Assignment 1.1 AAI-500

Name: Ajmal Jalal

Date: 09/06/2024

For this assignment, you will refer to the textbook to solve the practice exercises. **Use Python to answer any coding problems (not R, even if indicated in your textbook).** Use Jupyter Notebook, Google Colab, or a similar software program to complete your assignment. Submit your answers as a **PDF or HTML** file. As a best practice, always label your axes and provide titles for any graphs generated on this assignment. Round all quantitative answers to 2 decimal places.

Problem # 1.1.

In the 2018 election for Senate in California, a CNN exit poll of 1882 voters stated that 52.5% voted for the Democratic candidate, Diane Feinstein. Of all 11.1 million voters, 54.2% voted for Feinstein.

(a) What was the (i) subject, (ii) sample, (iii) population?

Answer:

Subject: The 2018 election for Senate in California

Sample: 1882 voters

Population: 11.1 million voters in California

Problem # 1.2.

The `Students` data file at http://stat4ds.rwth-aachen.de/data/Students.dat responses of a class of 60 social science graduate students at the University of Florida to a questionnaire that asked about *gender* (1 = female, 0 = male), *age*, *hsgpa* = high school GPA (on a four-point scale), cogpa = college GPA, *dhome* = distance (in miles) of the campus from your home town, *dres* = distance (in miles) of the classroom from your current residence, *tv* = average number of hours per week that you watch TV, *sport* = average number of hours per week that you participate in sports or have other physical exercise, *news* = number of times a week you read a newspaper, *aids* = number of people you know who have died from AIDS or who are HIV+, *veg* = whether you are a vegetarian (1 = yes, 0 = no), *affil* = political affiliation (1 = Democrat, 2 = Republican, 3 =

independent), *ideol* = political ideology (1 = very liberal, 2 = liberal, 3 = slightly liberal, 4 = moderate, 5 = slightly conservative, 6 = conservative, 7 = very conservative), *relig* = how often you attend religious services (0 = never, 1 = occasionally, 2 = most weeks, 3 = every week), *abor* = opinion about whether abortion should be legal in the first three months of pregnancy (1 = yes, 0 = no), *affirm* = support affirmative action (1 = yes, 0 = no), and *life* = belief in life after death (1 = yes, 2 = no, 3 = undecided). You will use this data file for some exercises in this book.

(a) Practice accessing a data file for statistical analysis with your software by going to the book's website and copying and then displaying this data file.

Answer:

In [ ]:
```
# Importing the necessary libraries
from pandas import read_csv


# Loading the data
studnet_data = read_csv('https://stat4ds.rwth-aachen.de/data/Students.dat')

# Displaying the shape and the first few rows of the data (just to check if
print(studnet_data.shape)
print(studnet_data.head())
```

```
(60, 1)
  subject gender age hsgpa cogpa dhome dres tv sport news aids veg affil ide
ol relig abor affirm life
0  1     0  32  2.2  3.5  0     5.0  3   5  0   0...
1  2     1  23  2.1  3.5  1200  0.3  15  7  5   6...
2  3     1  27  3.3  3.0  1300  1.5  0   4  3   0...
3  4     1  35  3.5  3.2  1500  8    5   5  6   3...
4  5     0  23  3.1  3.5  1600  10   6   6  3   0...
```

(b) Using responses on *abor*, state a question that could be addressed with (i) descriptive statistics, (ii) inferential statistics.

Descriptive Statistics: What is the most common response towards abortion among the participants?

Inferential Statistics: Is there a difference in responses towards abortion (abor) between men and women in the data?

Problem # 1.4.

Give an example of a variable that is (a) categorical; (b) quantitative; (c) discrete; (d) continuous

Answer:

A catagorical variable: color_of_eyes

A Quantitative variable: number_of_cars

A discrete variable: hours_spent_on_tv

A continuous variable: shcool_gpa

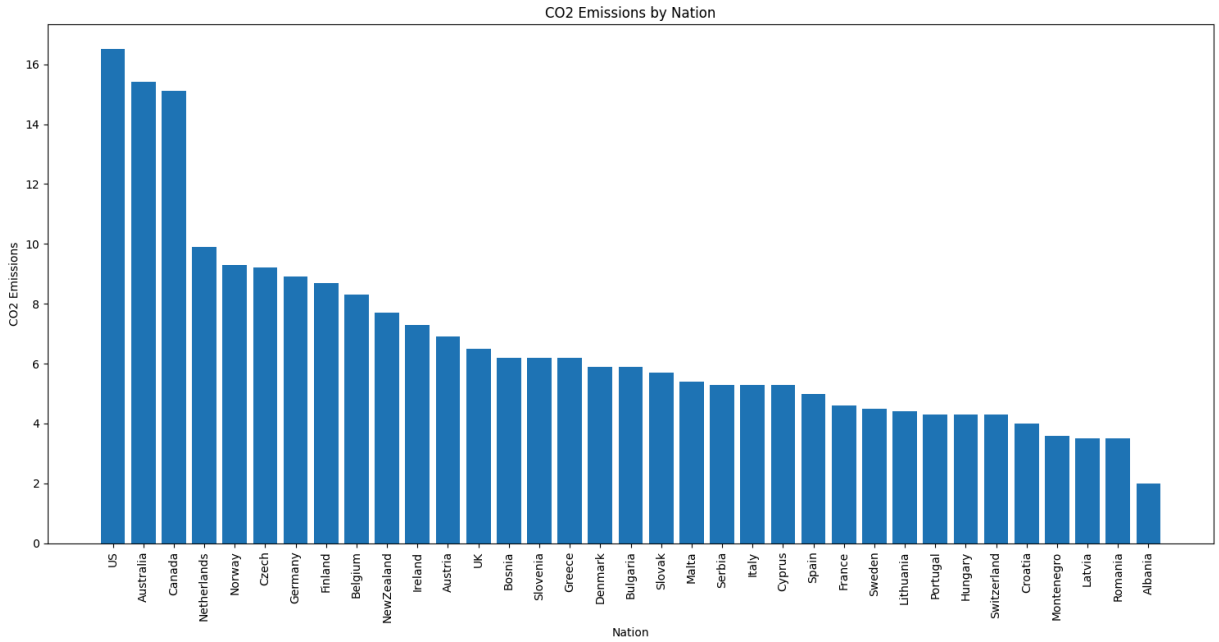
Problem # 1.10.

Analyze the `Carbon_West` (http://stat4ds.rwth-aachen.de/data/Carbon_West.dat) data file at the book's website by **(a)** constructing a frequency distribution and a histogram, **(b)** finding the mean, median, and standard deviation. Interpret each.

Answer:

`(a) Constructing frequency distribution and histogram`

In [ ]:
```
# Importing the necessary libraries
import pandas as pd
import matplotlib.pyplot as plt

# Loading the data from the provided URL and storing in a variable
carbon_data = pd.read_csv('http://stat4ds.rwth-aachen.de/data/Carbon_West.da
                          sep=r'\s+',
                          header=None,
                          names=['Nation', 'CO2'])
# The CO2 value is string, converting CO2 to numeric andreplacing any non-nu
carbon_data['CO2'] = pd.to_numeric(carbon_data['CO2'], errors='coerce')

# Remove any rows with NaN values if exist
carbon_data = carbon_data.dropna()

# Sort the data by CO2 emissions in descending order, this will help us to s
carbon_data_sorted = carbon_data.sort_values('CO2', ascending=False)

# (a) Construct bar plot
plt.figure(figsize=(15, 8))
plt.bar(carbon_data_sorted['Nation'], carbon_data_sorted['CO2'])
plt.title('CO2 Emissions by Nation')
plt.xlabel('Nation')
plt.ylabel('CO2 Emissions')
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```



(b) Calculate mean, median, and standard deviation

In [ ]:
```
# Calculating the mean, median, and standard deviation
mean = carbon_data['CO2'].mean()
median = carbon_data['CO2'].median()
std_dev = carbon_data['CO2'].std()

# Printing the results
print(f"Mean: {mean:.2f}")
print(f"Median: {median:.2f}")
print(f"Standard Deviation: {std_dev:.2f}")
```

```
Mean: 6.72
Median: 5.90
Standard Deviation: 3.36
```

Problem # 1.11.

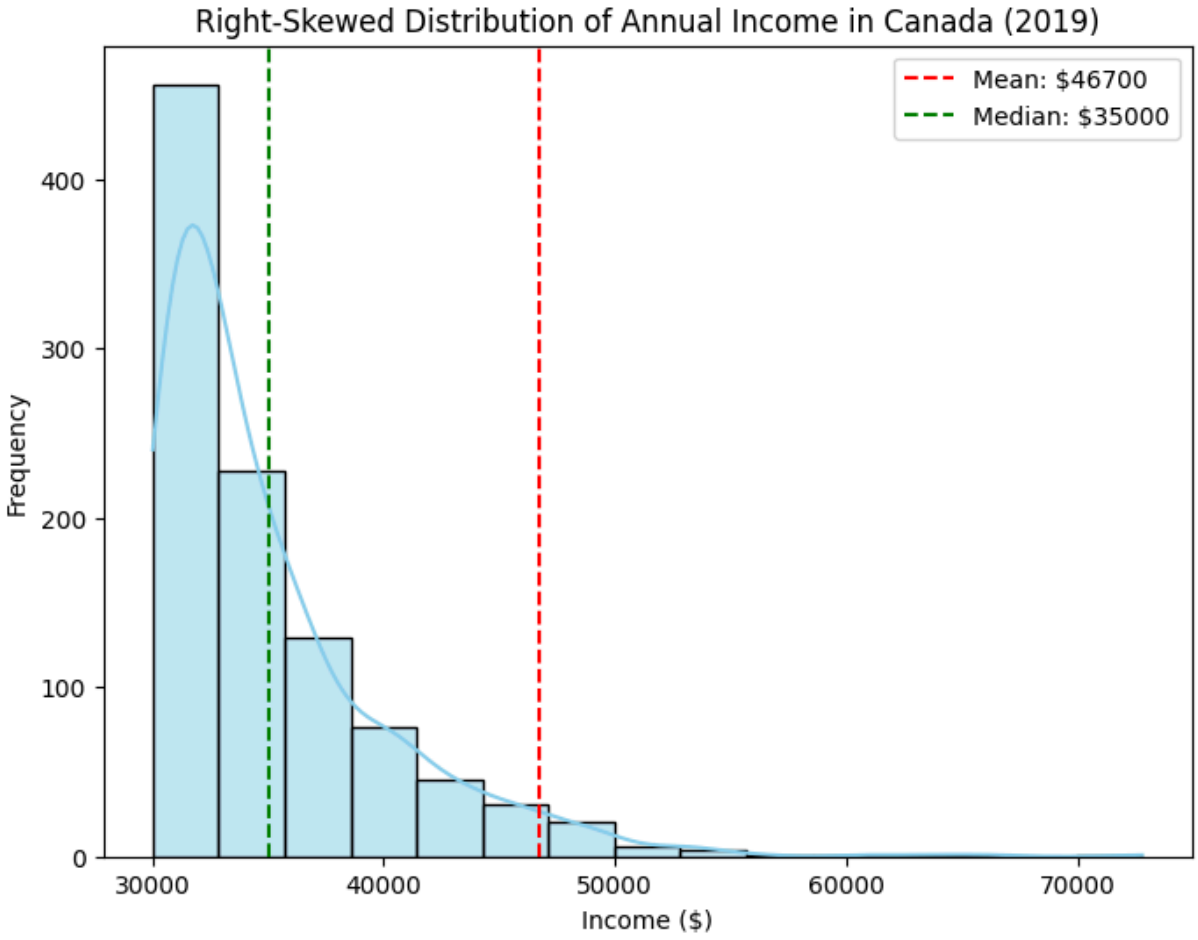
According to Statistics Canada, for the Canadian population having income in 2019, annual income had a median of $ 35,000 and mean of $ 46,700. What would you predict about the shape of the distribution? Why?

Answer:

Because the mean annual income $(46,700) is higher than the median (35,000)$, it suggests that the distribution of income in Canada is right-skewed (positively skewed) for the year 2019. This happens because a few individuals with very high incomes pull the mean upward, while the median remains closer to the income of the general population.

We can describe the situation better with a sample graph:

In [ ]:
```python
# Importing the necessary libraries
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Sample mean and median
median_income = 35000
mean_income = 46700

# Generating a right-skewed distribution
np.random.seed(0)
right_skewed_data = np.random.exponential(scale=5000, size=1000) + 30000

# Plotting the distribution
plt.figure(figsize=(8,6))
sns.histplot(right_skewed_data, bins=15, kde=True, color='skyblue', edgecolo
plt.axvline(mean_income, color='red', linestyle='--', label=f'Mean: ${mean_i
plt.axvline(median_income, color='green', linestyle='--', label=f'Median: ${
plt.title('Right-Skewed Distribution of Annual Income in Canada (2019)')
plt.xlabel('Income ($)')
plt.ylabel('Frequency')
plt.legend()
plt.show()
```



Problem # 1.13.

A report indicates that public school teacher's annual salaries in New York city have an approximate mean of `$` 69,000 and standard deviation of `$` 6,000. If the distribution has approximately a bell shape, report intervals that contain about (a) 68%, (b) 95%, (c) all or nearly all salaries. Would a salary of `$` 100,000 be unusual? Why?

Answer:

To answer this question, let's use the properties of a normal distribution (bell-shaped curve) and the empirical rule, also known as the 68-95-99.7 rule. We will break this down step by step:

Given information as bellow:

Mean (μ) = $69,000

Standard deviation (σ) = $6,000

Distribution is approximately bell-shaped (normal)

Calculating the intervals:

a - 68% of the data: This interval is μ ± 1σ, so $69,000 \pm$6,000 and that would be ($63,000$,75,000)

b - 95% of the data: This interval is μ ± 2σ, so $69,000 \pm (2 \times$6,000) = $69,000 \pm$12,000 and that would be ($57,000$,81,000)

c - All or nearly all (99.7%) of the data: This interval is μ ± 3σ $69,000 \pm (3 \times$6,000) $69,000 \pm$18,000 ($51,000$,87,000)

Now, a salary of $100,000 is more than 3 standard deviations above the mean$:100,000 - $69,000$ =31,000 $31,000 \div$6,000 ≈ 5.17 standard deviations

This is way outside the range that contains 99.7% of the data. Therefore, a salary of $100,000 would be considered unusual for a public school teacher in New York City, according to the the data and given distribution.

Problem # 1.17.

From the `Murder` data file (http://stat4ds.rwth-aachen.de/data/Murder.dat) at the book's website, use the variable murder, which is the murder rate (per 100,000 population) for each state in the U.S. in 2017 according to the FBI Uniform Crime Reports. At first, do not use the observation for D.C. (DC). Using software:
(a) Find the mean and standard deviation and interpret their values.
(b) Find the five-number summary, and construct the corresponding box plot. Interpret.

(c) Now include the observation for D.C. What is affected more by this outlier: The mean or the median? The range or the inter-quartile range?

Answer:

`(a) Finding the mean and standard deviation and interpret their values`

In [ ]:
```
# Importing the necessary libraries
import pandas as pd
import matplotlib.pyplot as plt

# Loading the data
data_url = 'http://stat4ds.rwth-aachen.de/data/Murder.dat'
dset = pd.read_csv(data_url, sep=r'\s+')

# Excluding DC initially
df_no_dc_data = dset[dset['state'] != 'DC']

# Calculating the mean and standard deviation
mean = df_no_dc_data['murder'].mean().round(2) # rounded to the nearest whol
std = df_no_dc_data['murder'].std().round(2) # rounded to the nearest whole
median = df_no_dc_data['murder'].median().round(2) # rounded to the nearest
range = df_no_dc_data['murder'].max() - df_no_dc_data['murder'].min()
iqr = df_no_dc_data['murder'].quantile(0.75) - df_no_dc_data['murder'].quant

print(f"(a) Mean: {mean}, Standard Deviation: {std}, Median: {median}, Range
```

```
(a) Mean: 4.87, Standard Deviation: 2.59, Median: 4.85, Range: 11.4, Interqu
artile Range: 3.55
```

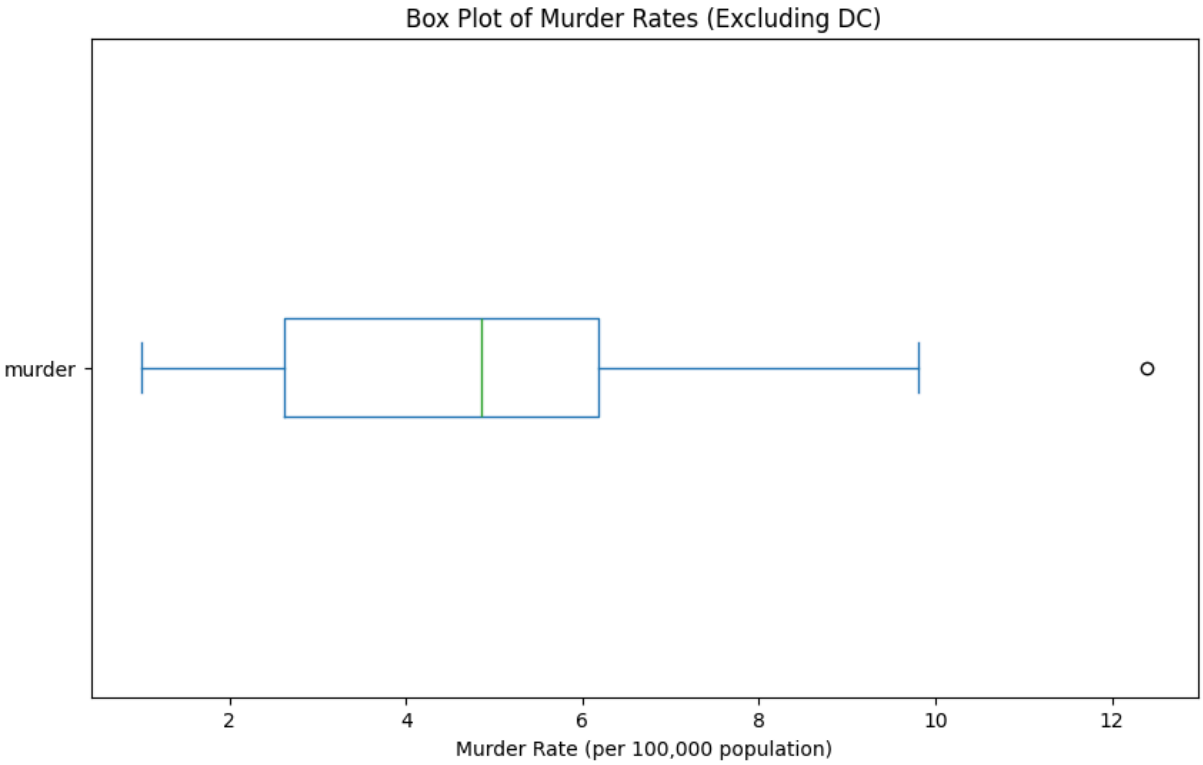
Interpretation:

The mean murder rate is almost 4.87 per 100,000 population and the median is 4.85 as well, with a standard deviation of 2.59. This indicates that on average, states have about 5 murders per 100,000 people, but there's considerable variation between states.

`(b) Finding the five-number summary, and construct the corresponding box plot.`

In [ ]:
```
# Five-number summary and box plot
summary = df_no_dc_data['murder'].describe()
print("Five-number summary:")
print(summary)

plt.figure(figsize=(10, 6))
df_no_dc_data['murder'].plot(kind='box', vert=False)
plt.title('Box Plot of Murder Rates (Excluding DC)')
plt.xlabel('Murder Rate (per 100,000 population)')
plt.show()
```

```
Five-number summary:
count    50.000000
mean      4.874000
std       2.586291
min       1.000000
25%       2.625000
50%       4.850000
75%       6.175000
max      12.400000
Name: murder, dtype: float64
```



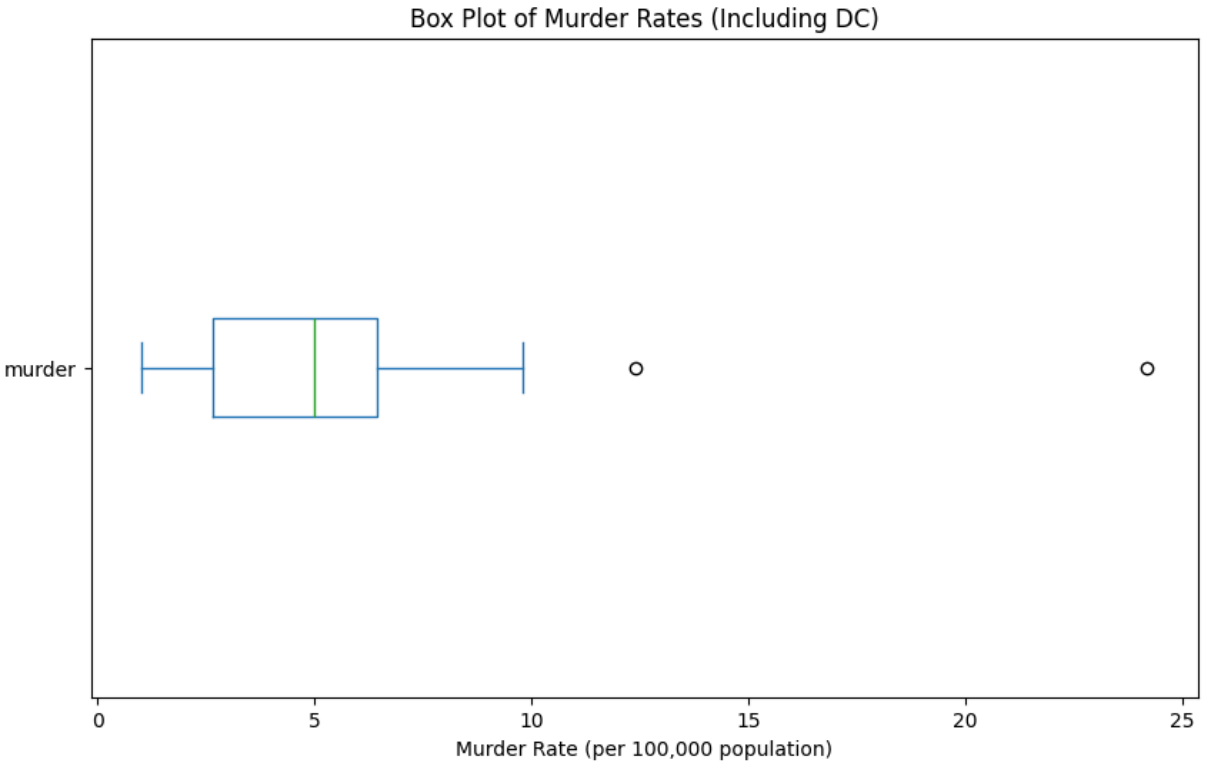
Interpretation:

The murder rate data across 50 U.S. states (excluding DC) shows a mean of 4.87 and a median of 4.85 murders per 100,000 population, which is a symmetric distribution. The standard deviation of 2.59 shows some variation among states. Murder rates range from a minimum of 1.0 to a maximum of 12.4, with the middle 50% of states falling between 2.63 (25th percentile) and 6.18 (75th percentile) murders per 100,000 population. The closeness of the mean and median, coupled with the range and interquartile range, suggests a fairly normal distribution with possible slight right-skewness and potential outliers on the high end. This conclusion can easily be read from the box plot as well. The box plot the median line is slightly below the center of the box, suggesting a slight positive skew. There are a few potential outliers on the high end.

`(c) Including the observation for D.C. What is affected more by this outlier: The mean or the median? The range or the inter-quartile range?`

In [ ]:
```
# Including DC
mean_with_dc = dset['murder'].mean()
median_with_dc = dset['murder'].median()
range_with_dc = dset['murder'].max() - dset['murder'].min()
iqr_with_dc = dset['murder'].quantile(0.75) - dset['murder'].quantile(0.25)
range_with_dc = dset['murder'].max() - dset['murder'].min()

print("\nIncluding DC:")
print(f"Mean: {mean_with_dc:.2f}")
print(f"Median: {median_with_dc:.2f}")
print(f"Interquartile Range: {iqr_with_dc:.2f}")
print(f"Range: {range_with_dc:.2f}")

plt.figure(figsize=(10, 6))
dset['murder'].plot(kind='box', vert=False)
plt.title('Box Plot of Murder Rates (Including DC)')
plt.xlabel('Murder Rate (per 100,000 population)')
plt.show()
```

```
Including DC:
Mean: 5.25
Median: 5.00
Interquartile Range: 3.80
Range: 23.20
```



Interpretation:

Including DC in the dataset affects the summary statistics and the box plot. The mean increases more significantly (from 4.87 to 5.25) than the median (from 4.85 to 5.00), indicating that DC's murder rate is likely much higher than most states. The substantial increase in range (from 11.4 to 23.20) suggests that DC's rate is an extreme outlier, far

exceeding the previous maximum. The interquartile range also increases (from 3.55 to 3.80), but less dramatically, demonstrating its robustness to outliers. These changes highlight DC's unique status, likely due to its urban nature, and show how a single extreme value can impact different statistical measures, with the mean and range being more sensitive than the median and interquartile range.

Problem # 1.18.

The `Income` data file (http://stat4ds.rwth-aachen.de/data/Income.dat) at the book's website reports annual income values in the U.S., in thousands of dollars.

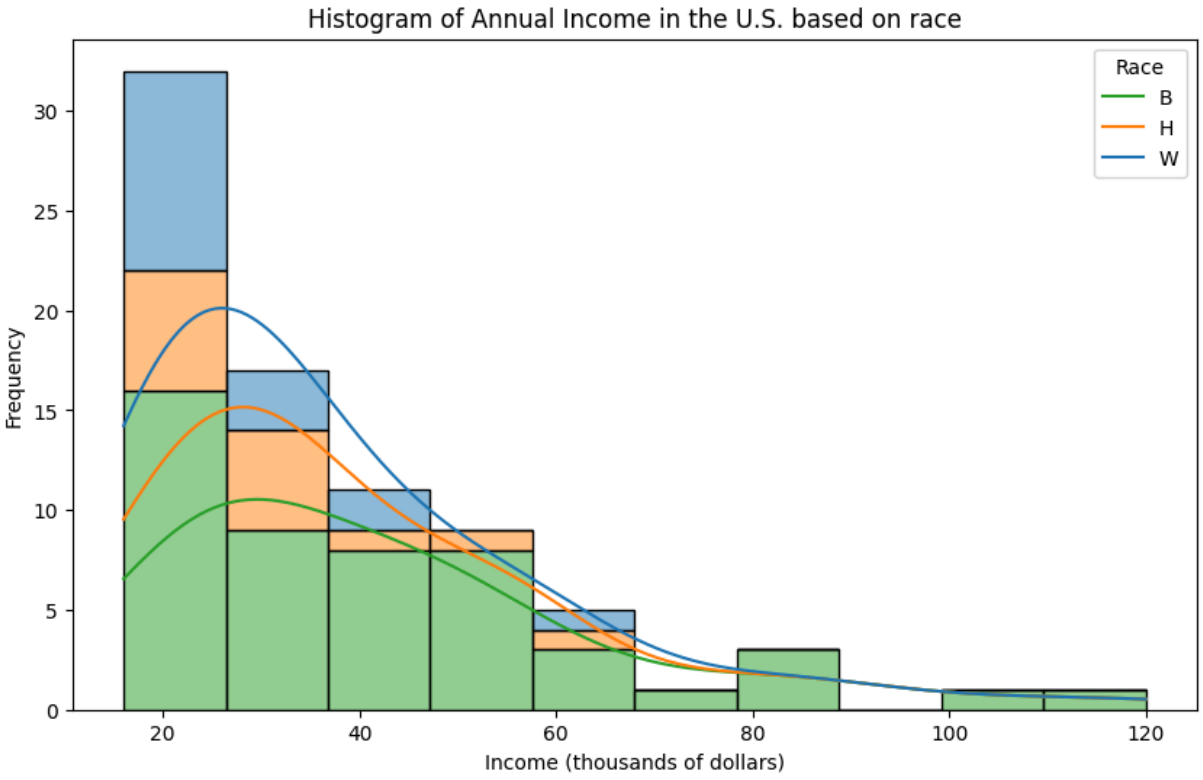
(a) Using software, construct a histogram. Describe its shape.
(b) Find descriptive statistics to summarize the data. Interpret them.
(c) The kernel density estimation method finds a smooth-curve approximation for a histogram. At each value, it takes into account how many observations are nearby and their distance, with more weight given those closer. Increasing the bandwidth increases the influence of observations further away. Plot a smooth-curve approximation for the histogram of income values. Summarize the impact of increasing and of decreasing the bandwidth substantially from the default value.
(d) Construct and interpret side-by-side box plots of income by race (B = Black, H = Hispanic, W = White). Compare the incomes using numerical descriptive statistics

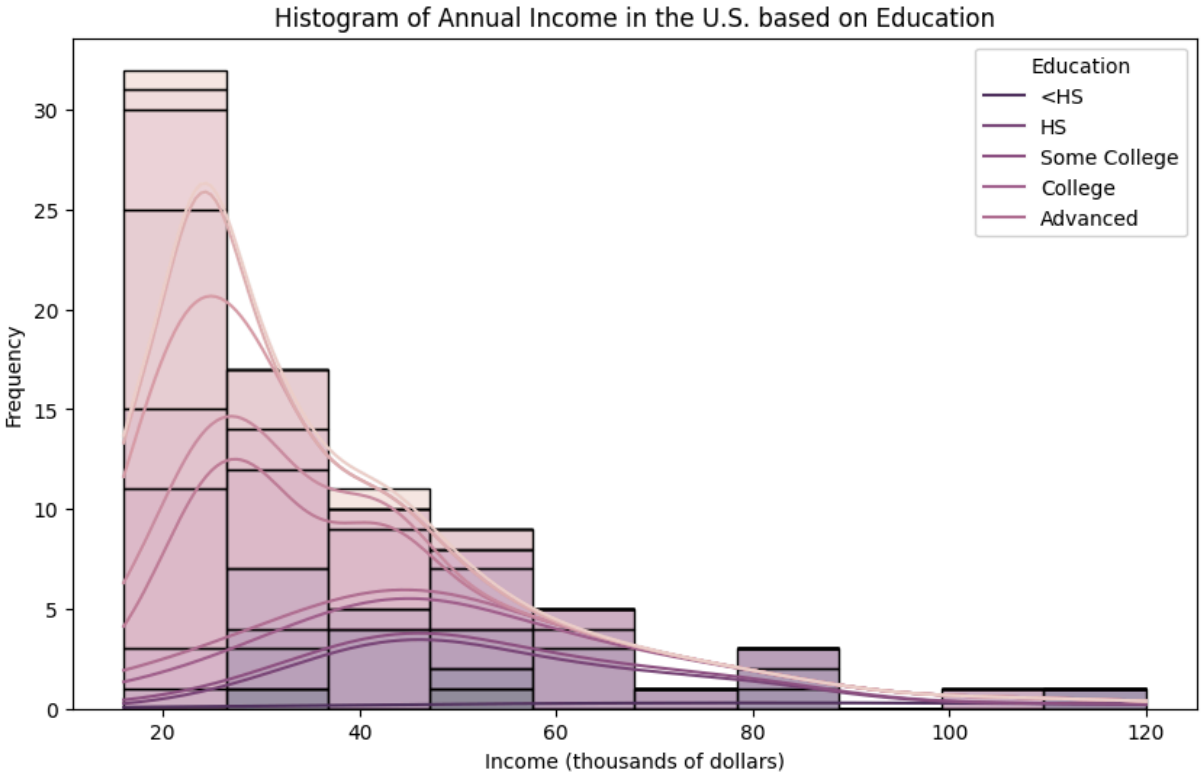
Answer:

`(a) Using software, construct a histogram and describe its shape.`

In [ ]:
```
# Importing the necessary libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Loading the data
data_url = "http://stat4ds.rwth-aachen.de/data/Income.dat"
income_data = pd.read_csv(data_url, sep=r'\s+')

# Creating a histogram (Income based on Race)
plt.figure(figsize=(10, 6))
sns.histplot(data=income_data, x='income', hue='race', multiple='stack', kde
plt.title('Histogram of Annual Income in the U.S. based on race')
plt.xlabel('Income (thousands of dollars)')
plt.ylabel('Frequency')
plt.legend(title='Race', labels=['B', 'H', 'W'])
plt.show()

# Creating a histogram (Income based on Education)
plt.figure(figsize=(10, 6))
sns.histplot(data=income_data, x='income', hue='education', multiple='stack'
```

```
plt.title('Histogram of Annual Income in the U.S. based on Education')
plt.xlabel('Income (thousands of dollars)')
plt.ylabel('Frequency')
plt.legend(title='Education', labels=['<HS', 'HS', 'Some College', 'College'
plt.show()
```





Shape of the histograms:

The histogram of annual income in the U.S. based on race and education appears to be right-skewed. It has a peak around 20-30 thousand dollars and a long tail extending to the right, Indicating that while most incomes are clustered at lower values, There are some individuals with much higher incomes pulling the distribution to the right. It indicates that individuals with higher educations seems to be earning more.

`(b) Finding descriptive statistics to summarize the data and interpret them`

In [ ]:
```
# Calculating descriptive statistics for income
income_stats = income_data['income'].describe()

# Displaying the statistics
print("Descriptive Statistics for Income:")
print(income_stats)

# Calculating additional statistics
median_income = income_data['income'].median()
skewness = income_data['income'].skew()
kurtosis = income_data['income'].kurtosis()

print(f"\nMedian Income: {median_income:.2f}")
print(f"Skewness: {skewness:.2f}")
print(f"Kurtosis: {kurtosis:.2f}")
```

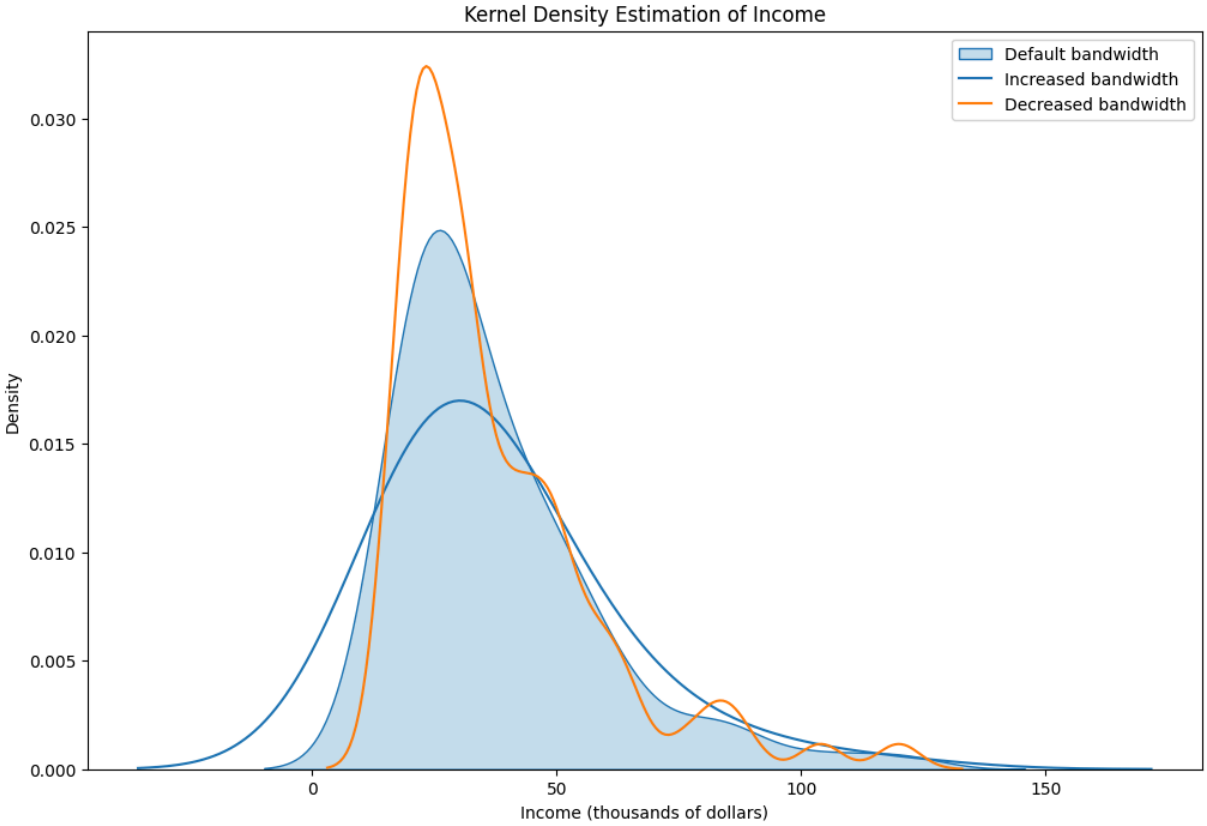
```
Descriptive Statistics for Income:
count     80.000000
mean      37.525000
std       20.672843
min       16.000000
25%       22.000000
50%       30.000000
75%       46.500000
max      120.000000
Name: income, dtype: float64

Median Income: 30.00
Skewness: 1.73
Kurtosis: 3.47
```

Interpretation:

1. The average (mean) income is $37.52 thousand.
2. The median income is $30.00 thousand, which is lower than the mean, indicating right-skewed distribution.
3. The standard deviation is $20.67 thousand, showing considerable variability in incomes.
4. The minimum income is $16.00 thousand, while the maximum is 120.00 thousand.
5. The positive skewness (1.73) confirms the right-skewed nature of the distribution.
6. The kurtosis (3.47) indicates the distribution has heavier tails than a normal distribution.

`(c) Ploting a smooth-curve approximation for the histogram of income values. Summarizing the impact of increasing and of decreasing the bandwidth substantially from the default value.`

In [ ]:
```
# Importing the necessary libraries
import seaborn as sns
import matplotlib.pyplot as plt

# Setting up the plot
plt.figure(figsize=(12, 8))

# Plotting KDE with default bandwidth
sns.kdeplot(data=income_data['income'], fill=True, label='Default bandwidth'

# Plotting KDE with increased bandwidth
sns.kdeplot(data=income_data['income'], bw_adjust=2, label='Increased bandwi

# Plotting KDE with decreased bandwidth
sns.kdeplot(data=income_data['income'], bw_adjust=0.5, label='Decreased banc

plt.title('Kernel Density Estimation of Income')
plt.xlabel('Income (thousands of dollars)')
plt.ylabel('Density')
plt.legend()
plt.show()
```



Impact of changing bandwidth:

1. Default bandwidth: Provides a balanced smoothing of the data.
2. Increased bandwidth: Results in a smoother curve, potentially obscuring local variations.
3. Decreased bandwidth: Produces a more detailed curve, potentially revealing local patterns but may be noisier.

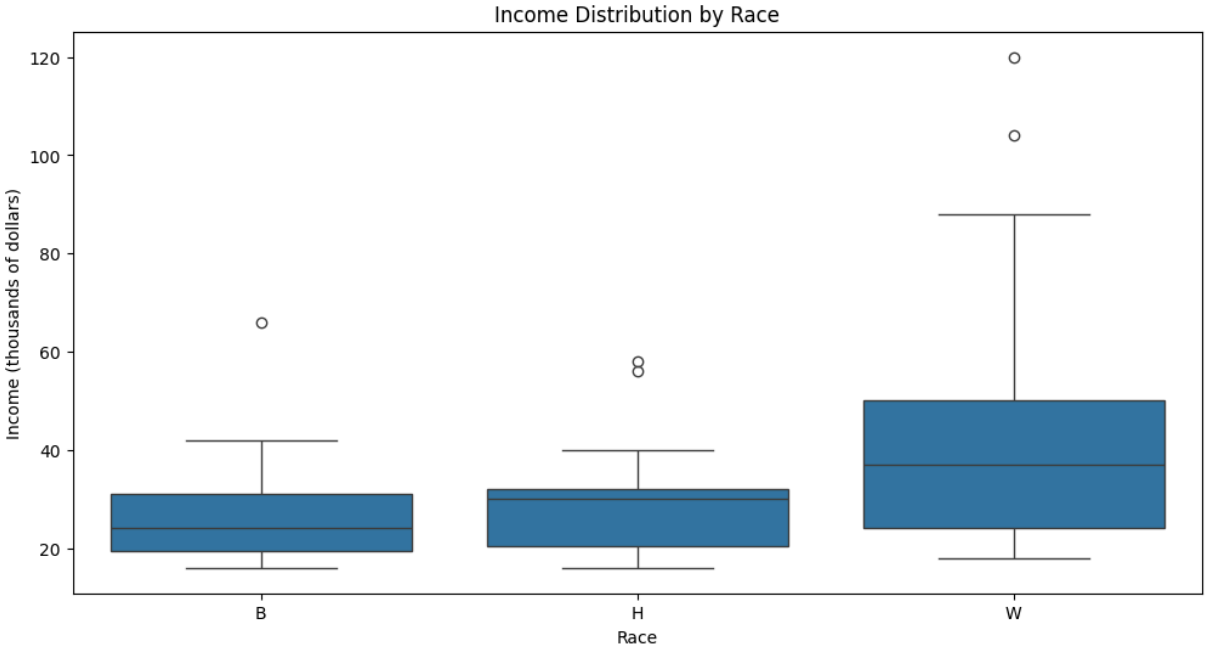
Increasing the bandwidth from the default value results in a smoother curve that may hide some local features of the data distribution. It gives more weight to observations that are further away, potentially oversimplifying the overall shape.

Decreasing the bandwidth from the default value leads to a more detailed and potentially noisier curve. It gives more weight to nearby observations, which can reveal local patterns in the data but may also be more sensitive to random fluctuations in the sample.

`(d) Constructing and interpret side-by-side box plots of income by race (B = Black, H = Hispanic, W = White).`

In [ ]:
```
# Constructing side-by-side box plots of income by race
plt.figure(figsize=(12, 6))
sns.boxplot(x='race', y='income', data=income_data)
plt.title('Income Distribution by Race')
plt.xlabel('Race')
plt.ylabel('Income (thousands of dollars)')
```

```
plt.show()

# Calculating numerical descriptive statistics
race_stats = income_data.groupby('race')['income'].describe()
print("Descriptive Statistics of Income by Race:")
print(race_stats)

# Calculating additional statistics
race_stats['median'] = income_data.groupby('race')['income'].median()
race_stats['iqr'] = income_data.groupby('race')['income'].quantile(0.75) - i

# Interpreting the results
print("\nInterpretation:")
print("1. Central Tendency:")
for race in ['B', 'H', 'W']:
    print(f"   - {race}: Median income is ${race_stats.loc[race, 'median']:.

print("\n2. Spread:")
for race in ['B', 'H', 'W']:
    print(f"   - {race}: IQR is ${race_stats.loc[race, 'iqr']:.2f}k")
```



```
Descriptive Statistics of Income by Race:
      count   mean        std   min   25%   50%   75%    max
race
B      16.0  27.75  13.284076  16.0  19.5  24.0  31.0   66.0
H      14.0  31.00  12.812254  16.0  20.5  30.0  32.0   58.0
W      50.0  42.48  22.869854  18.0  24.0  37.0  50.0  120.0

Interpretation:
1. Central Tendency:
   - B: Median income is $24.00k
   - H: Median income is $30.00k
   - W: Median income is $37.00k

2. Spread:
   - B: IQR is $11.50k
   - H: IQR is $11.50k
   - W: IQR is $26.00k
```

Problem # 1.19.

The `Houses` data file (http://stat4ds.rwth-aachen.de/data/Houses.dat) at the book's website lists the selling price (thousands of dollars), size (square feet), tax bill (dollars), number of bathrooms, number of bedrooms, and whether the house is new (1 = yes, 0 = no) for 100 home sales in Gainesville, Florida. Let's analyze the selling prices.

(a) Construct a frequency distribution and a histogram. Describe the shape.
(b) Find the percentage of observations that fall within one standard deviation of the mean. Why is this not close to 68%?
(c) Construct a box plot, and interpret.
(d) Use descriptive statistics to compare selling prices according to whether the house is new.

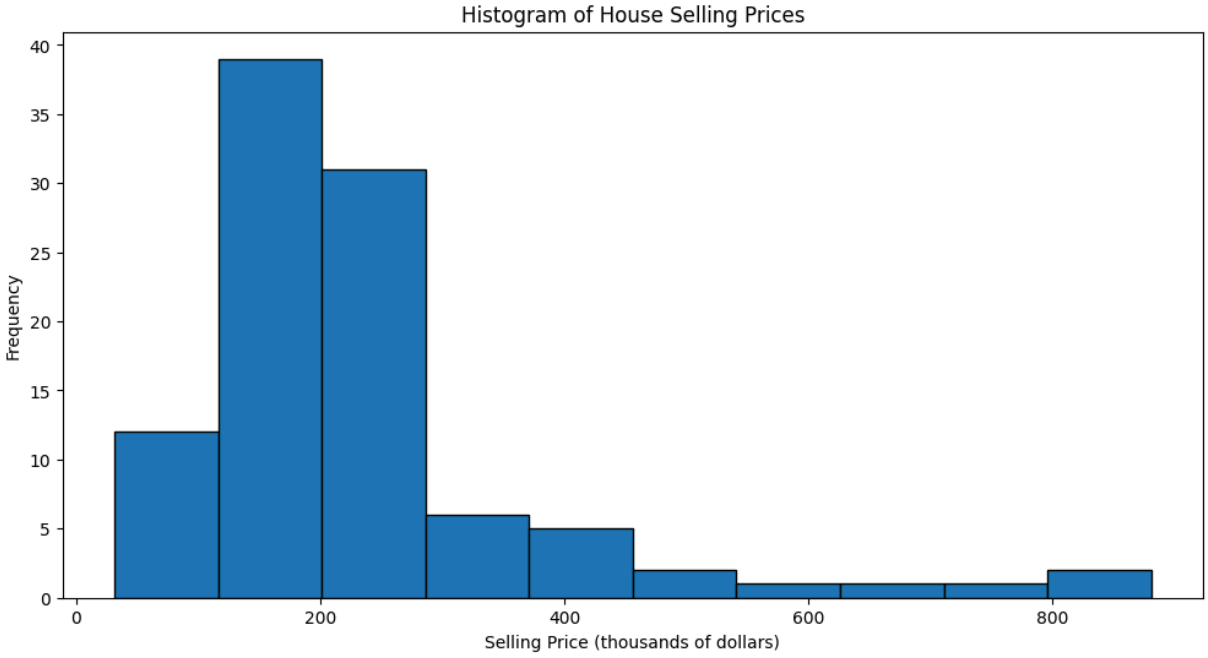
Answer:

`(a) Constructing a frequency distribution and a histogram and escribe the shape of it.`

In [ ]:
```python
# Import necessary libraries
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# Loading the data
data_url = "http://stat4ds.rwth-aachen.de/data/Houses.dat"
houses_data = pd.read_csv(data_url, sep=r'\s+')

# Extracting selling prices
selling_prices = houses_data['price']

# Creating a frequency distribution
```

```python
freq_dist = pd.cut(selling_prices, bins=10).value_counts().sort_index()
print("Frequency Distribution of Selling Prices:")
print(freq_dist)

# Creating a histogram
plt.figure(figsize=(12, 6))
plt.hist(selling_prices, bins=10, edgecolor='black')
plt.title('Histogram of House Selling Prices')
plt.xlabel('Selling Price (thousands of dollars)')
plt.ylabel('Frequency')
plt.show()
```

```
Frequency Distribution of Selling Prices:
price
(30.651, 116.4]    12
(116.4, 201.3]     39
(201.3, 286.2]     31
(286.2, 371.1]      6
(371.1, 456.0]      5
(456.0, 540.9]      2
(540.9, 625.8]      1
(625.8, 710.7]      1
(710.7, 795.6]      1
(795.6, 880.5]      2
Name: count, dtype: int64
```



Description of the shape:

The histogram of selling prices appears to be right-skewed (positively skewed). This means that while most houses are clustered around lower to middle price ranges, there are some houses with significantly higher prices, creating a tail on the right side of the distribution. Looks like most of the houses are piced between 116k and 286k.

`(b) Finding the percentage of observations that fall within one standard deviation of the mean. Why is this not close to 68%?`

In [ ]:
```python
# Calculating mean and standard deviation
mean_price = selling_prices.mean()
std_price = selling_prices.std()

# Calculating the range for one standard deviation
lower_bound = mean_price - std_price
upper_bound = mean_price + std_price

# Calculating the percentage of observations within one standard deviation
within_one_std = selling_prices[(selling_prices >= lower_bound) & (selling_p
percentage_within_one_std = (len(within_one_std) / len(selling_prices)) * 10

print(f"Mean price: ${mean_price:.2f}")
print(f"Standard deviation: ${std_price:.2f}")
print(f"Range: ${lower_bound:.2f} to ${upper_bound:.2f}")
print(f"Percentage within one standard deviation: {percentage_within_one_std
```

```
Mean price: $233.00
Standard deviation: $151.89
Range: $81.10 to $384.89
Percentage within one standard deviation: 85.00%
```

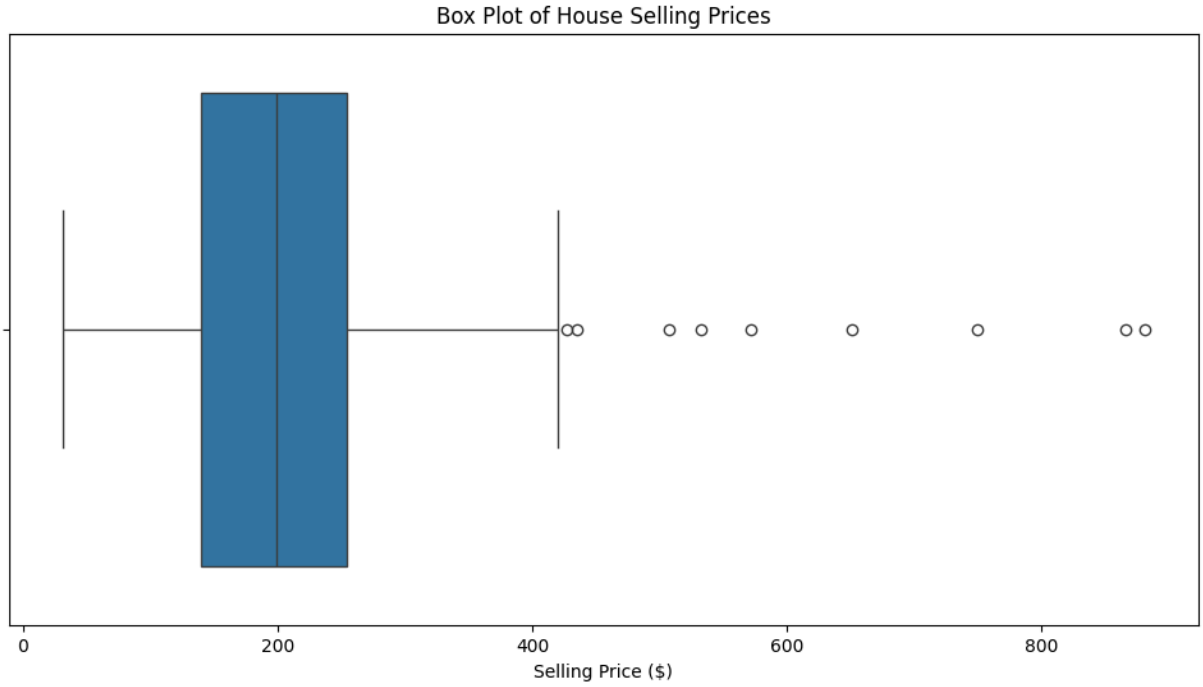
Explanation:

The percentage of observations within one standard deviation (85.00%) exceeds the expected 68% for a normal distribution due to:

1. As observed in the histogram, right-skewed, non-normal distribution
2. Presence of high-value outliers to the right
3. Data concentration on the left side

`(c) Constructing a box plot, and interpreting it`

In [ ]:
```python
# Creating a box plot of selling prices
plt.figure(figsize=(12, 6))
sns.boxplot(x=selling_prices)
plt.title('Box Plot of House Selling Prices')
plt.xlabel('Selling Price ($)')
plt.show()
```



Interpretation:

1. The box plot confirms the right-skewed distribution observed in the histogram.
2. The median (Q2) is closer to Q1 than Q3, further indicating right skewness.
3. There are numerous outliers on the high end of the price range.
4. The IQR (box) represents the middle 50% of the data, showing the bulk of house prices.
5. This visualization helps identify extreme values and the overall spread of house prices.

`(d) Using descriptive statistics to compare selling prices according to whether the house is new.`

In [ ]:
```
# Separating the data into new and old houses
new_houses = houses_data[houses_data['new'] == 1]
old_houses = houses_data[houses_data['new'] == 0]

# Calculating descriptive statistics for new and old houses
new_stats = new_houses['price'].describe()
old_stats = old_houses['price'].describe()

# Printing the descriptive statistics
print("Descriptive Statistics for New Houses:")
print(new_stats)
print("\nDescriptive Statistics for Old Houses:")
print(old_stats)

# Calculating and printing the median prices
new_median = new_houses['price'].median()
old_median = old_houses['price'].median()
print(f"\nMedian price for new houses: ${new_median:,.2f}")
```

```python
print(f"Median price for old houses: ${old_median:,.2f}")

# Calculating mean prices for new and old houses
new_mean = new_houses['price'].mean()
old_mean = old_houses['price'].mean()

# Calculating the difference in means
mean_difference = new_mean - old_mean

print(f"\nMean price for new houses: ${new_mean:,.2f}")
print(f"Mean price for old houses: ${old_mean:,.2f}")
print(f"Difference in mean prices: ${mean_difference:,.2f}")
```

```
Descriptive Statistics for New Houses:
count     11.000000
mean     436.445455
std      219.832789
min      158.850000
25%      256.950000
50%      427.500000
75%      519.675000
max      866.250000
Name: price, dtype: float64

Descriptive Statistics for Old Houses:
count     89.000000
mean     207.851124
std      121.039149
min       31.500000
25%      135.000000
50%      190.800000
75%      240.000000
max      880.500000
Name: price, dtype: float64

Median price for new houses: $427.50
Median price for old houses: $190.80

Mean price for new houses: $436.45
Mean price for old houses: $207.85
Difference in mean prices: $228.59
```