Statistics Project for Data Science

Import the libraries needed for the project

In [1]:
```
## Import the libraries needed

import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import matplotlib.ticker as ticker
import numpy as np; np.random.seed(1)
```

Accessing the dataframe's URL and reading in the CSV file from the URL using the request library

In [2]:
```
## Access the Boston Dataframe URL
## Read in the csv file from the url using the request library

boston_url = 'https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-ST0151EN-Ski
boston_df=pd.read_csv(boston_url)
```

Previewing the first ten rows of the dataframe

In [3]:
```
## Preview the Boston Dataframe

boston_df.head(10)
```

Out[3]:

	Unnamed: 0	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	LSTAT	MEDV
0	0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	296.0	15.3	4.98	24.0
1	1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	242.0	17.8	9.14	21.6
2	2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	242.0	17.8	4.03	34.7
3	3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	222.0	18.7	2.94	33.4
4	4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	222.0	18.7	5.33	36.2
5	5	0.02985	0.0	2.18	0.0	0.458	6.430	58.7	6.0622	3.0	222.0	18.7	5.21	28.7
6	6	0.08829	12.5	7.87	0.0	0.524	6.012	66.6	5.5605	5.0	311.0	15.2	12.43	22.9
7	7	0.14455	12.5	7.87	0.0	0.524	6.172	96.1	5.9505	5.0	311.0	15.2	19.15	27.1
8	8	0.21124	12.5	7.87	0.0	0.524	5.631	100.0	6.0821	5.0	311.0	15.2	29.93	16.5
9	9	0.17004	12.5	7.87	0.0	0.524	6.004	85.9	6.5921	5.0	311.0	15.2	17.10	18.9

Finding out the data type of each variable in the table

In [4]:
```
## Get information about each variable

boston_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 506 entries, 0 to 505
Data columns (total 14 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   Unnamed: 0  506 non-null    int64  
 1   CRIM        506 non-null    float64
 2   ZN          506 non-null    float64
 3   INDUS       506 non-null    float64
 4   CHAS        506 non-null    float64
 5   NOX         506 non-null    float64
 6   RM          506 non-null    float64
 7   AGE         506 non-null    float64
 8   DIS         506 non-null    float64
 9   RAD         506 non-null    float64
 10  TAX         506 non-null    float64
 11  PTRATIO     506 non-null    float64
 12  LSTAT       506 non-null    float64
 13  MEDV        506 non-null    float64
dtypes: float64(13), int64(1)
memory usage: 55.5 KB
```

Getting the number of rows and columns

In [5]:
```
## Get the number of rows and columns - prints as (number of rows, number of columns)

boston_df.shape
```

Out[5]:
```
(506, 14)
```

We can see that the dataset has 506 rows and 14 columns.

Creating a box plot for the Median Value of Owner-Occupied Homes

In [6]:
```
## Create a box plot for the Median value of owner-occupied homes

## Set the grid layout and create the boxplot
sns.set_style("darkgrid", {"grid.color": ".8", "grid.linestyle": ":"})
plt.rcParams["figure.figsize"] = [7, 7]
plt.rcParams["figure.autolayout"] = True
ax = sns.boxplot(y="MEDV", data=boston_df)
```

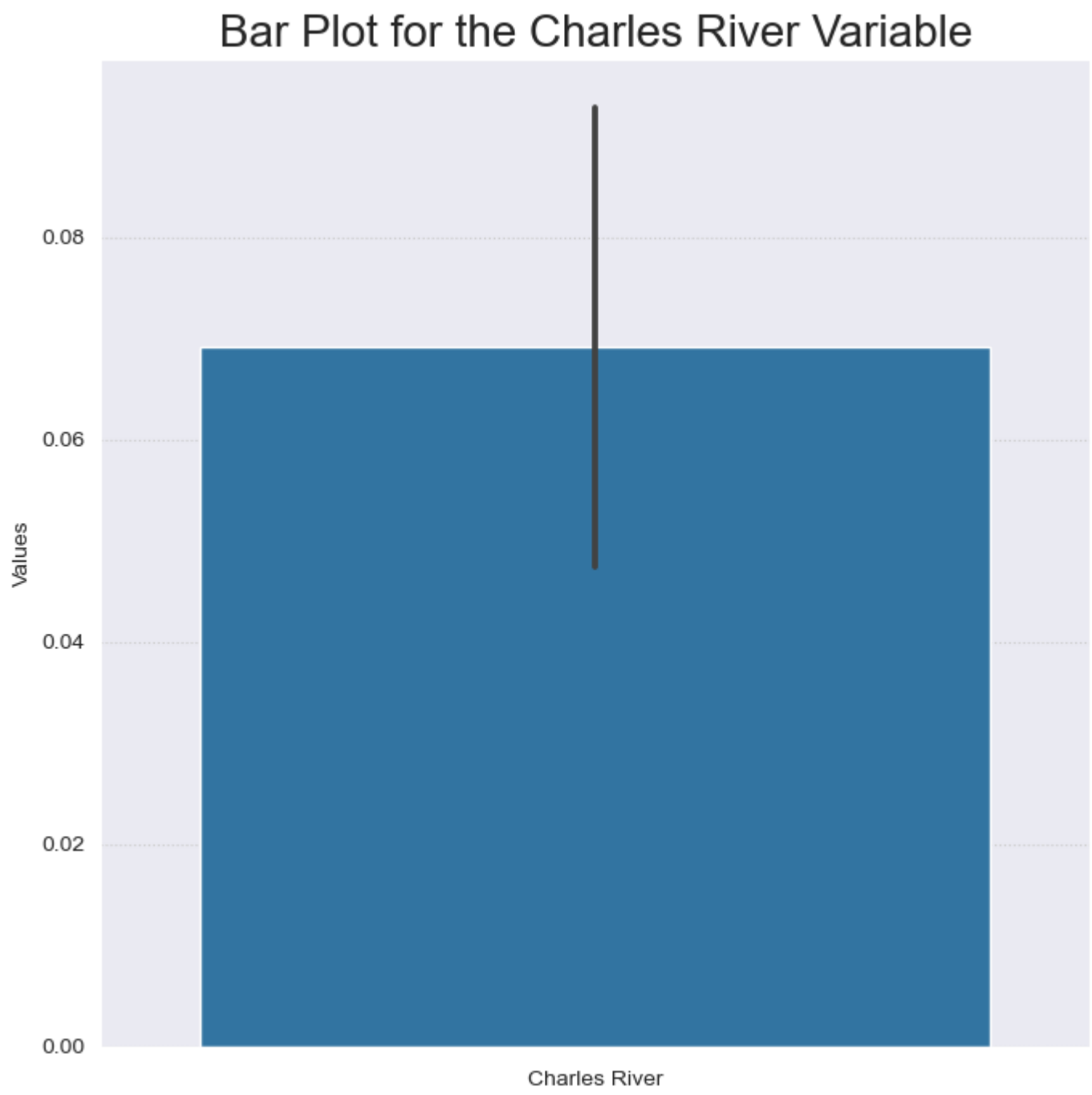
```
## Create the Graph's Title and Axis labels
plt.xlabel("Owners")
plt.ylabel("Values")
plt.title("Box Plot for the Median Value of Owner-occupied Homes in $1000's", fontsize=20)

## Display the Figure
plt.show()
```


Box Plot for the Median Value of Owner-occupied Homes in $1000's
50
40
30
20
10
Values
Owners

Here, the result shows that the Median Value of Owner-Occupied Homes in $1000's is slightly above 20, the minimum value is btween 0 and 10, the first quartile is approaching 20, the third quartile is about 25, and the maximum value is approaching 40.

Plotting a Bar Plot for the Charles River variable

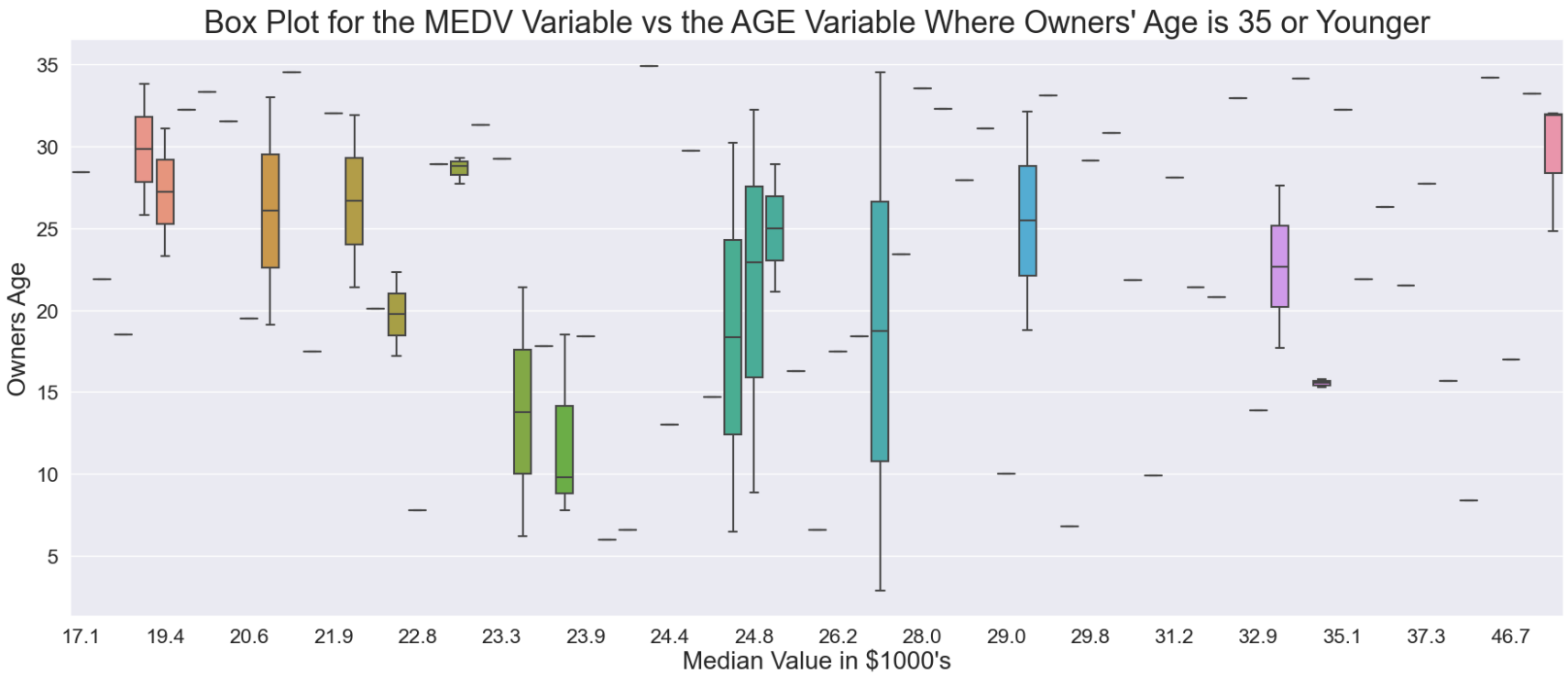
In [7]:
```
## Plot the barplot for the Charles river variable

## Set the grid layout and create the barplot
sns.set_style("darkgrid", {"grid.color": ".8", "grid.linestyle": ":"})
plt.rcParams["figure.figsize"] = [7, 7]
plt.rcParams["figure.autolayout"] = True
ax = sns.barplot(y="CHAS", data=boston_df)

## Create the Graph's Title and Axis labels
plt.xlabel("Charles River")
plt.ylabel("Values")
plt.title("Bar Plot for the Charles River Variable", fontsize=20)

## Display the figure
plt.show()
```



According to the Bar Plot, the Charles River average value is about 0.07.

Creating boxplot for the MEDV variable vs the AGE variable for owners who are 35 and younger

In [8]:
```
## Create a box plot for the MEDV variable vs the AGE variable where the age variable is 35 years and younger

## Query the dataframe's age range for 35 or younger
young_df=boston_df.query('AGE <= 35')

## Set the grid layout, grid size, color, and axis font size
sns.set_style("darkgrid", {"grid.color": ".8", "grid.linestyle": ":"})
sns.set(font_scale = 1.5)
plt.rcParams["figure.figsize"] = [18, 8]
plt.rcParams["figure.autolayout"] = True

## Create the boxplot
ax = sns.boxplot(x="MEDV", y="AGE", data=young_df)

## Iterate ax.get_xticklabels() method. If index is even at intervals of four, then make them visible; else, don't.
for index, label in enumerate(ax.get_xticklabels()):
   if index % 4 == 0:
      label.set_visible(True)
   else:
      label.set_visible(False)

## Create the Graph's Title and Axis labels and set the font sizes
plt.xlabel("Median Value in $1000's", fontsize=20)
plt.ylabel("Owners Age", fontsize=20)
plt.title("Box Plot for the MEDV Variable vs the AGE Variable Where Owners' Age is 35 or Younger", fontsize=25)

## Display the figure
plt.show()
```

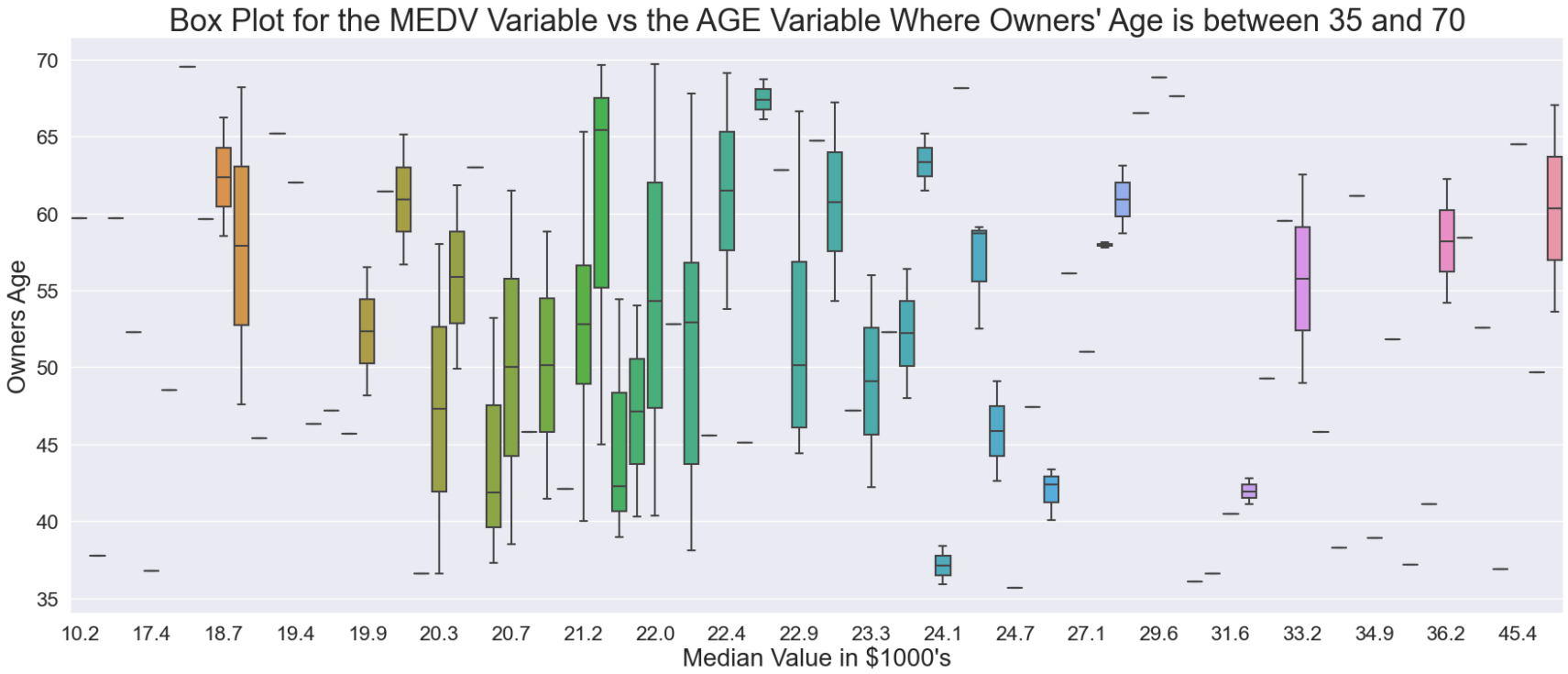


According to the graph, there are very few owners ages 35 or younger who own properties with the Median Value in $1000's.

Creating boxplot for the MEDV variable vs the AGE variable for owners who are between 35 and 70 years old

In [9]:
```
## Create box plot for the MEDV variable vs the AGE variable where the age variable is between 35 and 70

## Query the dataframe's age range for ages between 35 and 70
mid_df=boston_df.query('AGE > 35 & AGE < 70')

## Set the grid layout, grid size, color, and axis font size
sns.set_style("darkgrid", {"grid.color": ".8", "grid.linestyle": ":"})
sns.set(font_scale = 1.5)
plt.rcParams["figure.figsize"] = [18, 8]
plt.rcParams["figure.autolayout"] = True

## Create the boxplot
ax = sns.boxplot(x="MEDV", y="AGE", data=mid_df)
```

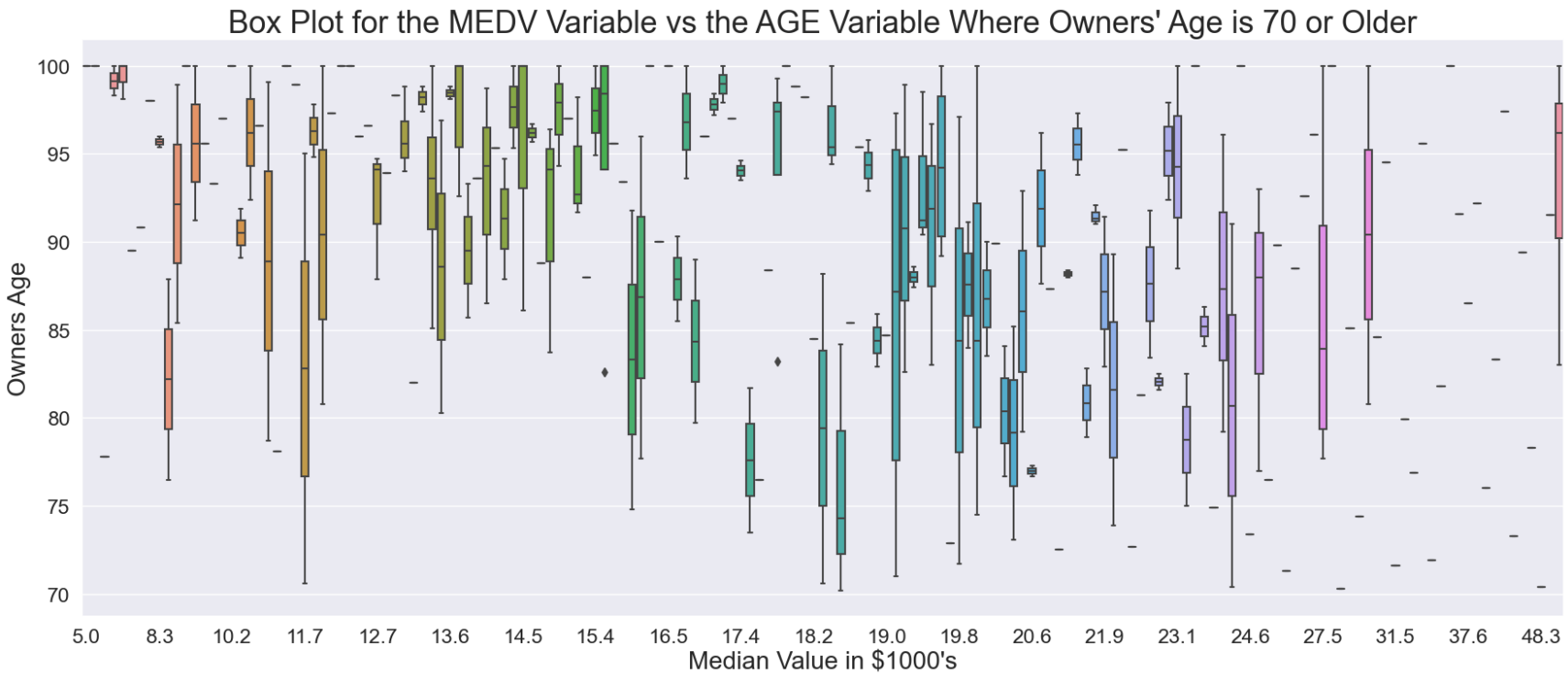
```
## Iterate ax.get_xticklabels() method. If index is even at intervals of four, then make them visible; else, don't.
for index, label in enumerate(ax.get_xticklabels()):
   if index % 4 == 0:
      label.set_visible(True)
   else:
      label.set_visible(False)

## Create the Graph's Title and Axis labels and set the font sizes
plt.xlabel("Median Value in $1000's", fontsize=20)
plt.ylabel("Owners Age", fontsize=20)
plt.title("Box Plot for the MEDV Variable vs the AGE Variable Where Owners' Age is between 35 and 70", fontsize=25)

## Display the figure
plt.show()
```



This graph shows that owners who are between 35 and 70 years old are twice as likely to own properties with the Median Value in $1000's than those who are 35 or younger.

Creating boxplot for the MEDV variable vs the AGE variable for owners whose ages are 70 years or older

In [10]:
```
## Create a box plot for the MEDV variable vs the AGE variable where the age variable is greater than or equal to 70

## Query the dataframe's age range for ages between 70 or older
old_df=boston_df.query('AGE >= 70')

## Set the grid layout, grid size, color, and axis font size
sns.set_style("darkgrid", {"grid.color": ".8", "grid.linestyle": ":"})
sns.set(font_scale = 1.5)
plt.rcParams["figure.figsize"] = [18, 8]
plt.rcParams["figure.autolayout"] = True

## Create the boxplot
ax = sns.boxplot(x="MEDV", y="AGE", data=old_df)

## Iterate ax.get_xticklabels() method. If index is even at intervals of eight, then make them visible; else, don't.
for index, label in enumerate(ax.get_xticklabels()):
   if index % 8 == 0:
      label.set_visible(True)
   else:
      label.set_visible(False)

## Create the Graph's Title and Axis labels and set the font sizes
plt.xlabel("Median Value in $1000's", fontsize=20)
plt.ylabel("Owners Age", fontsize=20)
plt.title("Box Plot for the MEDV Variable vs the AGE Variable Where Owners' Age is 70 or Older", fontsize=25)

## Display the figure
plt.show()
```

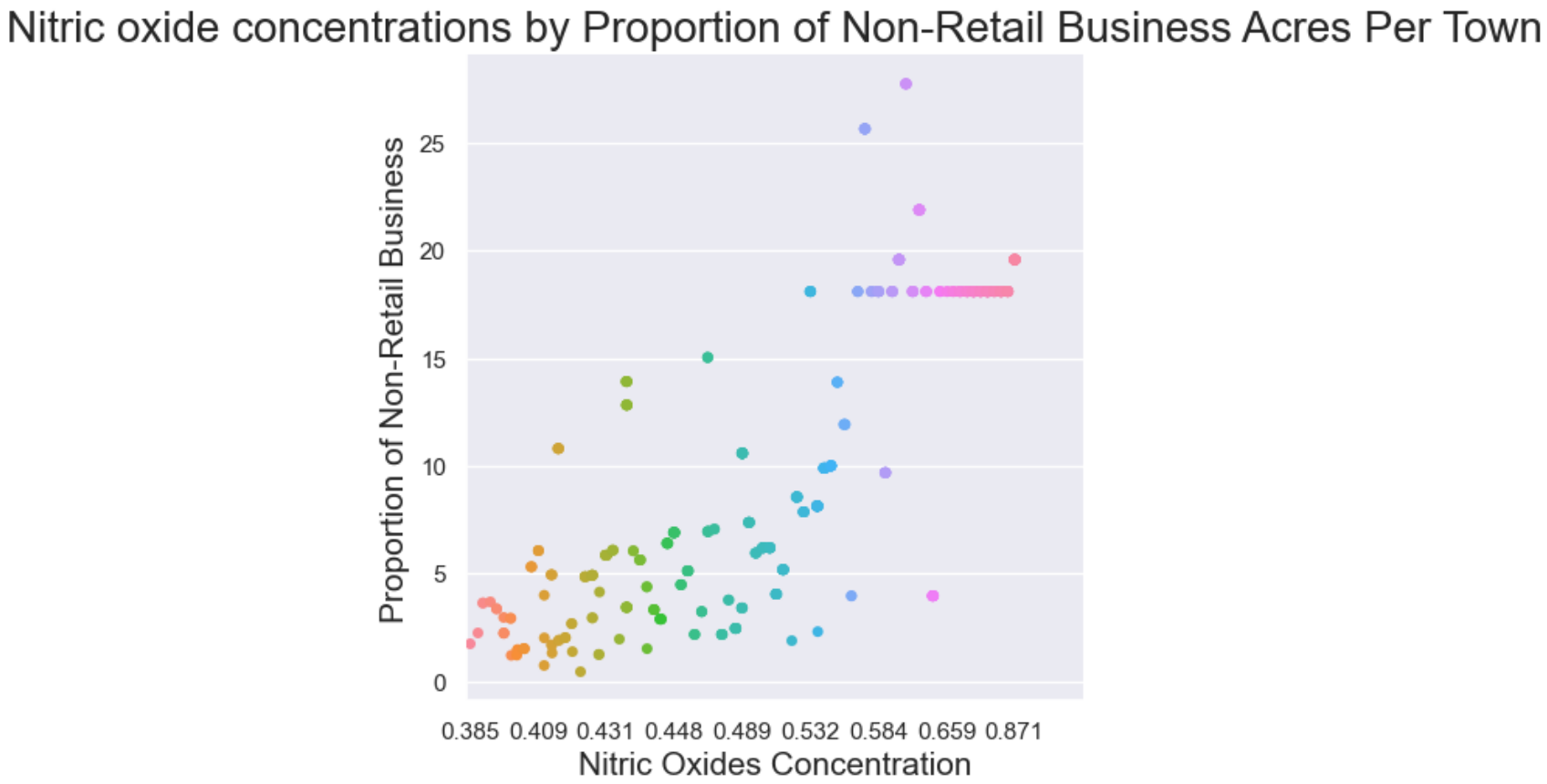


Based on this graph owners whose ages are 70 years or older have the highest concentration of property ownership with the Median Value in $1000's than the two previous groups.

Creating a scatter plot to show the relationship between Nitric oxide concentrations and the proportion of non-retail business acres per town

In [11]:
```
## Scatter plot to show the relationship between Nitric oxide concentrations and the proportion of non-retail business

## Set the grid layout, grid size, color, and axis font size
sns.set_style("darkgrid", {"grid.color": ".8", "grid.linestyle": ":"})
sns.set(font_scale = 1)
plt.rcParams["figure.figsize"] = [8, 8]
plt.rcParams["figure.autolayout"] = True

## Create the scatter plot
ax = sns.catplot(x='NOX', y= 'INDUS', data=boston_df)

## Create the Graph's Title and Axis labels, set the font sizes and ticks density
plt.xlabel("Nitric Oxides Concentration", fontsize=15)
```

```
plt.ylabel("Proportion of Non-Retail Business", fontsize=15)
plt.title("Nitric oxide concentrations by Proportion of Non-Retail Business Acres Per Town", fontsize=20)
plt.xticks([0, 10, 20, 30, 40, 50, 60, 70, 80, 90])

## Display the figure
plt.show()
```



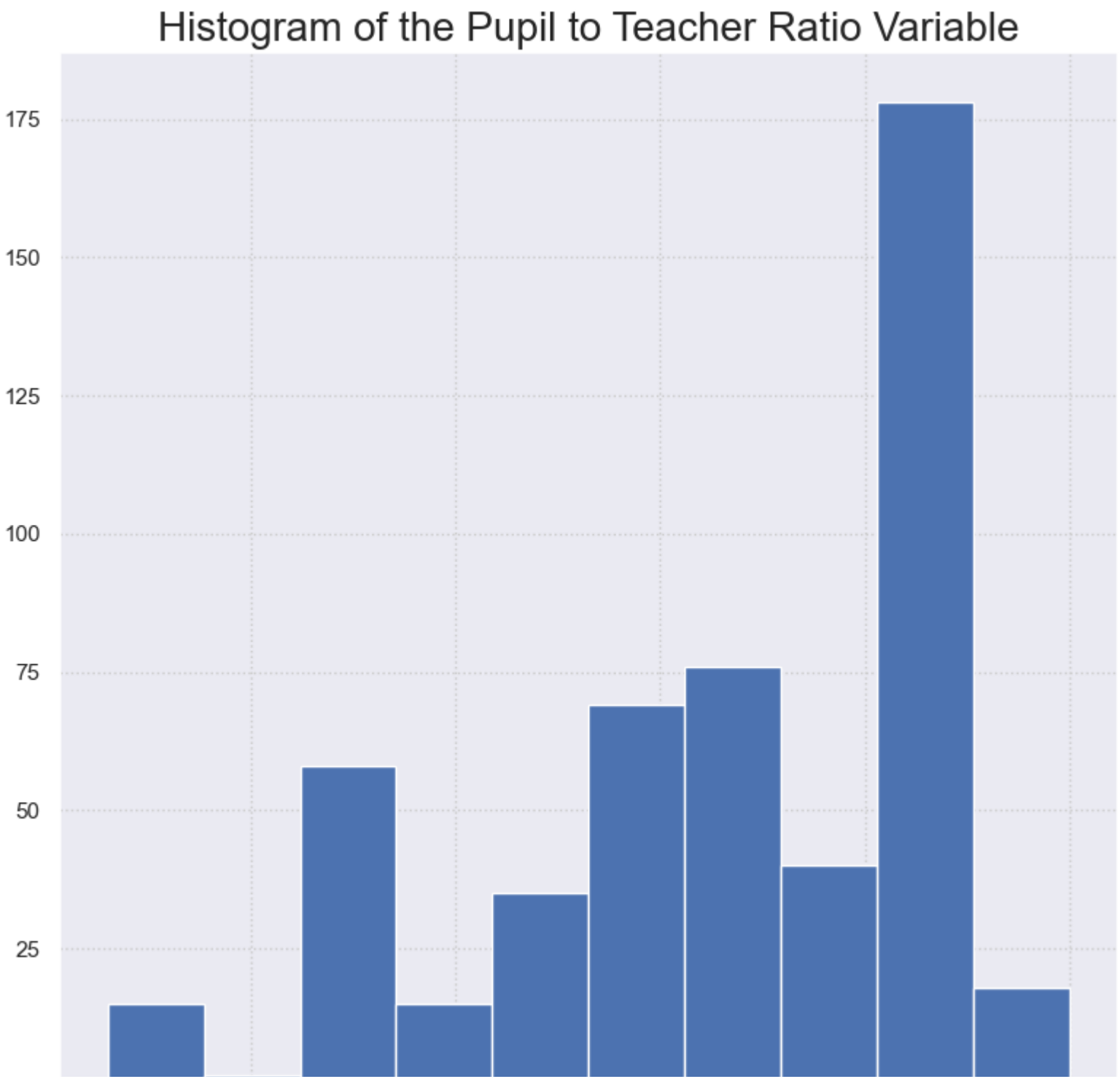
The scatter plot demonstrates that the Nitric Oxides Concentration level is higher as the Proportion of Non-Retail Business is also higher.

Creating a histogram for the Pupil to Teacher Ratio Variable

In [12]:
```
## Creating a histogram for the pupil to teacher ratio variable using the matplotlib library

## Set the grid background color
sns.set_style("darkgrid", {"grid.color": ".8", "grid.linestyle": ":"})
```

```
## Create the histogram for
plt.hist(boston_df['PTRATIO'])

## Create the histogram's Title and set the font size
plt.title("Histogram of the Pupil to Teacher Ratio Variable", fontsize=20)
```

Out[12]:
```
Text(0.5, 1.0, 'Histogram of the Pupil to Teacher Ratio Variable')
```


Histogram of the Pupil to Teacher Ratio Variable
175
150
125
100
75
50
25

This histogram is left-skewed. It does not represent the population evenly and is bias towards the higher end of the range.

Created by Fritz Tardieu (February 2023)

Thank you!

In [ ]: