Settings Help

Launcher | Optional_Peer_Graded_Assig

Code | Python (Pyodide)

Add your code below following the instructions given in the course to complete the peer graded assignment

[3]:

```
# Install necessary packages
import piplite
await piplite.install(['numpy', 'pandas', 'seaborn'])

# Import libraries
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from js import fetch
import io

# Fetch the Boston housing dataset
URL = 'https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-ST0151EN-SkillsNet
resp = await fetch(URL)
boston_url = io.BytesIO((await resp.arrayBuffer()).to_py())

# Load dataset into a DataFrame
boston_df = pd.read_csv(boston_url)

# Create a boxplot for the MEDV (Median value of owner-occupied homes)
plt.figure(figsize=(8, 6))
sns.boxplot(y=boston_df['MEDV'], color='skyblue')
plt.title('Boxplot of Median Value of Owner-Occupied Homes (MEDV)')
plt.ylabel('MEDV ($1000s)')
plt.grid(True)
```

dide) | Idle

Mode: Edit | Ln 1, Col 1 | Optional_Peer_Graded_Assignment.jupyterlite.ipynb

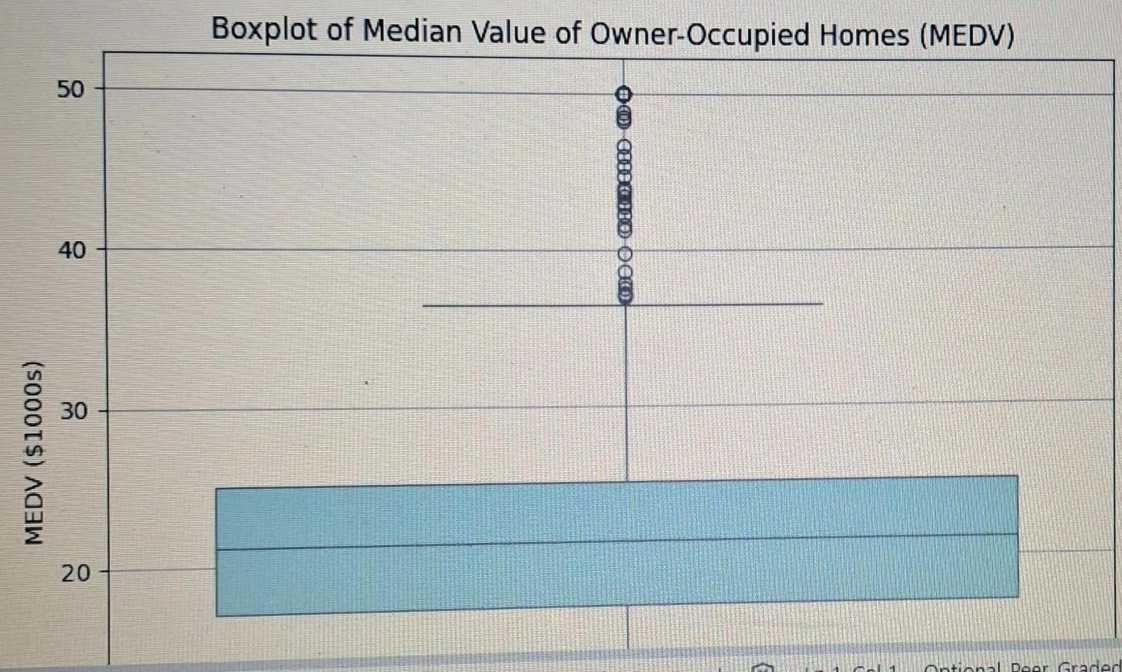
Settings Help

Launcher | Optional_Peer_Graded_Assig

Code

Python (Pyodide)

```
plt.figure(figsize=(8, 6))
sns.boxplot(y=boston_df['MEDV'], color='skyblue')
plt.title('Boxplot of Median Value of Owner-Occupied Homes (MEDV)')
plt.ylabel('MEDV ($1000s)')
plt.grid(True)
plt.show()
```



Mode: Command Ln 1, Col 1 Optional_Peer_Graded_Assignment.jupyterlite.ipynb 3

Graded Assignment | Cour

Skills Network Labs

ls/jupyterlite?ulid=ulid-d16e51dafa13eb88c14656f2910b259abb9b48ed

Verify it's you

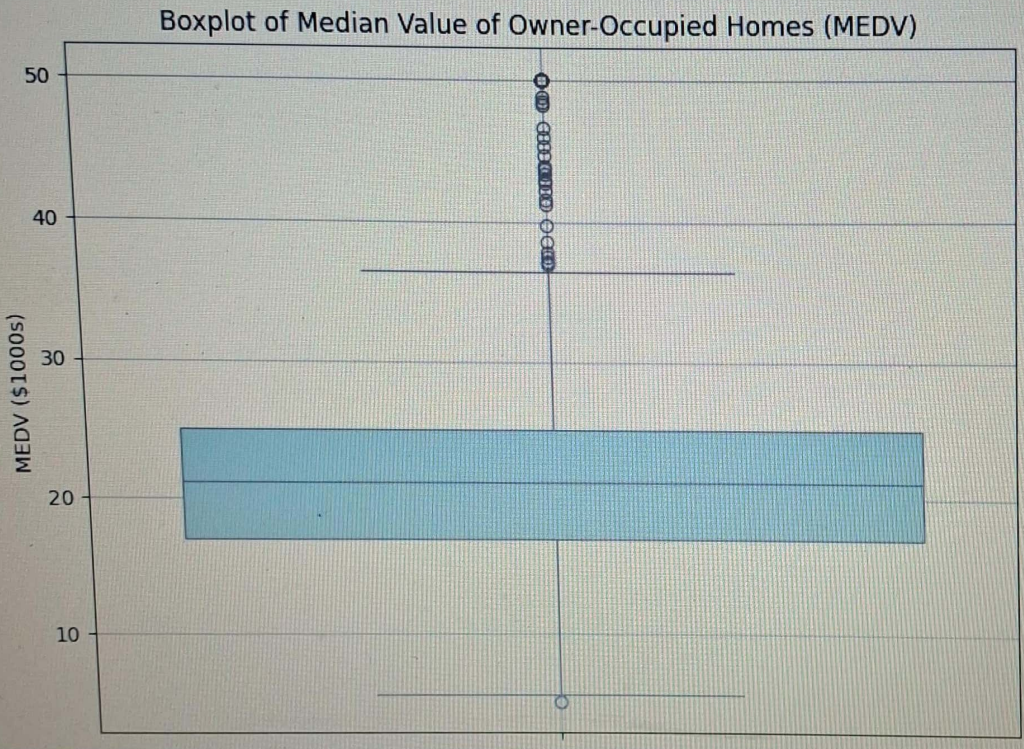
Tabs Settings Help

Launcher

Optional_Peer_Graded_Assig

Code

Python (Pyodide)



Python (Pyodide) | Idle

Mode: Command

Ln 1, Col 1

Optional_Peer_Graded_Assignment.jupyterlite.ipynb

3

ed Assignment | Cour × Skills Network Labs ×

yterlite?ulid=ulid-d16e51dafa13eb88c14656f2910b259abb9b48ed

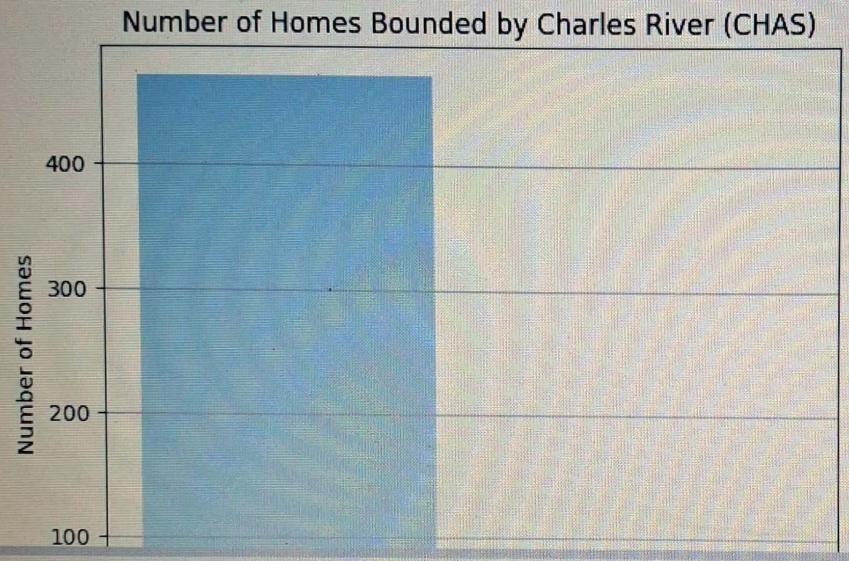
Verify it's you

Settings Help

Launcher × Optional_Peer_Graded_Assig ×

Code Python (Pyodide)

[5]:
```
# Barplot of CHAS (Charles River Dummy Variable)
plt.figure(figsize=(6, 5))
sns.countplot(x='CHAS', hue='CHAS', data=boston_df, palette='pastel', legend=False)
plt.title('Number of Homes Bounded by Charles River (CHAS)')
plt.xlabel('CHAS (1 = bounds river, 0 = does not)')
plt.ylabel('Number of Homes')
plt.grid(True, axis='y')
plt.show()
```



odide) | Idle Mode: Command Ln 1, Col 1 Optional_Peer_Graded_Assignment.jupyterlite.ipynb 3

Skills Network Labs... Rainy days ahead ENG 11:13

ed Assignment | Cour | Skills Network Labs

yterlite?ulid=ulid-d16e51dafa13eb88c14656f2910b259abb9b48ed

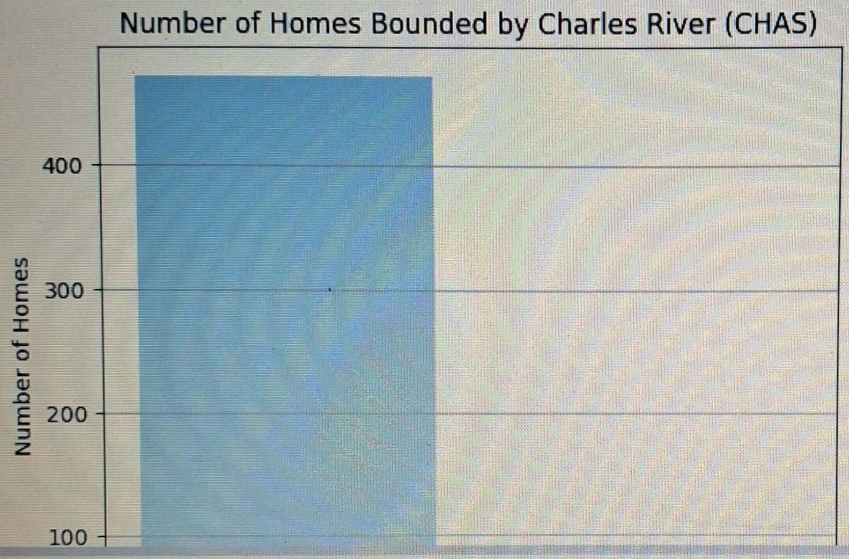
Verify it's you

Settings Help

Launcher | Optional_Peer_Graded_Assig

Code | Python (Pyodide)

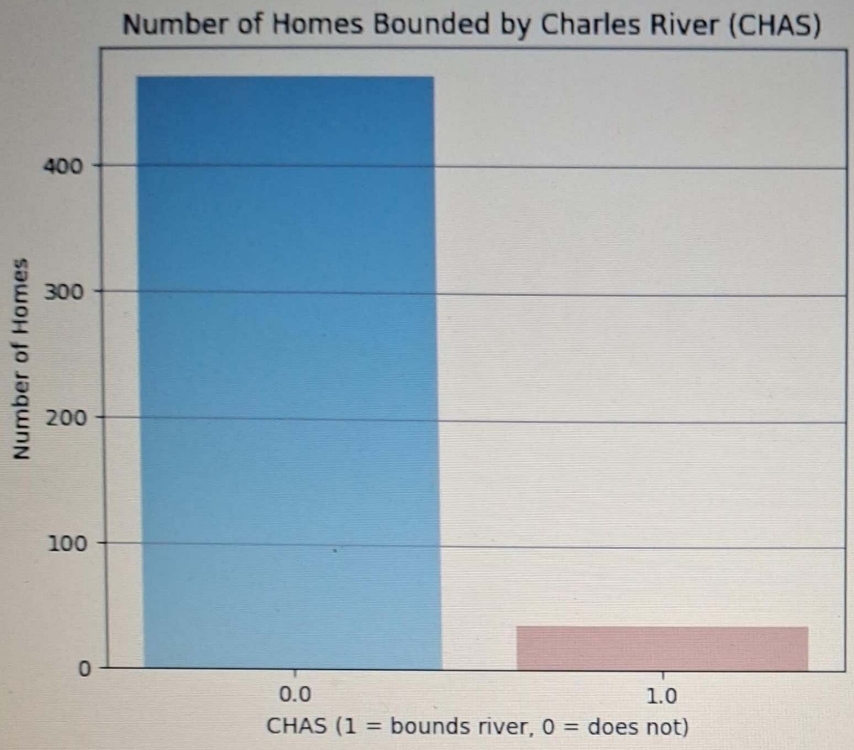
[5]:
```
# Barplot of CHAS (Charles River Dummy Variable)
plt.figure(figsize=(6, 5))
sns.countplot(x='CHAS', hue='CHAS', data=boston_df, palette='pastel', legend=False)
plt.title('Number of Homes Bounded by Charles River (CHAS)')
plt.xlabel('CHAS (1 = bounds river, 0 = does not)')
plt.ylabel('Number of Homes')
plt.grid(True, axis='y')
plt.show()
```



odide) | Idle | Mode: Command | Ln 1, Col 1 | Optional_Peer_Graded_Assignment.jupyterlite.ipynb | 3

Skills Network Labs... | Rainy days ahead | ENG | 11:13 | 14-06-2025

Launcher | Optional_Peer_Graded_Assig | Code | Python (Pyodide)

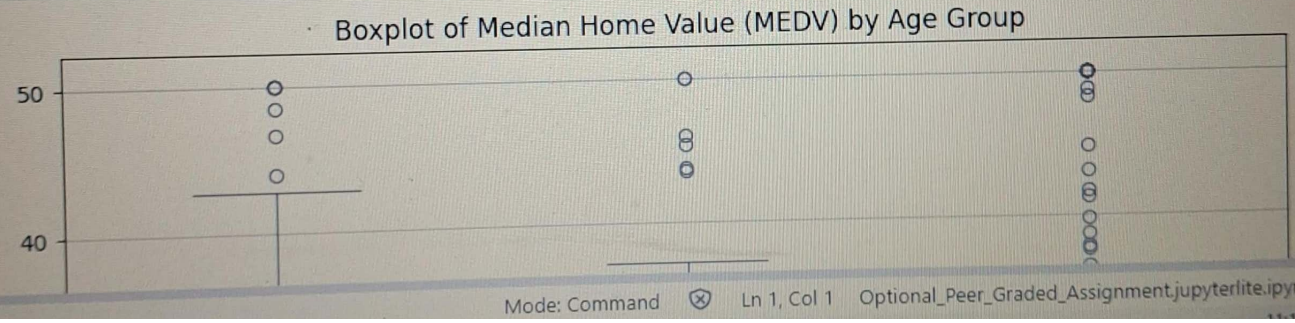



```
[7]: import pandas as pd
```

Mode: Command | Ln 1, Col 1 | Optional_Peer_Graded_Assignment.ipynb

Launcher | Optional_Peer_Graded_Assig | Code | Python (Pyodide)

[7]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Discretize the AGE variable
bins = [0, 35, 70, 100]
labels = ['35 years and younger', 'Between 35 and 70 years', '70 years and older']
boston_df['AGE_group'] = pd.cut(boston_df['AGE'], bins=bins, labels=labels, right=True)

# Create boxplot of MEDV vs AGE group
plt.figure(figsize=(10, 6))
sns.boxplot(x='AGE_group', y='MEDV', hue='AGE_group', data=boston_df, palette='Set2', legend=False)
plt.title('Boxplot of Median Home Value (MEDV) by Age Group')
plt.xlabel('Age Group of Houses')
plt.ylabel('MEDV ($1000s)')
plt.grid(True, axis='y')
plt.show()
```

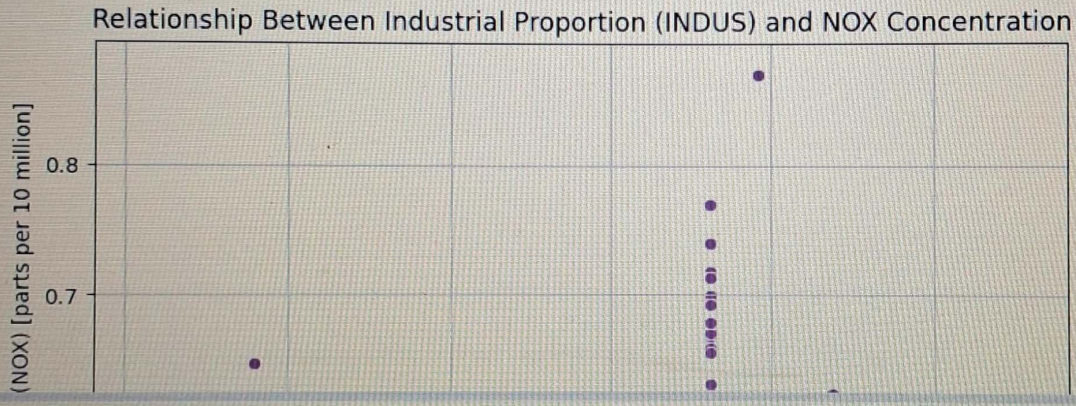


Mode: Command | Ln 1, Col 1 | Optional_Peer_Graded_Assignment.jupyterlite.ipynb | 3


Boxplot of Median Home Value (MEDV) by Age Group
MEDV ($1000s)
50
40
30
20
10
35 years and younger
Between 35 and 70 years
70 years and older

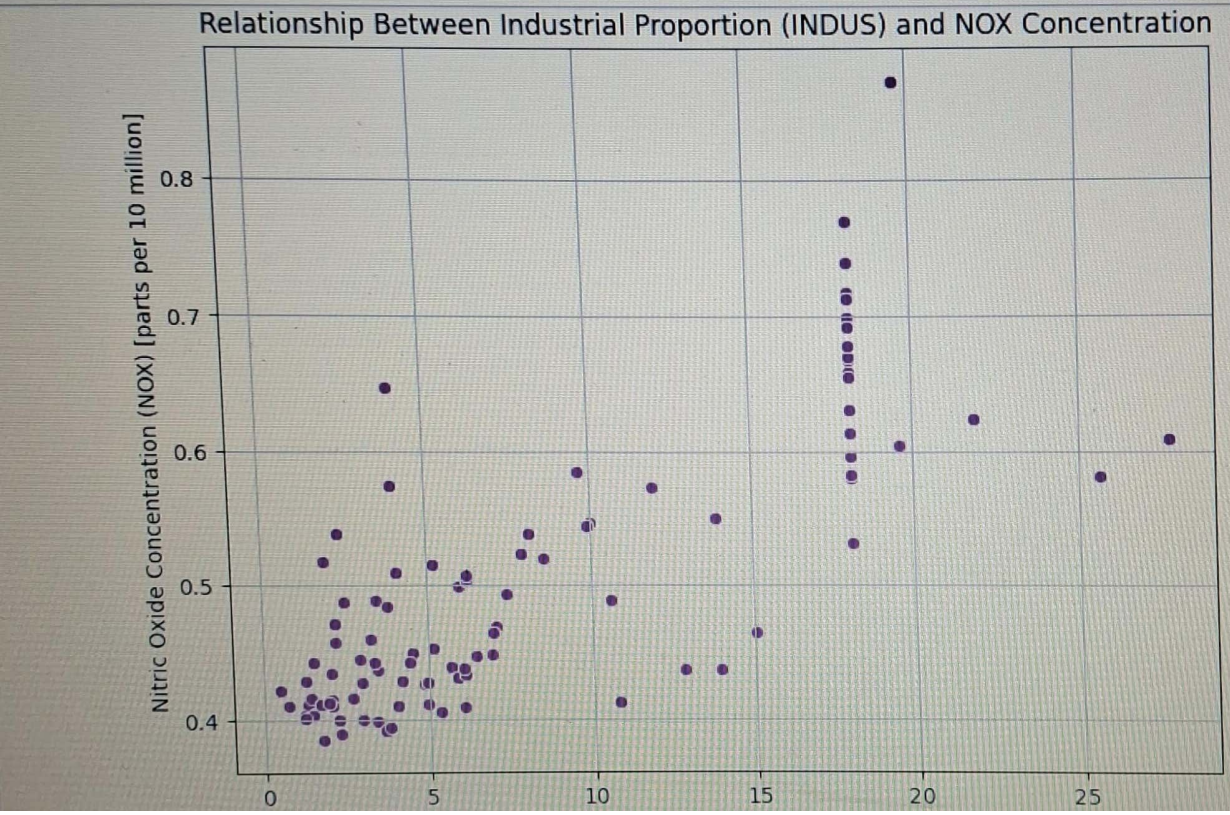
Launcher | Optional_Peer_Graded_Assig | Code | Python (Pyodide)

[8]:
```
import matplotlib.pyplot as plt
import seaborn as sns

# Create scatter plot
plt.figure(figsize=(8, 6))
sns.scatterplot(x='INDUS', y='NOX', data=boston_df, color='purple', edgecolor='white')

# Add labels and title
plt.title('Relationship Between Industrial Proportion (INDUS) and NOX Concentration')
plt.xlabel('Proportion of Non-Retail Business Acres per Town (INDUS)')
plt.ylabel('Nitric Oxide Concentration (NOX) [parts per 10 million]')
plt.grid(True)
plt.show()
```

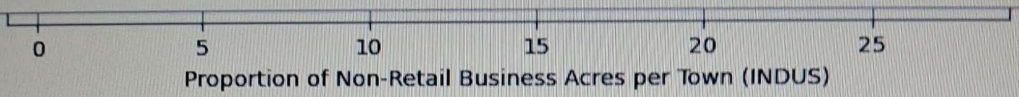


odide) | Idle Mode: Command Ln 1, Col 1 Optional_Peer_Graded_Assignment.jupyterlite.ipynb 3

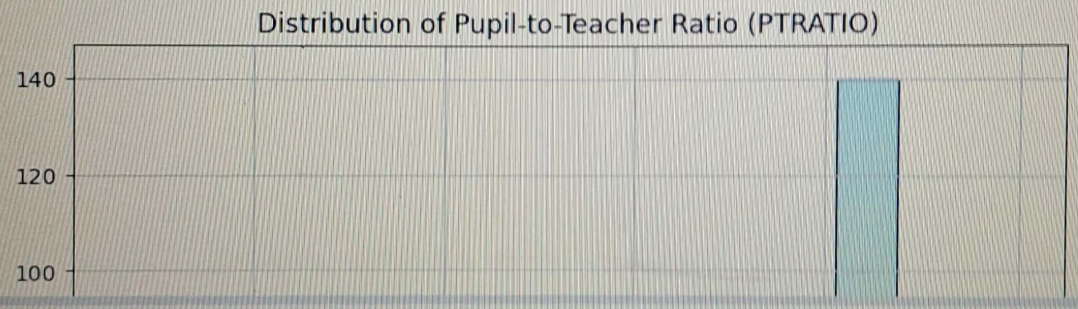
Launcher | Optional_Peer_Graded_Assig | Code | Python (Pyodide)



Launcher | Optional_Peer_Graded_Assig | Code | Python (Pyodide)

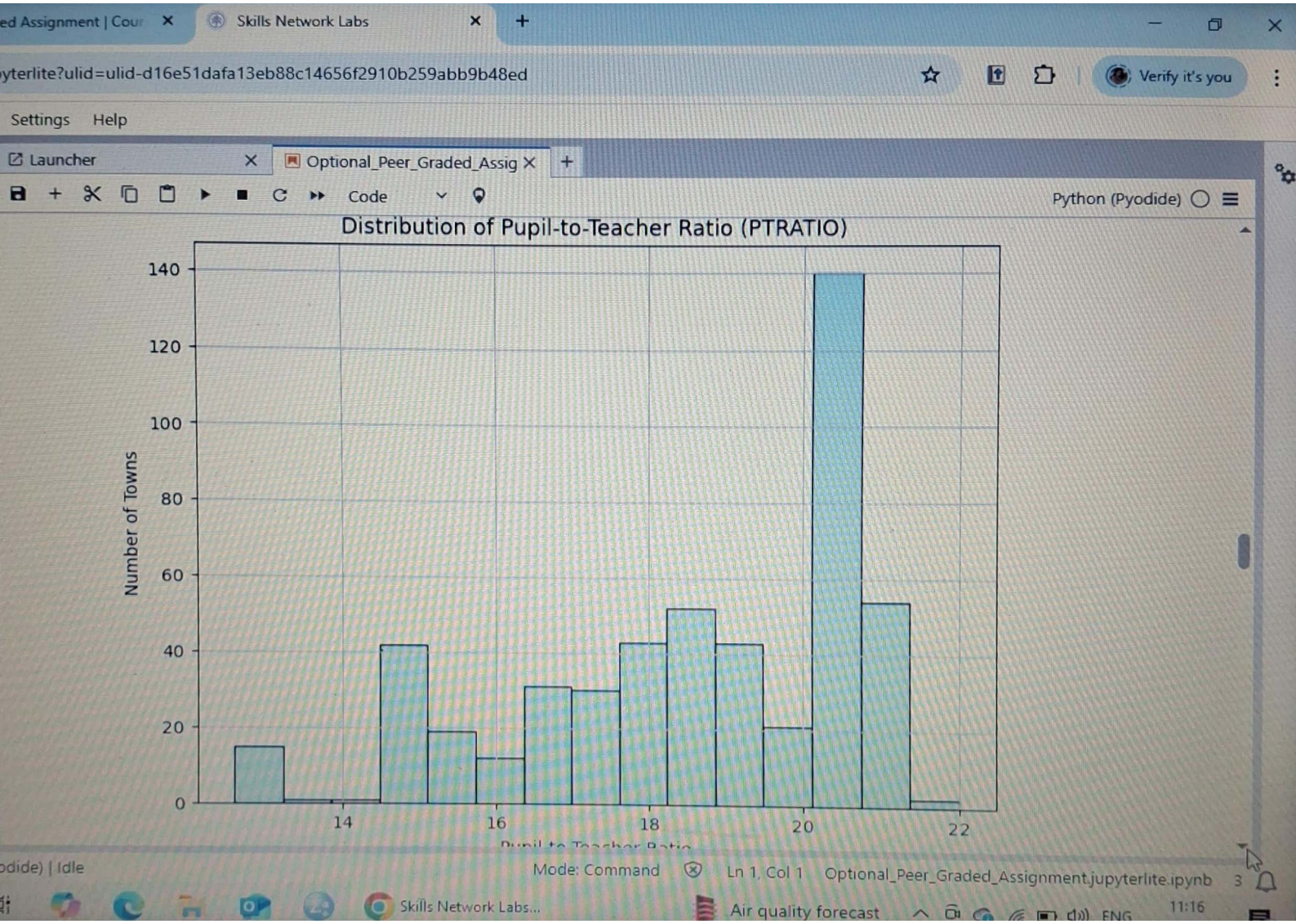


[9]:
```
import matplotlib.pyplot as plt
import seaborn as sns

# Create histogram for PTRATIO
plt.figure(figsize=(8, 6))
sns.histplot(boston_df['PTRATIO'], bins=15, kde=False, color='skyblue', edgecolor='black')

# Add labels and title
plt.title('Distribution of Pupil-to-Teacher Ratio (PTRATIO)')
plt.xlabel('Pupil-to-Teacher Ratio')
plt.ylabel('Number of Towns')
plt.grid(True)
plt.show()
```



lide) | Idle Mode: Command Ln 1, Col 1 Optional_Peer_Graded_Assignment.jupyterlite.ipynb 3

Skills Network Labs... Air quality forecast ENG 11:16 14-06-2025


Distribution of Pupil-to-Teacher Ratio (PTRATIO)
Number of Towns
140
120
100
80
60
40
20
0
14
16
18
20
22

Launcher | Optional_Peer_Graded_Assig | Code | Python (Pyodide)

Pupil-to-Teacher Ratio

[10]:
```
from scipy.stats import ttest_ind

# Separate MEDV values by CHAS groups
medv_river = boston_df[boston_df['CHAS'] == 1]['MEDV']
medv_no_river = boston_df[boston_df['CHAS'] == 0]['MEDV']

# Perform independent t-test (equal_var=False is safer when sample sizes or variances differ)
t_stat, p_value = ttest_ind(medv_river, medv_no_river, equal_var=False)

# Print results
print(f"T-statistic: {t_stat:.4f}")
print(f"P-value: {p_value:.4f}")

# Conclusion at a = 0.05
alpha = 0.05
if p_value < alpha:
    print("Conclusion: Reject the null hypothesis. There is a significant difference in median house values between ho
else:
    print("Conclusion: Fail to reject the null hypothesis. There is no significant difference in median house values b
```

```
T-statistic: 3.1133
P-value: 0.0036
Conclusion: Reject the null hypothesis. There is a significant difference in median house values between homes near t
he river and those not near the river.
```

[12]:
```
import pandas as pd
import statsmodels.api as sm
```

odide) | Idle | Mode: Command | Ln 1, Col 1 | Optional_Peer_Graded_Assignment.jupyterlite.ipynb | 3

Skills Network Labs... | Air quality forecast | ENG | 11:16

Launcher | Optional_Peer_Graded_Assig

Code | Python (Pyodide)

[12]:
```
import pandas as pd
import statsmodels.api as sm
from statsmodels.formula.api import ols

# Discretize AGE into 3 groups
bins = [0, 35, 70, 100]
labels = ['35 years and younger', 'Between 35 and 70 years', '70 years and older']
boston_df['AGE_group'] = pd.cut(boston_df['AGE'], bins=bins, labels=labels)

# One-way ANOVA
model = ols('MEDV ~ C(AGE_group)', data=boston_df).fit()
anova_table = sm.stats.anova_lm(model, typ=2)

# Display ANOVA table
print(anova_table)

# Interpret result
alpha = 0.05
p_value = anova_table['PR(>F)'].iloc[0]

print(f"\nP-value: {p_value:.4f}")
if p_value < alpha:
    print("Conclusion: Reject the null hypothesis. There is a significant difference in median home values among AGE g
else:
    print("Conclusion: Fail to reject the null hypothesis. No significant difference in median home values among AGE g
```

```
                  sum_sq     df          F        PR(>F)
C(AGE group)  5401.731883    2.0  36.40765  1.710501e-15
```

Mode: Command | Ln 1, Col 1 | Optional_Peer_Graded_Assignment.jupyterlite.ipynb

Launcher | Optional_Peer_Graded_Assig | +

Code | Python (Pyodide)

```
                    sum_sq     df         F        PR(>F)
C(AGE_group)   5401.731883    2.0  36.40765  1.710501e-15
Residual      37314.563532  503.0       NaN           NaN

P-value: 0.0000
Conclusion: Reject the null hypothesis. There is a significant difference in median home values among AGE groups.
```

[13]:
```
from scipy.stats import pearsonr

# Pearson correlation test
corr_coeff, p_value = pearsonr(boston_df['NOX'], boston_df['INDUS'])

# Print results
print(f"Pearson Correlation Coefficient: {corr_coeff:.4f}")
print(f"P-value: {p_value:.4f}")

# Interpret the result
alpha = 0.05
if p_value < alpha:
    print("Conclusion: Reject the null hypothesis. There is a significant linear relationship between NOX and INDUS.")
else:
    print("Conclusion: Fail to reject the null hypothesis. There is no significant linear relationship between NOX and
```

```
Pearson Correlation Coefficient: 0.7637
P-value: 0.0000
Conclusion: Reject the null hypothesis. There is a significant linear relationship between NOX and INDUS.
```

[14]:
```
import statsmodels.api as sm
```

(Pyodide) | Idle Mode: Command Ln 1, Col 1 Optional_Peer_Graded_Assignment.jupyterlite.ipynb 3

Launcher | Optional_Peer_Graded_Assig | Code | Python (Pyodide)

Conclusion: Reject the null hypothesis. There is a significant linear relationship between NOX and INDUS.

[14]:
```
import statsmodels.api as sm

# Define dependent and independent variables
X = boston_df['DIS']          # Independent variable
y = boston_df['MEDV']         # Dependent variable

# Add a constant term to the predictor (for intercept)
X = sm.add_constant(X)

# Fit the regression model
model = sm.OLS(y, X).fit()

# Display summary of regression results
print(model.summary())

# Extract p-value for DIS
p_value = model.pvalues['DIS']
alpha = 0.05

print(f"\nP-value for DIS: {p_value:.4f}")
if p_value < alpha:
    print("Conclusion: Reject the null hypothesis. The distance to employment centers has a significant effect on the
else:
    print("Conclusion: Fail to reject the null hypothesis. The distance to employment centers does not have a signific
```

odide) | Idle | Mode: Command | Ln 1, Col 1 | Optional_Peer_Graded_Assignment.jupyterlite.ipynb | 3

Skills Network Labs... | 34°C Haze | ENG | 11:18

Launcher | Optional_Peer_Graded_Assig | Code | Python (Pyodide)

OLS Regression Results

Dep. Variable:	MEDV	R-squared:	0.062
Model:	OLS	Adj. R-squared:	0.061
Method:	Least Squares	F-statistic:	33.58
Date:	Sat, 14 Jun 2025	Prob (F-statistic):	1.21e-08
Time:	10:01:29	Log-Likelihood:	-1823.9
No. Observations:	506	AIC:	3652.
Df Residuals:	504	BIC:	3660.
Df Model:	1		
Covariance Type:	nonrobust		

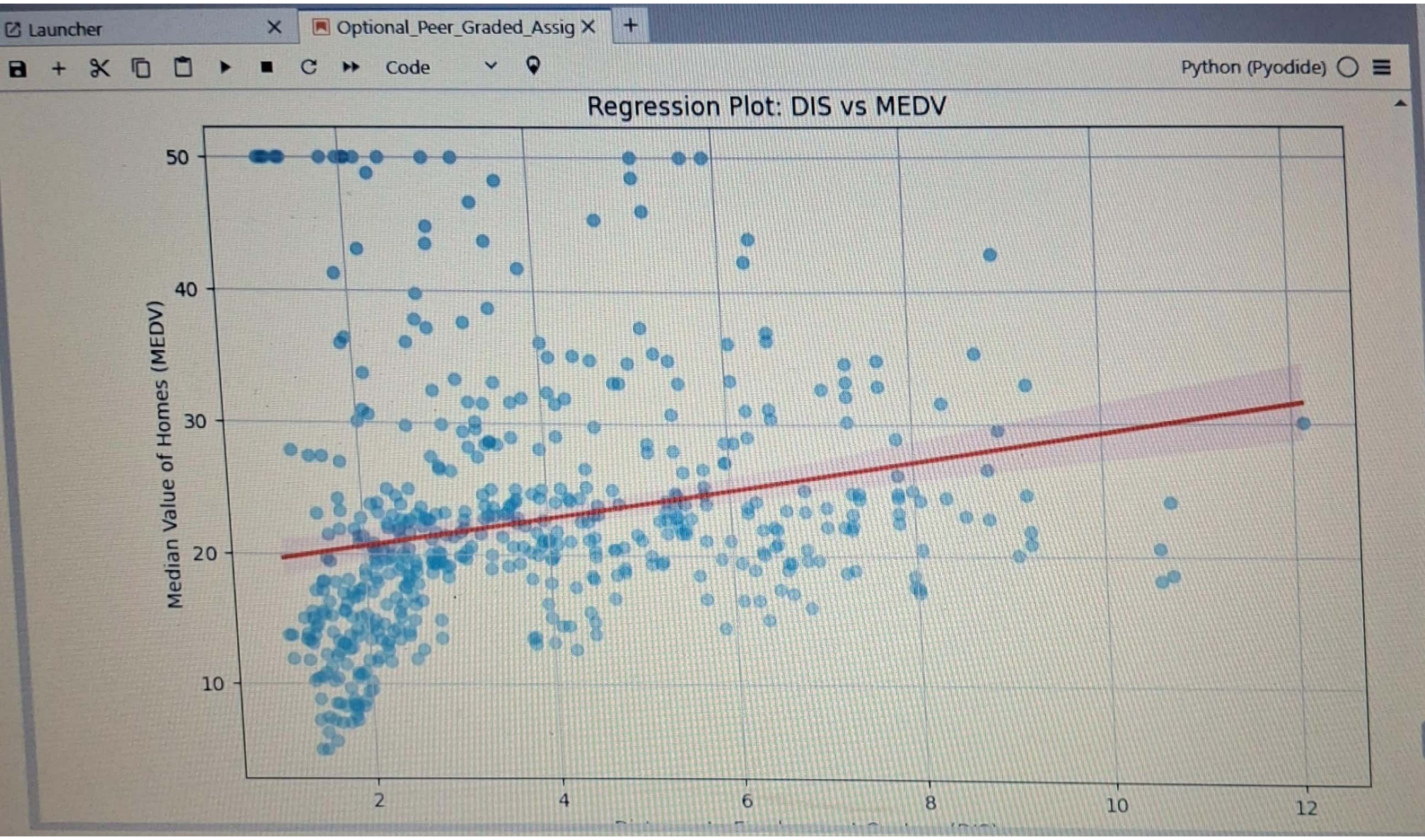
	coef	std err	t	P>|t|	[0.025	0.975]
const	18.3901	0.817	22.499	0.000	16.784	19.996
DIS	1.0916	0.188	5.795	0.000	0.722	1.462

Omnibus:	139.779	Durbin-Watson:	0.570
Prob(Omnibus):	0.000	Jarque-Bera (JB):	305.104
Skew:	1.466	Prob(JB):	5.59e-67
Kurtosis:	5.424	Cond. No.	9.32

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

P-value for DIS: 0.0000
Conclusion: Reject the null hypothesis. The distance to employment centers has a significant effect on the median home value.

odide) | Idle | Mode: Command | Ln 1, Col 1 | Optional_Peer_Graded_Assignment.jupyterlite.ipynb | 3

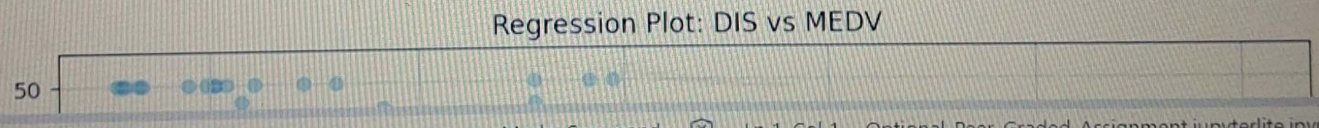

Launcher
Optional_Peer_Graded_Assig
Code
Python (Pyodide)
Regression Plot: DIS vs MEDV
Median Value of Homes (MEDV)
50
40
30
20
10
2
4
6
8
10
12

Launcher | Optional_Peer_Graded_Assig | Code | Python (Pyodide)

```
Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

P-value for DIS: 0.0000
Conclusion: Reject the null hypothesis. The distance to employment centers has a significant effect on the median hom
e value.
```

[15]:

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

# Set plot size
plt.figure(figsize=(10, 6))

# Create regression plot
sns.regplot(x='DIS', y='MEDV', data=boston_df, line_kws={'color': 'red'}, scatter_kws={'alpha': 0.5})

# Add labels and title
plt.title('Regression Plot: DIS vs MEDV')
plt.xlabel('Distance to Employment Centres (DIS)')
plt.ylabel('Median Value of Homes (MEDV)')
plt.grid(True)

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



odide) | Idle

Mode: Command | Ln 1, Col 1 | Optional_Peer_Graded_Assignment.jupyterlite.ipynb | 3