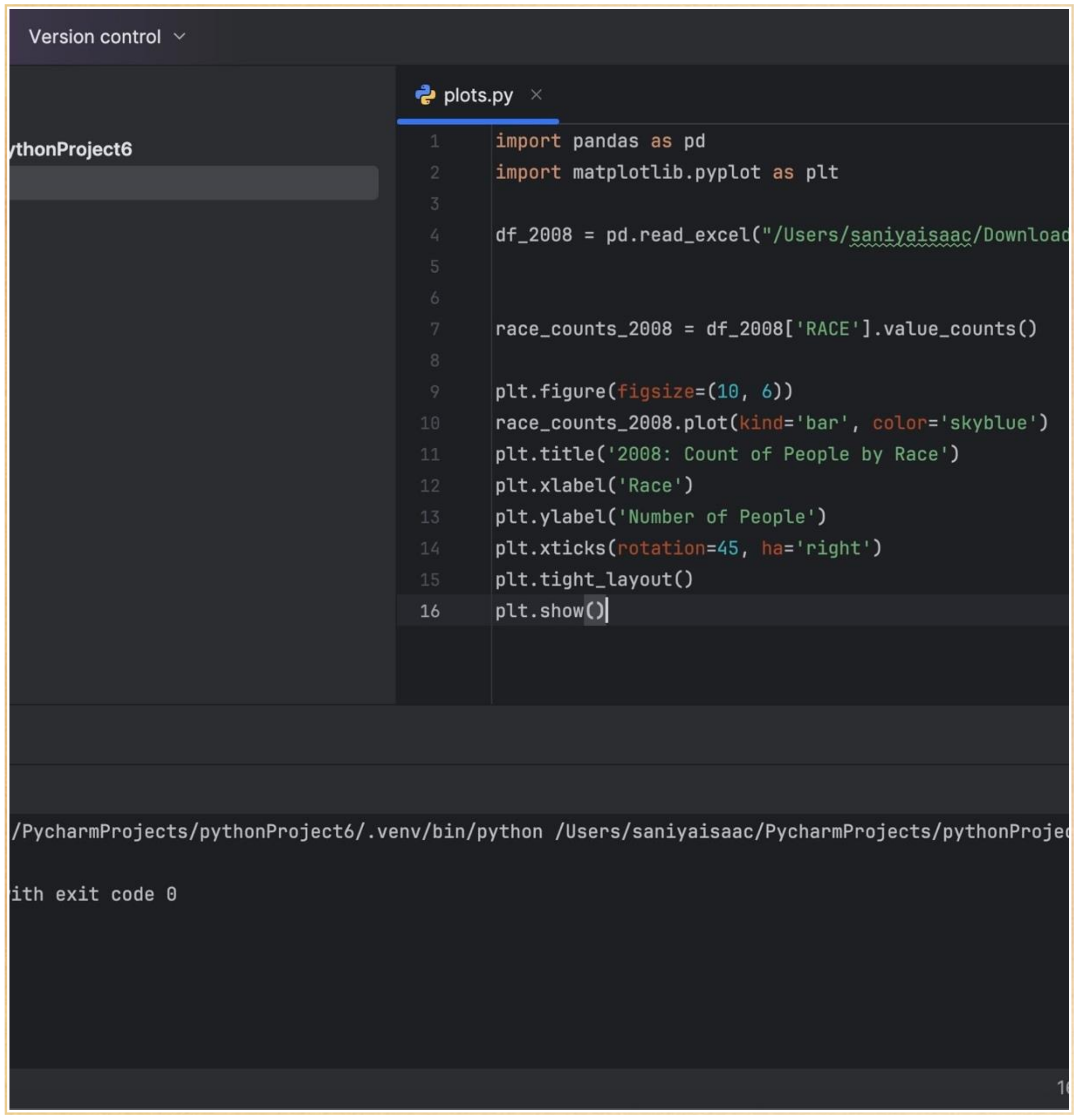

```python
import pandas as pd
import matplotlib.pyplot as plt

df = pd.DataFrame({"Race": ["White", "Black"], "Traffic Stop %": [36.7, 45.0]})

plt.figure(figsize=(10, 5))
plt.stackplot(df["Race"], [df["Traffic Stop %"]], labels=["Traffic Stop %"], colors=['purple'])
plt.title("Stack Plot: Traffic Stops by Race")
plt.legend(loc='upper left')
plt.savefig("stack_plot.png")
plt.show()
```





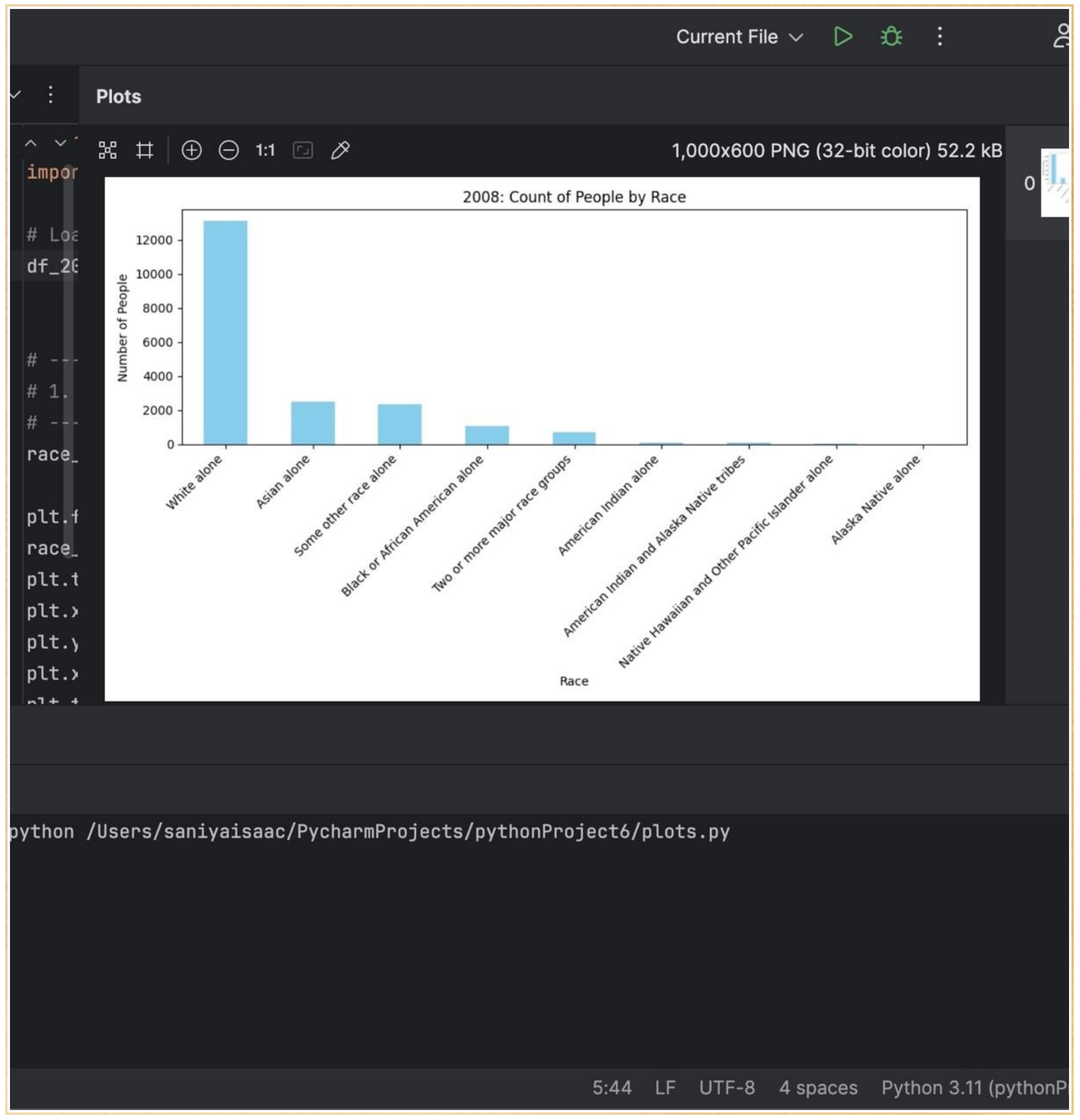
```python
import pandas as pd
import matplotlib.pyplot as plt

df = pd.DataFrame({"Race": ["White", "Black"], "Traffic Stop %": [36.7, 45.0]})

fig, axs = plt.subplots(3, 2, figsize=(14, 12))

axs[0, 0].bar(df["Race"], df["Traffic Stop %"], color=['skyblue', 'black'])
axs[0, 0].set_title("Bar Plot")
axs[0, 0].set_ylabel("Percentage")

axs[0, 1].pie(df["Traffic Stop %"], labels=df["Race"], autopct='%1.1f%%', colors=['skyblue', 'black'])
axs[0, 1].set_title("Pie Chart")

axs[1, 0].hist(df["Traffic Stop %"], bins=5, color='gray', edgecolor='black')
axs[1, 0].set_title("Histogram")
axs[1, 0].set_xlabel("Percentage")
axs[1, 0].set_ylabel("Frequency")

axs[1, 1].scatter(df["Race"], df["Traffic Stop %"], color=['skyblue', 'black'], s=100)
axs[1, 1].set_title("Scatter Plot")
axs[1, 1].set_ylabel("Percentage")

axs[2, 0].stackplot(df["Race"], [df["Traffic Stop %"]], labels=["Traffic Stop %"], colors=['purple'])
axs[2, 0].set_title("Stack Plot")
axs[2, 0].legend(loc='upper left')

axs[2, 1].axis('off')

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

Plot Deliverables (Member 1)



```python
import pandas as pd
import matplotlib.pyplot as plt

df_2008 = pd.read_excel("/Users/saniyaisaac/Download


race_counts_2008 = df_2008['RACE'].value_counts()

plt.figure(figsize=(10, 6))
race_counts_2008.plot(kind='bar', color='skyblue')
plt.title('2008: Count of People by Race')
plt.xlabel('Race')
plt.ylabel('Number of People')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```

```
/PycharmProjects/pythonProject6/.venv/bin/python /Users/saniyaisaac/PycharmProjects/pythonProje

ith exit code 0
```

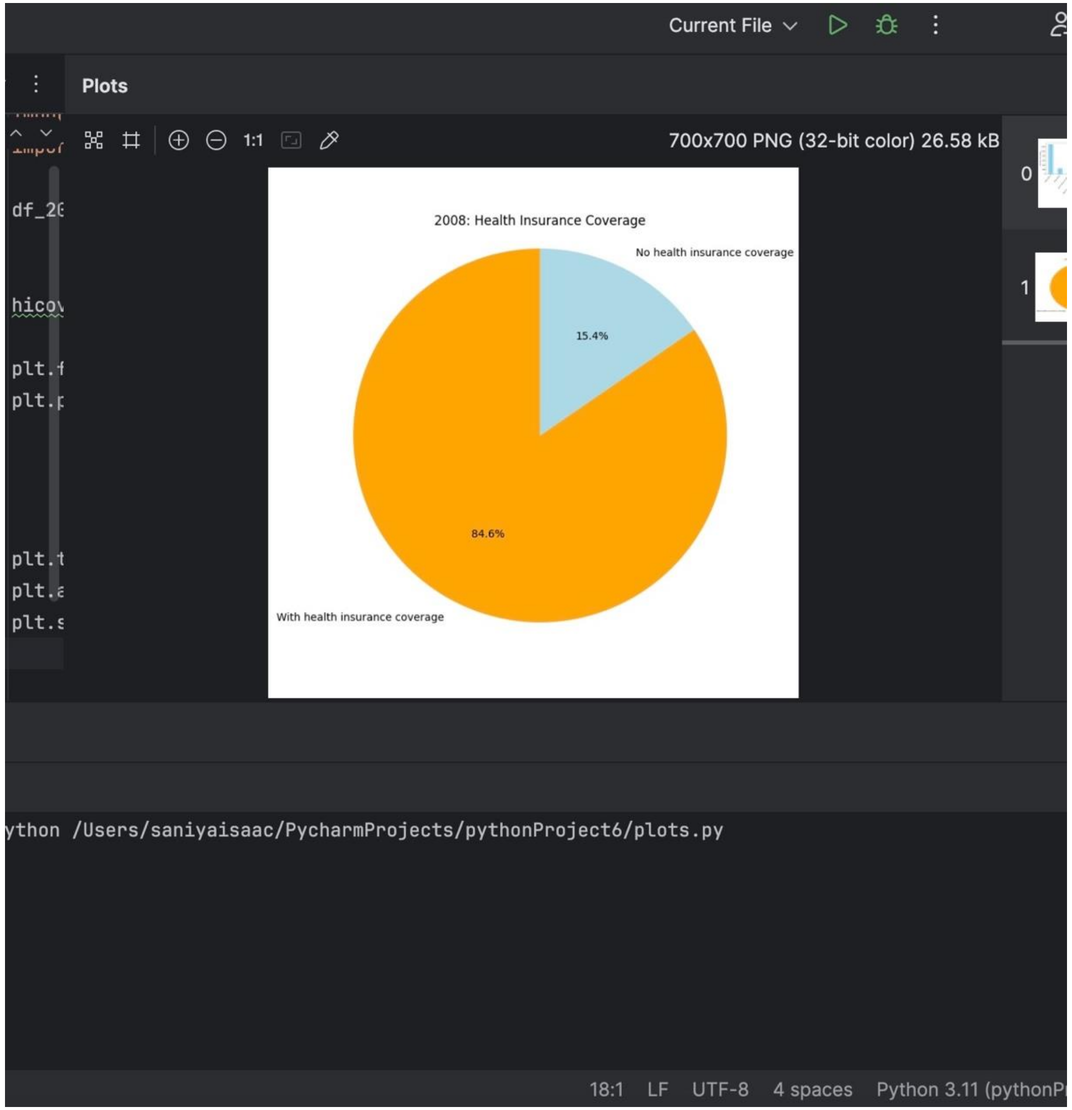


Current File
Plots
1:1
1,000x600 PNG (32-bit color) 52.2 kB
2008: Count of People by Race
Number of People
12000
10000
8000
6000
4000
2000
0
White alone
Asian alone
Some other race alone
Black or African American alone
Two or more major race groups
American Indian alone
American Indian and Alaska Native tribes
Native Hawaiian and Other Pacific Islander alone
Alaska Native alone
Race

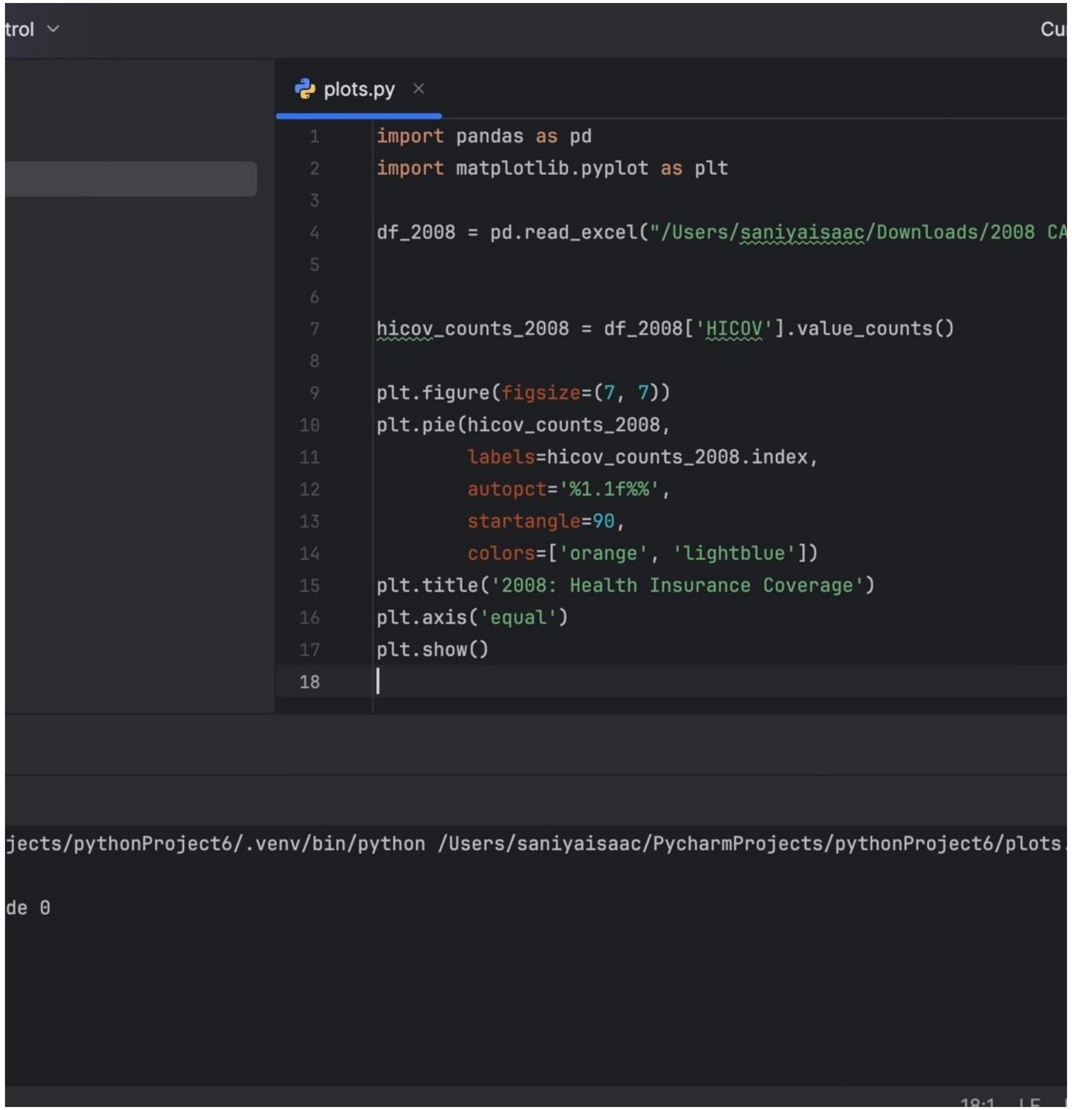
```
impor

# Loa
df_20


# ---
# 1.
# ---
race_

plt.f
race_
plt.t
plt.x
plt.y
plt.x
```

```
python /Users/saniyaisaac/PycharmProjects/pythonProject6/plots.py
```

5:44 LF UTF-8 4 spaces Python 3.11 (pythonP

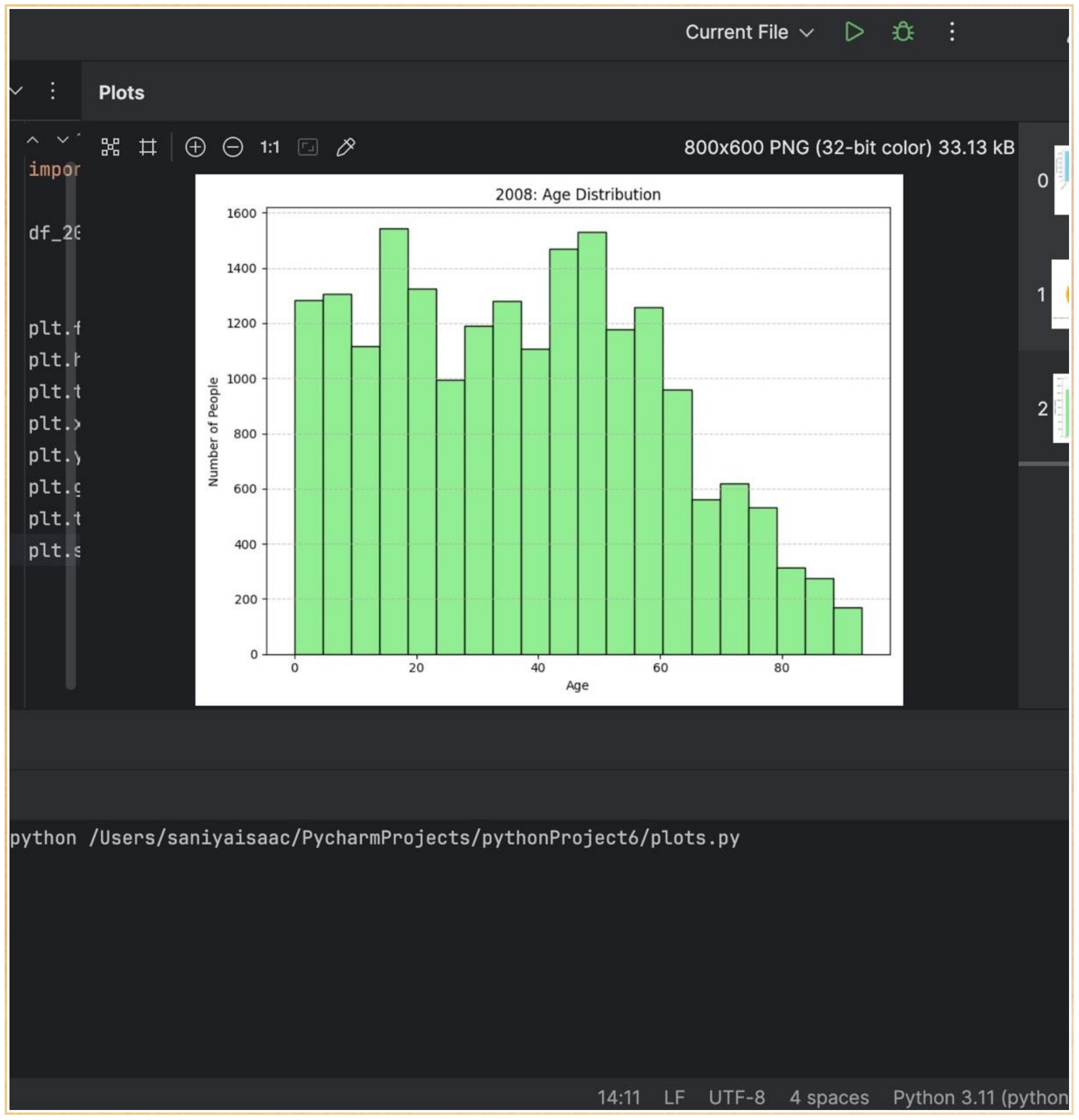

Current File
Plots
700x700 PNG (32-bit color) 26.58 kB
1:1
2008: Health Insurance Coverage
No health insurance coverage
15.4%
84.6%
With health insurance coverage
0
1
18:1
LF
UTF-8
4 spaces
Python 3.11 (pythonP

```
ython /Users/saniyaisaac/PycharmProjects/pythonProject6/plots.py
```

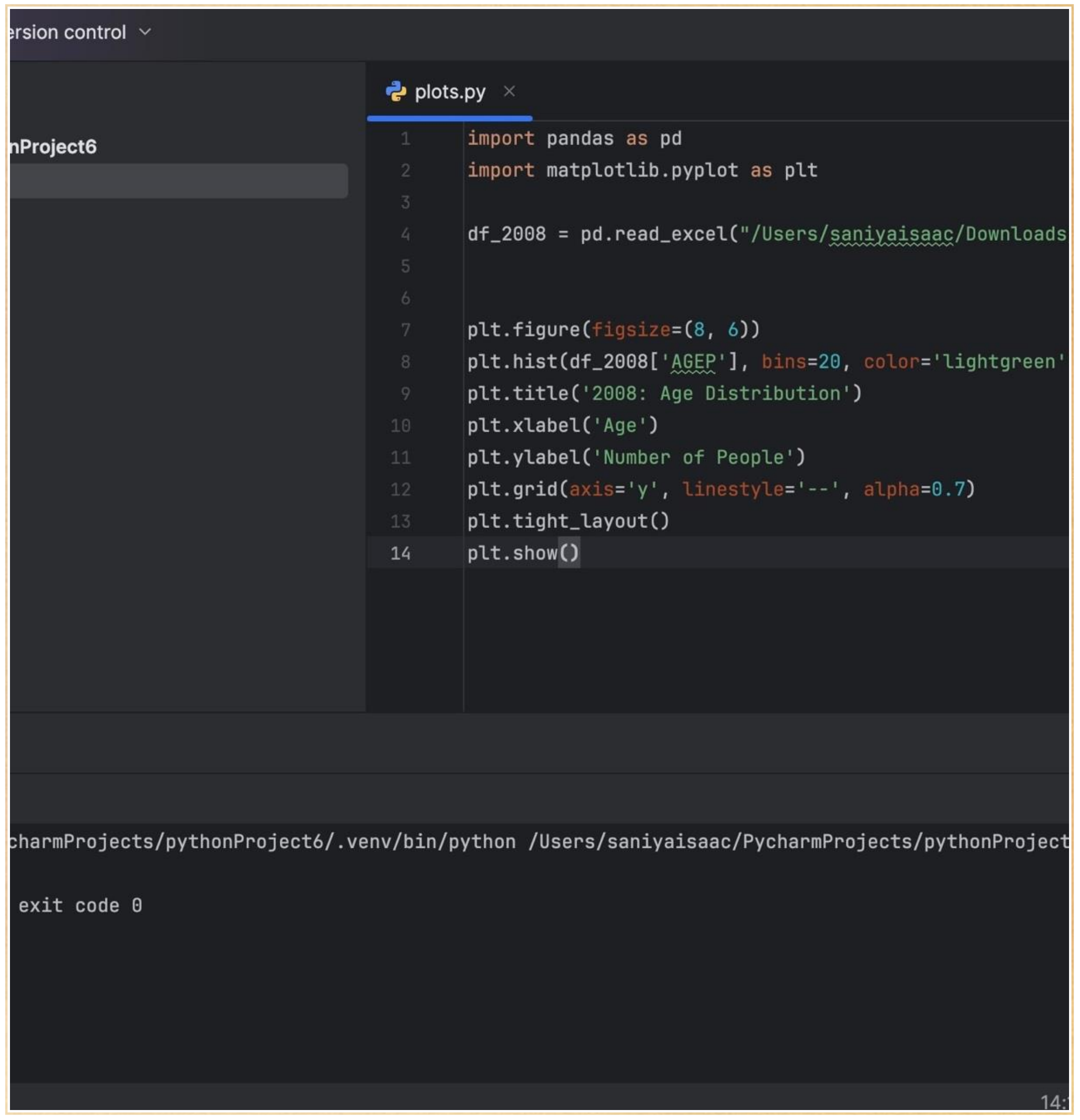


```
import pandas as pd
import matplotlib.pyplot as plt

df_2008 = pd.read_excel("/Users/saniyaisaac/Downloads/2008 CA


hicov_counts_2008 = df_2008['HICOV'].value_counts()

plt.figure(figsize=(7, 7))
plt.pie(hicov_counts_2008,
        labels=hicov_counts_2008.index,
        autopct='%1.1f%%',
        startangle=90,
        colors=['orange', 'lightblue'])
plt.title('2008: Health Insurance Coverage')
plt.axis('equal')
plt.show()
```

```
jects/pythonProject6/.venv/bin/python /Users/saniyaisaac/PycharmProjects/pythonProject6/plots.

de 0
```

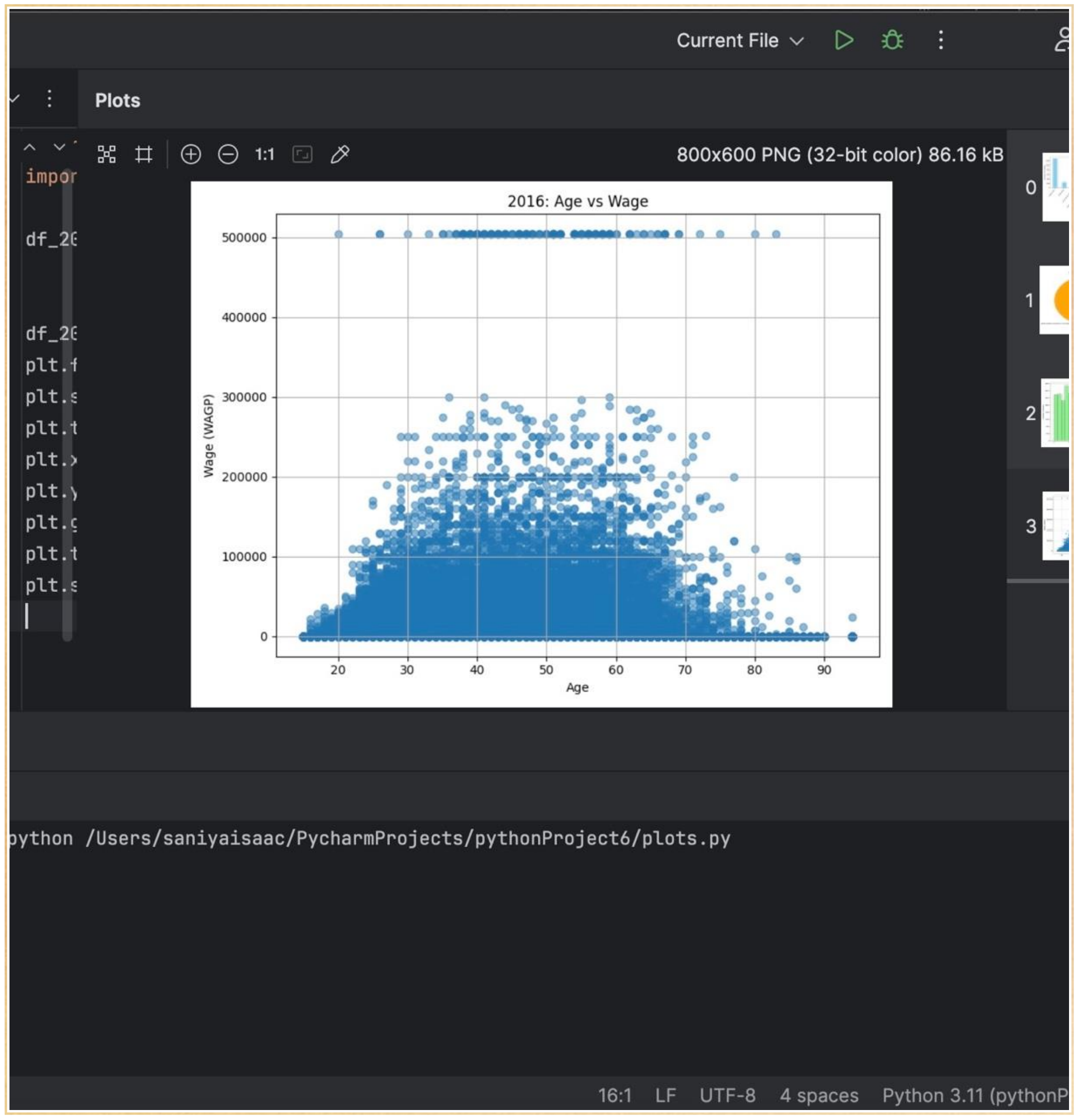


```
python /Users/saniyaisaac/PycharmProjects/pythonProject6/plots.py
```



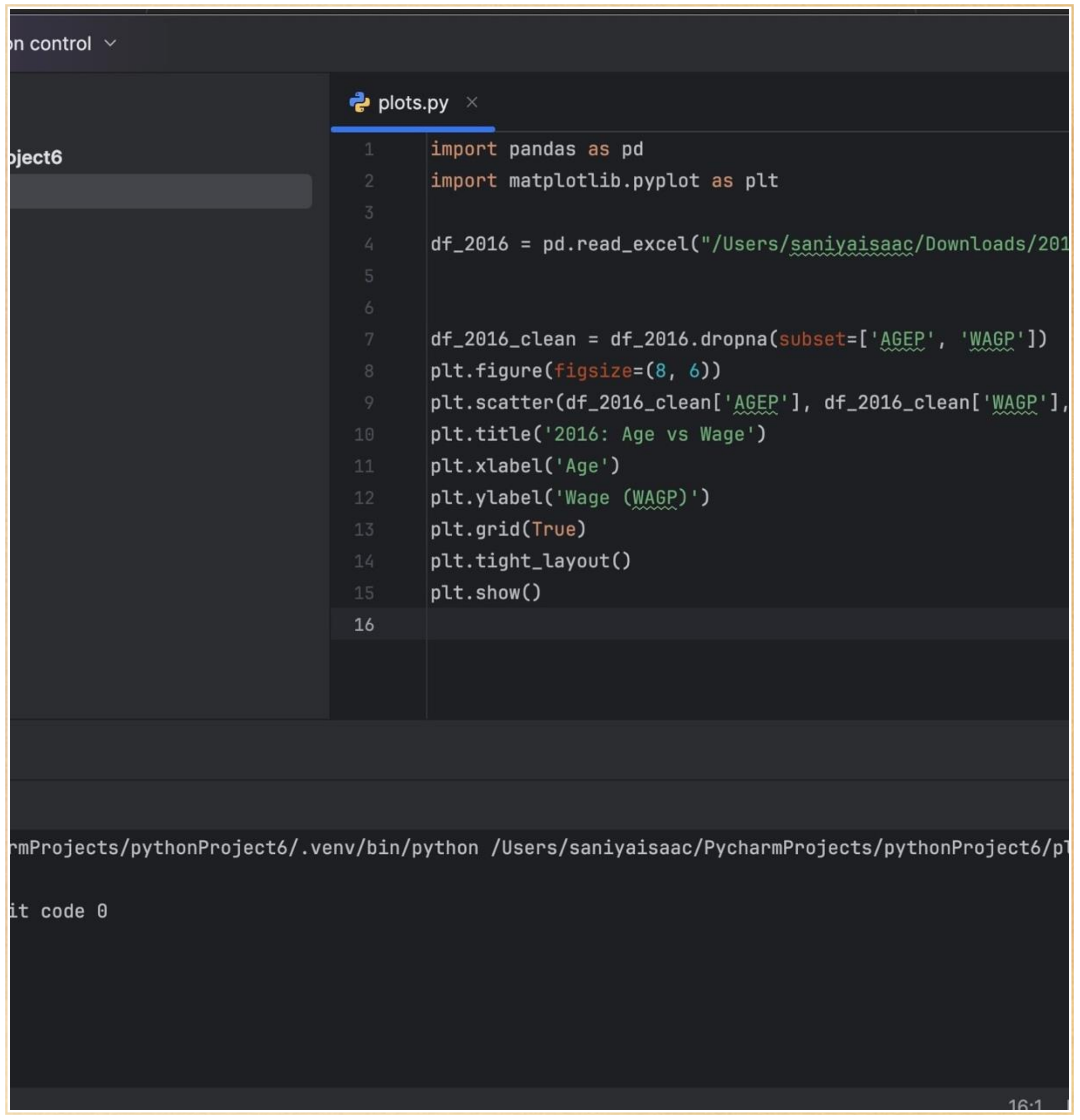
```
import pandas as pd
import matplotlib.pyplot as plt

df_2008 = pd.read_excel("/Users/saniyaisaac/Downloads


plt.figure(figsize=(8, 6))
plt.hist(df_2008['AGEP'], bins=20, color='lightgreen'
plt.title('2008: Age Distribution')
plt.xlabel('Age')
plt.ylabel('Number of People')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```

```
charmProjects/pythonProject6/.venv/bin/python /Users/saniyaisaac/PycharmProjects/pythonProject

 exit code 0
```



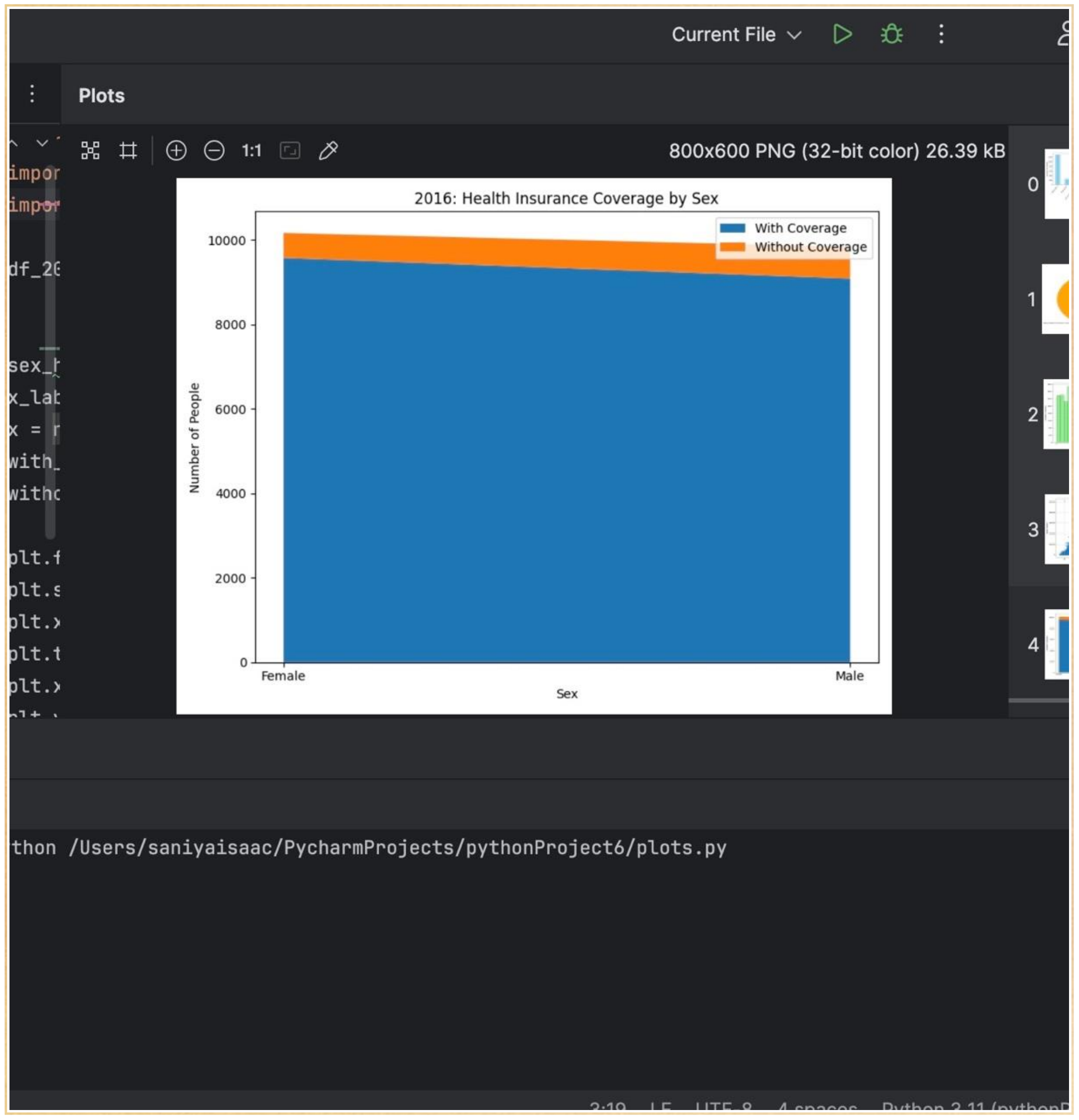
```
impor

df_20


df_20
plt.f
plt.s
plt.t
plt.x
plt.y
plt.g
plt.t
plt.s
```

```
python /Users/saniyaisaac/PycharmProjects/pythonProject6/plots.py
```



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

df_2016 = pd.read_excel("/Users/saniyaisaac/Downloads/201


df_2016_clean = df_2016.dropna(subset=['AGEP', 'WAGP'])
plt.figure(figsize=(8, 6))
plt.scatter(df_2016_clean['AGEP'], df_2016_clean['WAGP'],
plt.title('2016: Age vs Wage')
plt.xlabel('Age')
plt.ylabel('Wage (WAGP)')
plt.grid(True)
plt.tight_layout()
plt.show()

```

```
rmProjects/pythonProject6/.venv/bin/python /Users/saniyaisaac/PycharmProjects/pythonProject6/pl

it code 0
```

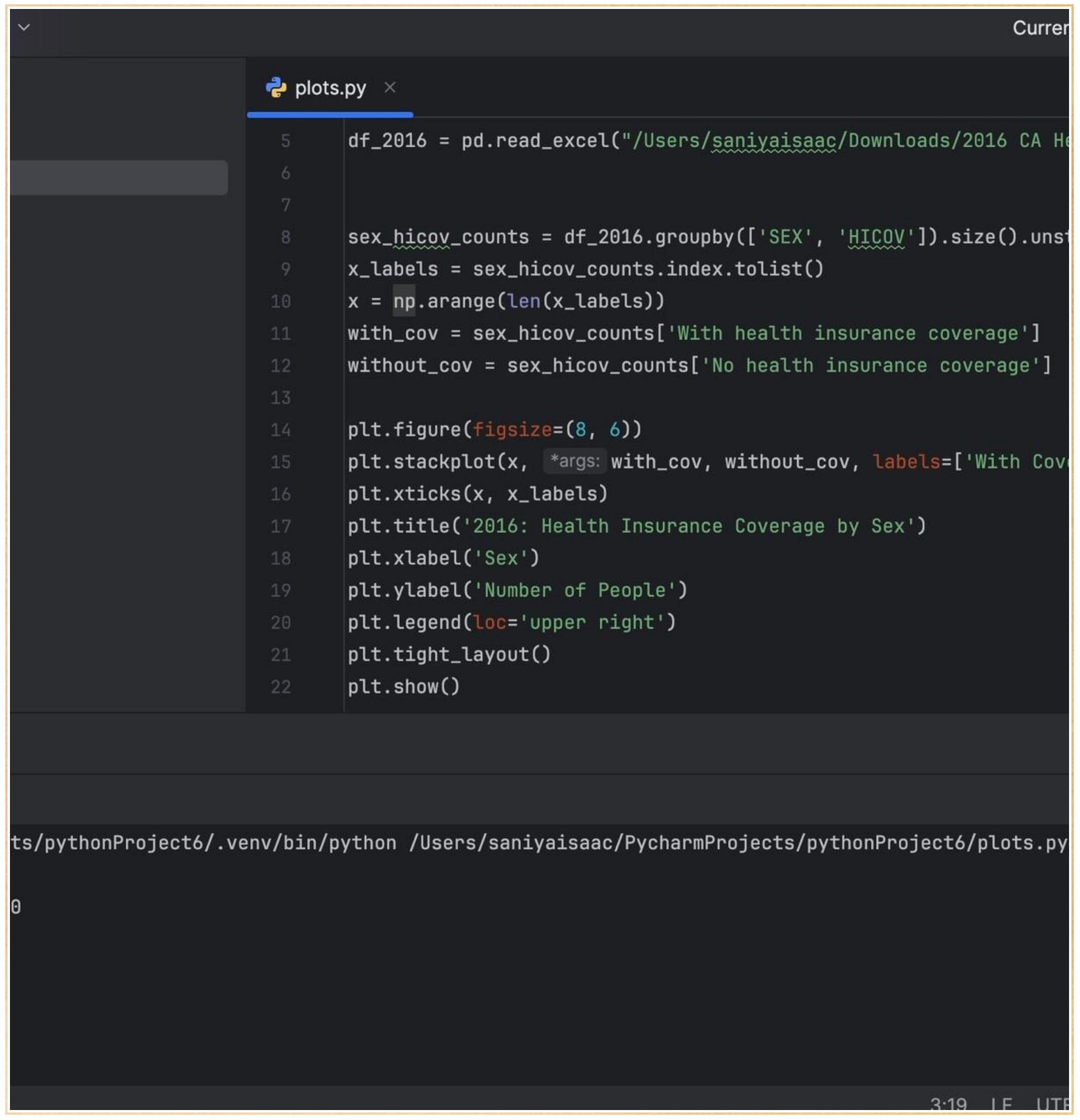


Current File
Plots
1:1
800x600 PNG (32-bit color) 26.39 kB
2016: Health Insurance Coverage by Sex
With Coverage
Without Coverage
Number of People
10000
8000
6000
4000
2000
0
Female
Male
Sex
0
1
2
3
4

```
impor
impor

df_20

sex_h
x_lab
x = n
with_
witho

plt.f
plt.s
plt.x
plt.t
plt.x
```

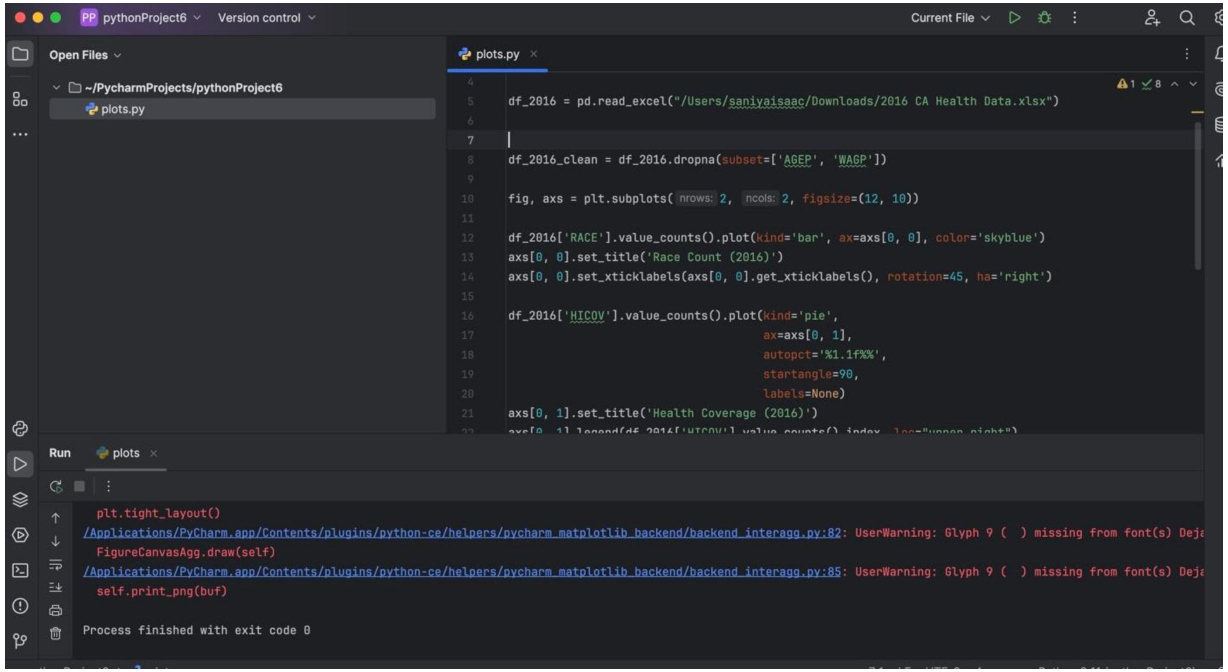
```
thon /Users/saniyaisaac/PycharmProjects/pythonProject6/plots.py
```

3:19 LF UTF-8 4 spaces Python 3.11 (pythonP



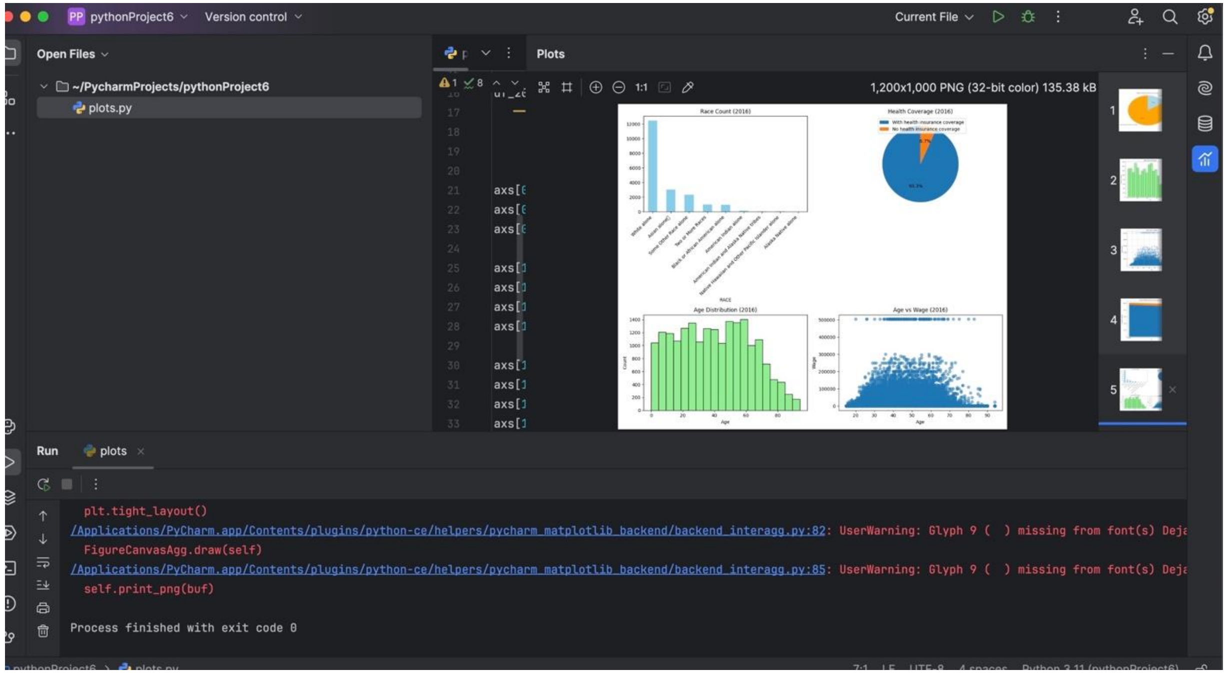
```
df_2016 = pd.read_excel("/Users/saniyaisaac/Downloads/2016 CA He


sex_hicov_counts = df_2016.groupby(['SEX', 'HICOV']).size().unst
x_labels = sex_hicov_counts.index.tolist()
x = np.arange(len(x_labels))
with_cov = sex_hicov_counts['With health insurance coverage']
without_cov = sex_hicov_counts['No health insurance coverage']

plt.figure(figsize=(8, 6))
plt.stackplot(x, with_cov, without_cov, labels=['With Cove
plt.xticks(x, x_labels)
plt.title('2016: Health Insurance Coverage by Sex')
plt.xlabel('Sex')
plt.ylabel('Number of People')
plt.legend(loc='upper right')
plt.tight_layout()
plt.show()
```

```
ts/pythonProject6/.venv/bin/python /Users/saniyaisaac/PycharmProjects/pythonProject6/plots.py

0
```

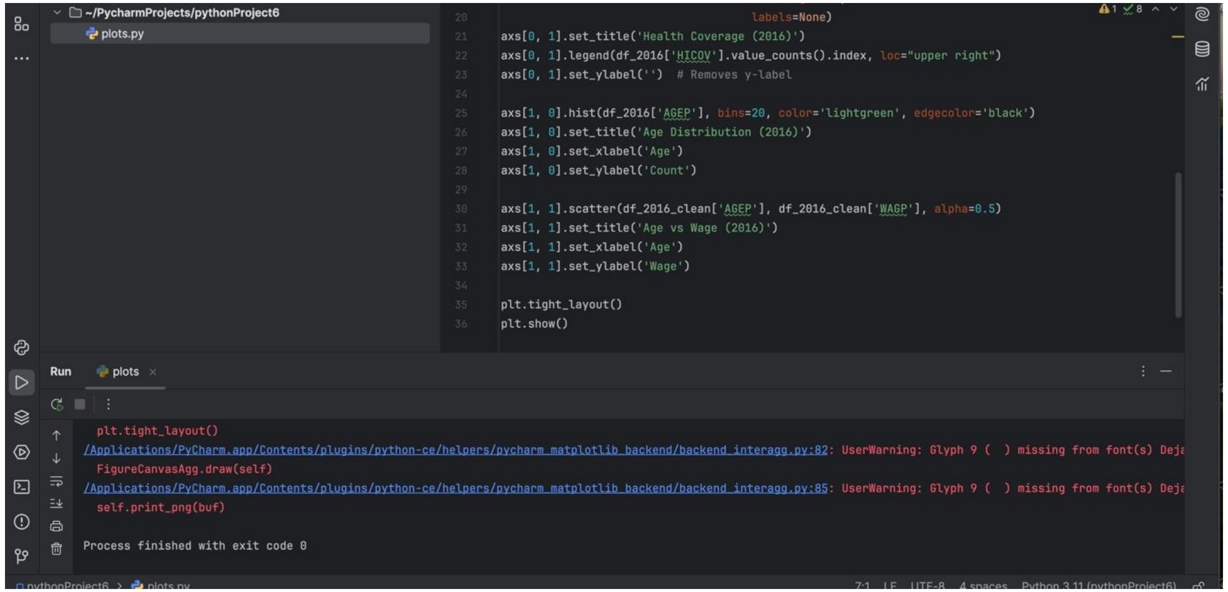


```python
df_2016 = pd.read_excel("/Users/saniyaisaac/Downloads/2016 CA Health Data.xlsx")


df_2016_clean = df_2016.dropna(subset=['AGEP', 'WAGP'])

fig, axs = plt.subplots( nrows: 2,  ncols: 2, figsize=(12, 10))

df_2016['RACE'].value_counts().plot(kind='bar', ax=axs[0, 0], color='skyblue')
axs[0, 0].set_title('Race Count (2016)')
axs[0, 0].set_xticklabels(axs[0, 0].get_xticklabels(), rotation=45, ha='right')

df_2016['HICOV'].value_counts().plot(kind='pie',
                                     ax=axs[0, 1],
                                     autopct='%1.1f%%',
                                     startangle=90,
                                     labels=None)
axs[0, 1].set_title('Health Coverage (2016)')
axs[0, 1].legend(df_2016['HICOV'].value_counts().index, loc="upper right")
```

```
  plt.tight_layout()
/Applications/PyCharm.app/Contents/plugins/python-ce/helpers/pycharm_matplotlib_backend/backend_interagg.py:82: UserWarning: Glyph 9 (  ) missing from font(s) Deja
  FigureCanvasAgg.draw(self)
/Applications/PyCharm.app/Contents/plugins/python-ce/helpers/pycharm_matplotlib_backend/backend_interagg.py:85: UserWarning: Glyph 9 (  ) missing from font(s) Deja
  self.print_png(buf)

Process finished with exit code 0
```

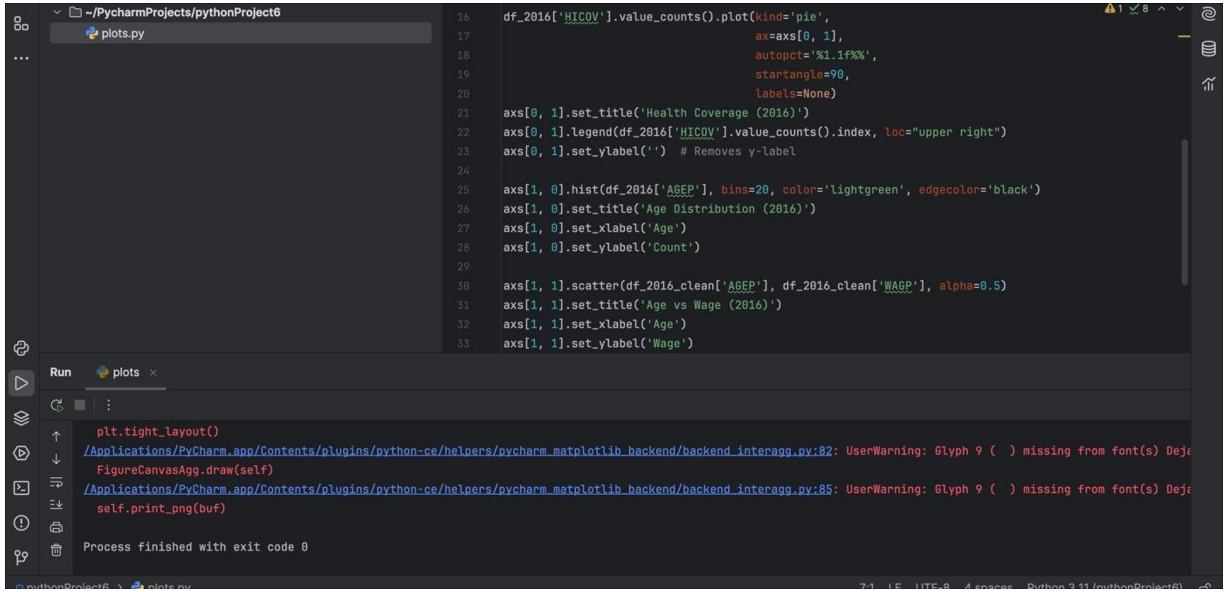


```
  plt.tight_layout()
/Applications/PyCharm.app/Contents/plugins/python-ce/helpers/pycharm_matplotlib_backend/backend_interagg.py:82: UserWarning: Glyph 9 (  ) missing from font(s) Deja
  FigureCanvasAgg.draw(self)
/Applications/PyCharm.app/Contents/plugins/python-ce/helpers/pycharm_matplotlib_backend/backend_interagg.py:85: UserWarning: Glyph 9 (  ) missing from font(s) Deja
  self.print_png(buf)

Process finished with exit code 0
```



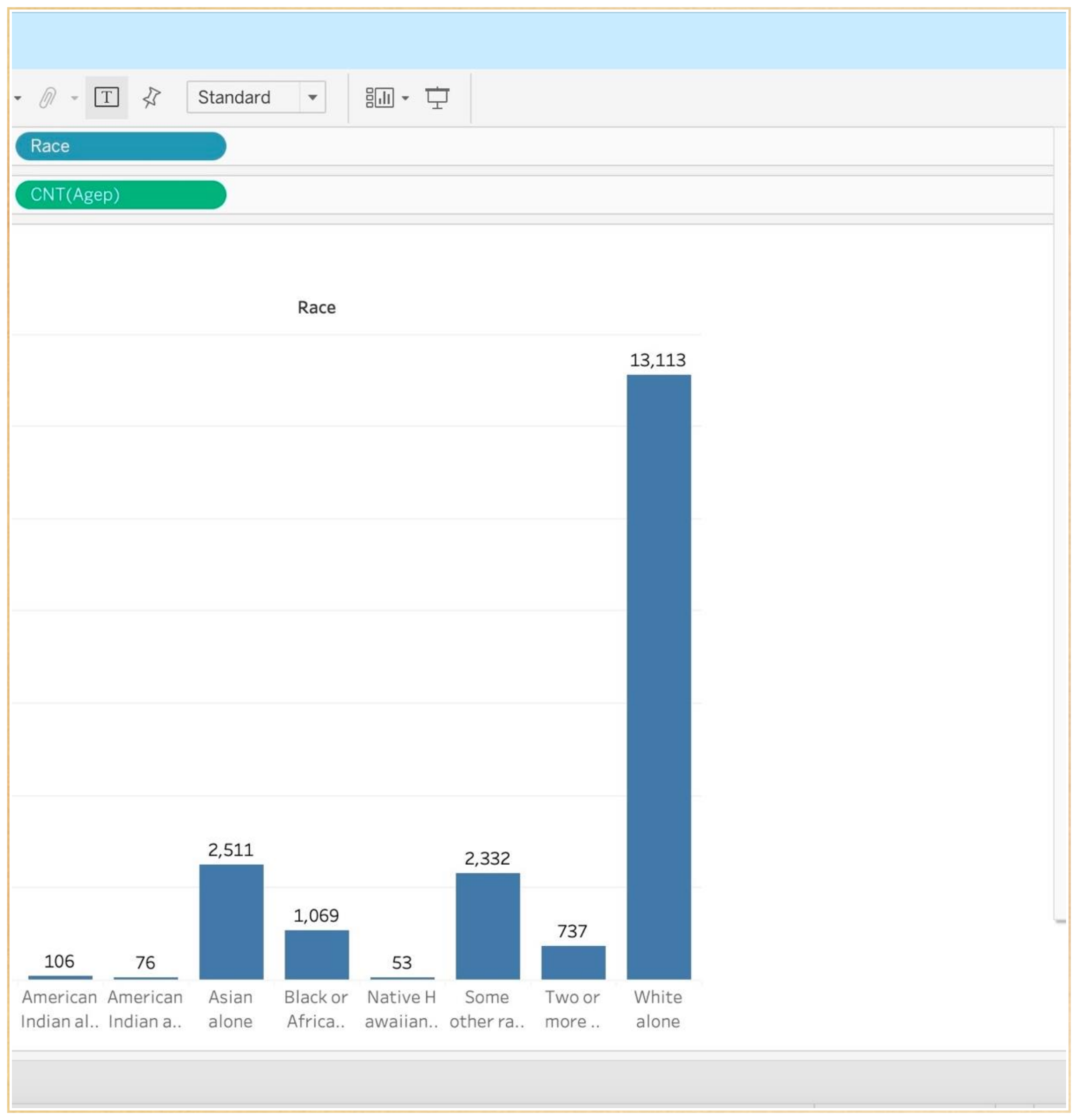
```python
                                     labels=None)
axs[0, 1].set_title('Health Coverage (2016)')
axs[0, 1].legend(df_2016['HICOV'].value_counts().index, loc="upper right")
axs[0, 1].set_ylabel('')  # Removes y-label

axs[1, 0].hist(df_2016['AGEP'], bins=20, color='lightgreen', edgecolor='black')
axs[1, 0].set_title('Age Distribution (2016)')
axs[1, 0].set_xlabel('Age')
axs[1, 0].set_ylabel('Count')

axs[1, 1].scatter(df_2016_clean['AGEP'], df_2016_clean['WAGP'], alpha=0.5)
axs[1, 1].set_title('Age vs Wage (2016)')
axs[1, 1].set_xlabel('Age')
axs[1, 1].set_ylabel('Wage')

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

```
  plt.tight_layout()
/Applications/PyCharm.app/Contents/plugins/python-ce/helpers/pycharm_matplotlib_backend/backend_interagg.py:82: UserWarning: Glyph 9 (  ) missing from font(s) Deja
  FigureCanvasAgg.draw(self)
/Applications/PyCharm.app/Contents/plugins/python-ce/helpers/pycharm_matplotlib_backend/backend_interagg.py:85: UserWarning: Glyph 9 (  ) missing from font(s) Deja
  self.print_png(buf)

Process finished with exit code 0
```



```python
df_2016['HICOV'].value_counts().plot(kind='pie',
                                     ax=axs[0, 1],
                                     autopct='%1.1f%%',
                                     startangle=90,
                                     labels=None)
axs[0, 1].set_title('Health Coverage (2016)')
axs[0, 1].legend(df_2016['HICOV'].value_counts().index, loc="upper right")
axs[0, 1].set_ylabel('')  # Removes y-label

axs[1, 0].hist(df_2016['AGEP'], bins=20, color='lightgreen', edgecolor='black')
axs[1, 0].set_title('Age Distribution (2016)')
axs[1, 0].set_xlabel('Age')
axs[1, 0].set_ylabel('Count')

axs[1, 1].scatter(df_2016_clean['AGEP'], df_2016_clean['WAGP'], alpha=0.5)
axs[1, 1].set_title('Age vs Wage (2016)')
axs[1, 1].set_xlabel('Age')
axs[1, 1].set_ylabel('Wage')
```

```
  plt.tight_layout()
/Applications/PyCharm.app/Contents/plugins/python-ce/helpers/pycharm_matplotlib_backend/backend_interagg.py:82: UserWarning: Glyph 9 (  ) missing from font(s) Deja
  FigureCanvasAgg.draw(self)
/Applications/PyCharm.app/Contents/plugins/python-ce/helpers/pycharm_matplotlib_backend/backend_interagg.py:85: UserWarning: Glyph 9 (  ) missing from font(s) Deja
  self.print_png(buf)

Process finished with exit code 0
```



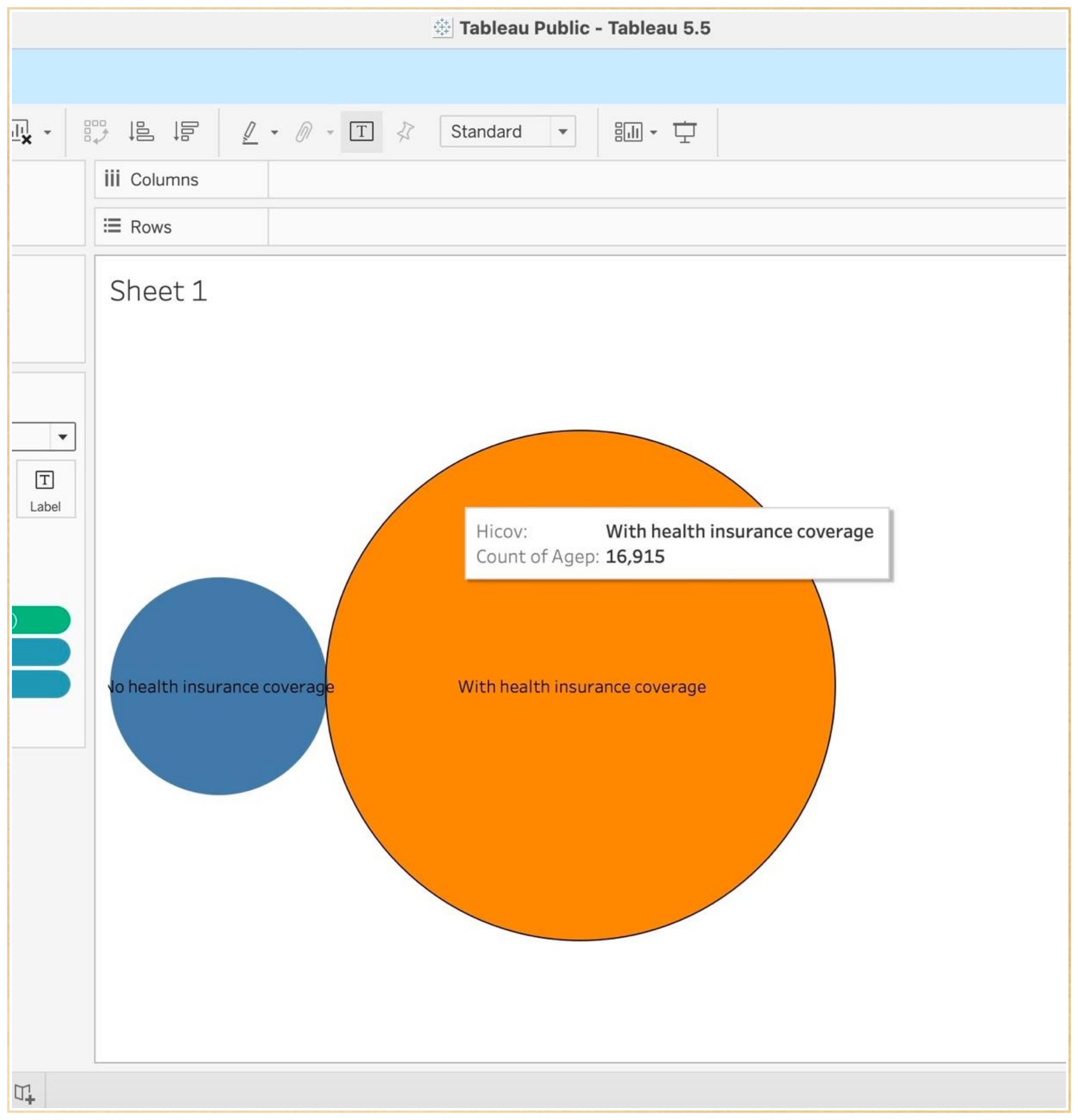


Tableau Public - Tableau 5.5

Sheet 1

Hicov: With health insurance coverage
Count of Agep: 16,915

Conclusion

- In both CTEC 128 and CTEC 298, we learned how to work with real data to explore important issues. In CTEC 128, one project looked at possible racial bias in traffic stops in Maryland , while the other focused on how the Affordable Care Act improved healthcare access in California. We used raw data, cleaned it, asked questions, and created charts to show our results. In CTEC 298, we expanded on those skills by learning Python, Pandas, Matplotlib, Tableau, and Jupyter Notebook. We completed tutorials and created six different plots based on our CTEC 128 data. These projects helped us see how math, coding, and analysis all work together to better understand real-world problems. In the end, we learned how data can be a powerful tool to uncover truth, support decision-making, and drive positive change in society.

References

- U.S. Census Bureau. (2008 & 2016). American Community Survey (ACS) Data. Retrieved from https://www.census.gov/programs-surveys/acs
- Healthcare.gov. (n.d.). The Affordable Care Act. Retrieved from https://www.healthcare.gov/where-can-i-read-the-affordable-care-act/
- KFF. (2024). Health Policy 101: The Politics of Health Care and the 2024 Election. Retrieved from https://www.kff.org
- Maryland Governor's Office of Crime Prevention, Youth, and Victim Services. Traffic Stop Data Dashboard. https://gocpp.maryland.gov/data-dashboards/traffic-stop-data-dashboard/
- Stanford Open Policing Project. Traffic Stop Data. https://openpolicing.stanford.edu/data/
- Python.org. (n.d.). Python Documentation. Retrieved from https://docs.python.org
- Matplotlib. (n.d.). Matplotlib Documentation. Retrieved from https://matplotlib.org/stable/index.html
- Pandas Documentation. (n.d.). Retrieved from https://pandas.pydata.org
- Tableau. (n.d.). Official Tableau Help. Retrieved from https://help.tableau.com
- GitHub. (n.d.). Project Repositories. Retrieved from https://github.com