```
#include all libraries we will need

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
```

```
from google.colab import files
upload = files.upload()
```

Choose Files The Impact … Survey.csv

- **The Impact of Question Wording on Survey.csv**(text/csv) - 19500 bytes, last modified: 3/19/2025 - 100% done

```
Saving The Impact of Question Wording on Survey.csv to The Impact of Question Wording on Survey.csv
```

```
df = pd.read_csv('The Impact of Question Wording on Survey.csv')
```

```
#shows the first 3 lines from beginning
df.head(3)
```

| | Timestamp | Do you think the wording of a question can influence how people respond? | Which of the following factors do you think has the most influence on survey responses? (Select up to 2) | Have you ever felt that a survey question was designed to push a specific answer? | How satisfied are you with your life in general? \n(Scale 1 - 5) | How happy and fulfilling is your life right now? \n(Scale 1-5) | To what extent do you doubt the news you read? | To what extent do you doubt mainstream media, which has been criticized for misinformation? | In your opinion, how many hours does the average person spend on social media daily? (Enter a number) | Considering how addictive social media has become, how many hours do you think the average person wastes on social media daily? (Enter a number) | Which statement do you agree with more? | urg do th clim cha to \n(Sc 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 2025/03/18 1:04:37 PM GMT+2 | Yes, significantly | The wording of the question;The number of answ... | Yes, frequently | 3 | 2 | 3 | 2 | 3 | 2 | The climate crisis is a disaster, and governme... | |
| **1** | 2025/03/18 1:05:06 | Yes, | The order of questions in the | Sometimes | 3 | 4 | 2 | 3 | 4 | 4 | Climate change is a global issue | |

Next steps: Generate code with df View recommended plots New interactive sheet

```
#shows the first 3 lines from end
df.tail(3)
```

| | Timestamp | Do you think the wording of a question can influence how people respond? | Which of the following factors do you think has the most influence on survey responses? (Select up to 2) | Have you ever felt that a survey question was designed to push a specific answer? | How satisfied are you with your life in general? \n(Scale 1 - 5) | How happy and fulfilling is your life right now? \n(Scale 1-5) | To what extent do you doubt the news you read? | To what extent do you doubt mainstream media, which has been criticized for misinformation? | In your opinion, how many hours does the average person spend on social media daily? (Enter a number) | Considering how addictive social media has become, how many hours do you think the average person wastes on social media daily? (Enter a number) | Which statement do you agree with more? | ur do t cli ch t \n(S |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **59** | 2025/03/19 12:28:32 AM GMT+2 | Yes, significantly | The wording of the question;The time and place... | Yes, frequently | 4 | 3 | 4 | 2 | 4-7 hours | 4 | The climate crisis is a disaster, and governme... | |
| **60** | 2025/03/19 1:15:57 | Yes, | The wording of the question;The | Sometimes | 4 | 3 | 3 | 2 | 3 | 4 | The climate crisis is a disaster | |

```
# Get shape
df.shape
```

```
(62, 14)
```

```
# Print DataFrame info

print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 62 entries, 0 to 61
Data columns (total 14 columns):
 #   Column
---  ------
 0   Timestamp
 1   Do you think the wording of a question can influence how people respond?
 2   Which of the following factors do you think has the most influence on survey responses? (Select up to 2)
 3   Have you ever felt that a survey question was designed to push a specific answer?
 4   How satisfied are you with your life in general?
(Scale 1 - 5)                                                                          62 non-null     int64
 5   How happy and fulfilling is your life right now?
(Scale 1-5)                                                                            62 non-null     int64
 6   To what extent do you doubt the news you read?
 7   To what extent do you doubt mainstream media, which has been criticized for misinformation?
 8   In your opinion, how many hours does the average person spend on social media daily? (Enter a number)
 9   Considering how addictive social media has become, how many hours do you think the average person wastes on social media daily? (En
 10  Which statement do you agree with more?
 11  How urgent do you think climate change topic is?
(Scale 1-5)                                                                           62 non-null     int64
 12  Which statement do you agree with more?
 13  How reliable do you think scientific research is?
(Scale 1-5)                                                                          62 non-null     int64
dtypes: int64(6), object(8)
memory usage: 6.9+ KB
None
```

```
# Checks for missing (null) values in each column and returns the count per column

df.isnull().sum()
```

| | 0 |
|---|---|
| **Timestamp** | 0 |
| **Do you think the wording of a question can influence how people respond?** | 0 |
| **Which of the following factors do you think has the most influence on survey responses? (Select up to 2)** | 0 |
| **Have you ever felt that a survey question was designed to push a specific answer?** | 0 |
| **How satisfied are you with your life in general? \n(Scale 1 - 5)** | 0 |
| **How happy and fulfilling is your life right now? \n(Scale 1-5)** | 0 |
| **To what extent do you doubt the news you read?** | 0 |
| **To what extent do you doubt mainstream media, which has been criticized for misinformation?** | 0 |
| **In your opinion, how many hours does the average person spend on social media daily? (Enter a number)** | 0 |
| **Considering how addictive social media has become, how many hours do you think the average person wastes on social media daily? (Enter a number)** | 0 |
| **Which statement do you agree with more?** | 0 |
| **How urgent do you think climate change topic is? \n(Scale 1-5)** | 0 |
| **Which statement do you agree with more?** | 0 |
| **How reliable do you think scientific research is? \n(Scale 1-5)** | 0 |

```
# Iterates over DataFrame columns (starting from the second column)
# Prints the index, column name, and unique values of each column
for i in range(1, len(df.columns)):
    print(i, " - ", df.columns[i], df[df.columns[i]].unique())
```

```
1  -  Do you think the wording of a question can influence how people respond?  ['Yes, significantly' 'Somewhat' 'No, not at all']
2  -  Which of the following factors do you think has the most influence on survey responses? (Select up to 2) ['The wording of the ques
 'The order of questions in the survey;The number of answer choices'
 'The time and place the survey is conducted'
 'The number of answer choices;The time and place the survey is conducted'
 'The wording of the question;The time and place the survey is conducted'
 'The wording of the question;The order of questions in the survey;The number of answer choices;The time and place the survey is conduct
 'The wording of the question;The order of questions in the survey;The time and place the survey is conducted'
 'The number of answer choices' 'The order of questions in the survey'
 'The wording of the question;The number of answer choices;The time and place the survey is conducted'
 'The wording of the question;The order of questions in the survey'
 'The wording of the question'
 'The order of questions in the survey;The time and place the survey is conducted']
3  -  Have you ever felt that a survey question was designed to push a specific answer? ['Yes, frequently' 'Sometimes' 'No, never']
4  -  How satisfied are you with your life in general?
(Scale 1 - 5) [3 4 1 2 5]
5  -  How happy and fulfilling is your life right now?
(Scale 1-5)   [2 4 1 3 5]
6  -  To what extent do you doubt the news you read? [3 2 1 5 4]
7  -  To what extent do you doubt mainstream media, which has been criticized for misinformation?   [2 3 4 1 5]
8  -  In your opinion, how many hours does the average person spend on social media daily? (Enter a number)   ['3' '4' '5' '8' '.' '2' '
9  -  Considering how addictive social media has become, how many hours do you think the average person wastes on social media daily? (E
10  -  Which statement do you agree with more?   ['The climate crisis is a disaster, and governments must act immediately!'
 'Climate change is a global issue that requires attention.']
11  -  How urgent do you think climate change topic is?
(Scale 1-5)   [4 3 2 5 1]
12  -  Which statement do you agree with more?  ['How reliable is scientific research?'
 'How much do you trust scientific studies, given that many are influenced by corporate funding?']
13  -  How reliable do you think scientific research is?
(Scale 1-5)   [4 1 3 2 5]
```

```
df["Do you think the wording of a question can influence how people respond? "].unique()
#Valid categorical Values , no problem in it
```

```
array(['Yes, significantly', 'Somewhat', 'No, not at all'], dtype=object)
```

```
df["Which of the following factors do you think has the most influence on survey responses? (Select up to 2)"].unique()
#Valid categorical Values , but in some choices they're merged in one large sentence for example in "The wording of the question;The order c
# so let's seperate it
```

```
array(['The wording of the question;The number of answer choices',
       'The order of questions in the survey;The number of answer choices',
       'The time and place the survey is conducted',
       'The number of answer choices;The time and place the survey is conducted',
       'The wording of the question;The time and place the survey is conducted',
       'The wording of the question;The order of questions in the survey;The number of answer choices;The time and place the survey is
conducted',
       'The wording of the question;The order of questions in the survey;The time and place the survey is conducted',
       'The number of answer choices',
       'The order of questions in the survey',
       'The wording of the question;The number of answer choices;The time and place the survey is conducted',
       'The wording of the question;The order of questions in the survey',
       'The wording of the question',
       'The order of questions in the survey;The time and place the survey is conducted'],
      dtype=object)
```

```
# Column name variable
column_name = "Which of the following factors do you think has the most influence on survey responses? (Select up to 2)"

# Define the main factors
main_factors = {
    "The wording of the question",
    "The order of questions in the survey",
    "The number of answer choices",
    "The time and place the survey is conducted"
}

# Split and explode the column
df[column_name] = df[column_name].str.split(";")  # Split on semicolon
df_exploded = df.explode(column_name)  # Separate multiple-choice responses into rows

# Keep only the main factors
df_exploded = df_exploded[df_exploded[column_name].isin(main_factors)]

# Count occurrences
factor_counts = df_exploded[column_name].value_counts()
```

```
df["Have you ever felt that a survey question was designed to push a specific answer?"].unique()
#Valid categorical Values , no problem in it
```

```
array(['Yes, frequently', 'Sometimes', 'No, never'], dtype=object)
```

```
df['How satisfied are you with your life in general? \n(Scale 1 - 5)'].unique()
#Numerical Values , no problem in it
```

```
array([3, 4, 1, 2, 5])
```

```
df['How happy and fulfilling is your life right now? \n(Scale 1-5)  '].unique()
#Numerical Values , no problem in it
```

```
array([2, 4, 1, 3, 5])
```

```
df['To what extent do you doubt the news you read?'].unique()
#Numerical Values , no problem in it
```

```
array([3, 2, 1, 5, 4])
```

```
df['To what extent do you doubt mainstream media, which has been criticized for misinformation?  '].unique()
#Numerical Values , no problem in it
```

```
array([2, 3, 4, 1, 5])
```

```
df['In your opinion, how many hours does the average person spend on social media daily? (Enter a number)  '].unique()

# it has been took from the from as string so we need to change the valid values into Integars
# there are some inputs we can't change like '.' and '4-8'
```

```
array(['3', '4', '5', '8', '.', '2', '6', '3 ', '4-8', '5 ', '3-6', '2h',
       '4-7 hours '], dtype=object)
```

```
new_column = []
for value in df['In your opinion, how many hours does the average person spend on social media daily? (Enter a number)  ']:
    try:
        value = str(value)  # Convert to string to avoid TypeError
        if '2h' in value:
            new_column.append(2)  # Replace '2h' with 2
        else:
            new_column.append(int(float(value)))  # Convert valid numbers to int
    except ValueError:
        new_column.append(None)  # Handle invalid inputs

df['In your opinion, how many hours does the average person spend on social media daily? (Enter a number)  '] = new_column

# we changed the invalid input with null till we fill it with the mean value of the column.
```

```
df['In your opinion, how many hours does the average person spend on social media daily? (Enter a number)  '].isnull().sum()
```

```
np.int64(5)
```

```
#fill the empty cells with the value of the mean
mean = df['In your opinion, how many hours does the average person spend on social media daily? (Enter a number)  '].mean()
df['In your opinion, how many hours does the average person spend on social media daily? (Enter a number)  '].fillna(mean, inplace=True)
```

```
<ipython-input-21-bf19a1572ade>:3: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignm
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting valu

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].me


  df['In your opinion, how many hours does the average person spend on social media daily? (Enter a number)  '].fillna(mean, inplace=Tru
```

```
df['In your opinion, how many hours does the average person spend on social media daily? (Enter a number)  '].isnull().sum()
```

```
np.int64(0)
```

```
df['Considering how addictive social media has become, how many hours do you think the average person wastes on social media daily? (Enter a
```

```
array(['2', '4', '6', '7', '9', '5', '.', '3', '8', '4-5', '16', '1-5',
       '3h', '1-2', ' 6'], dtype=object)
```

```
new_column = []
for value in df['Considering how addictive social media has become, how many hours do you think the average person wastes on social media da
  try:
    if '2h' in str(value): # Convert value to string to handle potential non-string values
      new_column.append(3) # to validate '3h' input.
    else:
      new_column.append(int(value))
  except ValueError:
    new_column.append(None)

df['Considering how addictive social media has become, how many hours do you think the average person wastes on social media daily? (Enter a

# we changed the invalid input with null till we fill it later with the mean value of the column.
```

```
df['Considering how addictive social media has become, how many hours do you think the average person wastes on social media daily? (Enter a
```

```
np.int64(6)
```

```
#fill the empty cells with the value of the mean
mean = df['Considering how addictive social media has become, how many hours do you think the average person wastes on social media daily? (
df['Considering how addictive social media has become, how many hours do you think the average person wastes on social media daily? (Enter a
```

```
<ipython-input-26-da99adf5a025>:3: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignm
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting valu

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].me


  df['Considering how addictive social media has become, how many hours do you think the average person wastes on social media daily? (E
```

```
df['Considering how addictive social media has become, how many hours do you think the average person wastes on social media daily? (Enter a
```

```
np.int64(0)
```

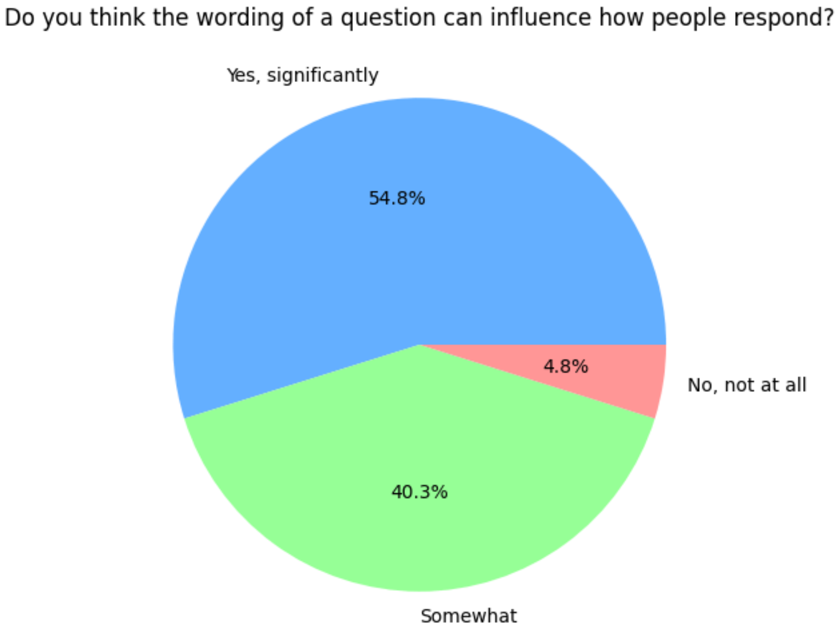
Now, we've cleaned our data. Let's start our analysis

**First: The Categorical data.**

```
df["Do you think the wording of a question can influence how people respond? "].value_counts()
```

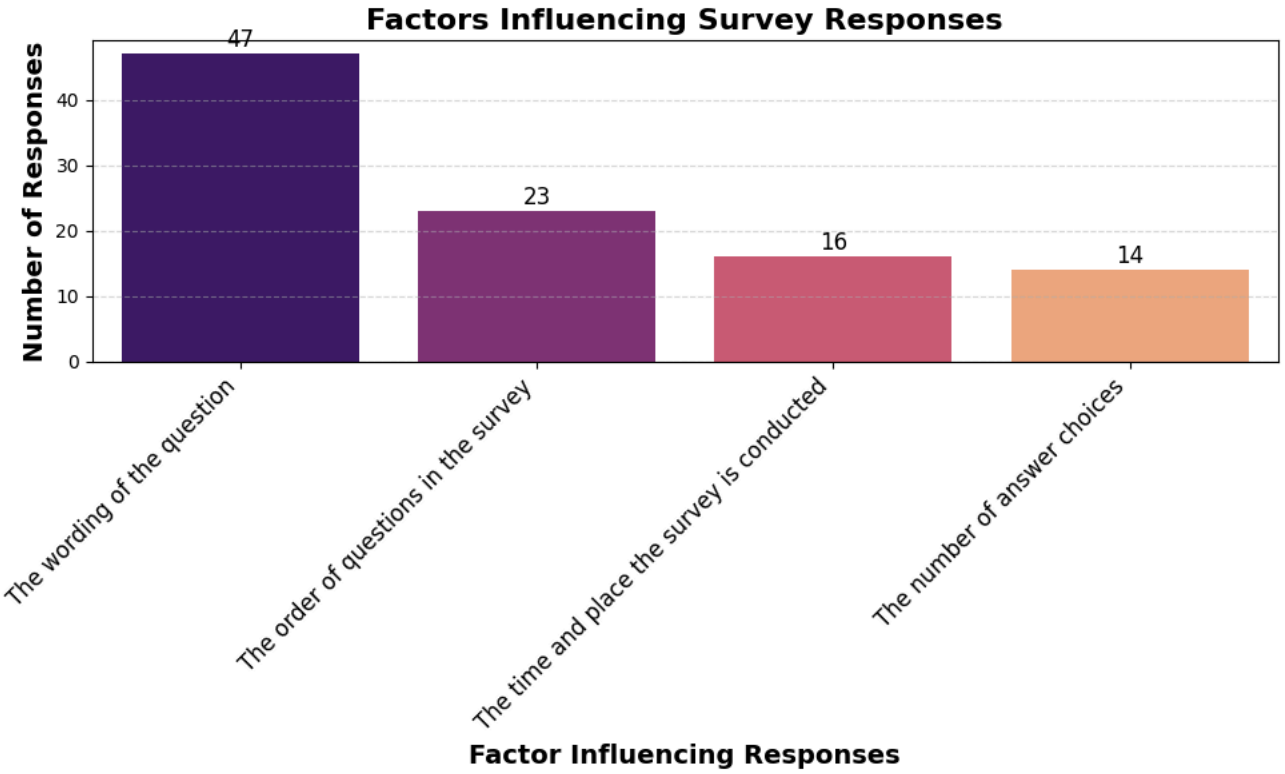
```
                                                                              count
Do you think the wording of a question can influence how people respond?
                          Yes, significantly                                     34
                               Somewhat                                          25
                            No, not at all                                        3

dtype: int64
```

```
values = df["Do you think the wording of a question can influence how people respond? "].value_counts()
labels = values.index  # Get unique response labels

# Create a pie chart
plt.figure(figsize=(6, 6))
plt.pie(values, labels=labels, autopct="%1.1f%%", colors=["#66b3ff", "#99ff99", "#ff9999"])
plt.title("Do you think the wording of a question can influence how people respond?")
plt.show()
```



```
# Create a vertical bar chart
plt.figure(figsize=(10, 6))
ax = sns.barplot(x=factor_counts.index, y=factor_counts.values, palette="magma")

# Rotate x-axis labels for readability
plt.xticks(rotation=45, ha="right", fontsize=12)

# Add labels above each bar
for i, v in enumerate(factor_counts.values):
    ax.text(i, v + 0.2, str(v), ha='center', va='bottom', fontsize=12)

# Styling
plt.xlabel("Factor Influencing Responses", fontsize=14, fontweight="bold")
plt.ylabel("Number of Responses", fontsize=14, fontweight="bold")
plt.title("Factors Influencing Survey Responses", fontsize=16, fontweight="bold")
plt.grid(axis="y", linestyle="--", alpha=0.5)
plt.ylim(0, max(factor_counts.values) + 2)  # Add space above bars

# Adjust layout to prevent clipping
plt.tight_layout()

plt.show()
```

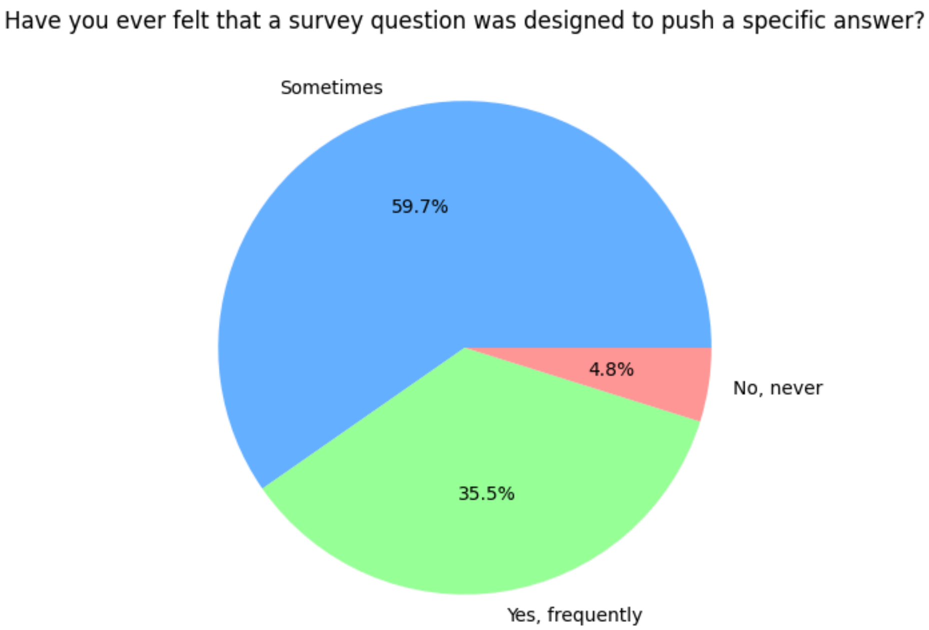
```
<ipython-input-30-8f19da972be1>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend

  ax = sns.barplot(x=factor_counts.index, y=factor_counts.values, palette="magma")
```



```
# Frequency table
df["Have you ever felt that a survey question was designed to push a specific answer?"].value_counts()
```

```
                                                                          count
 Have you ever felt that a survey question was designed to push a specific answer?
                              Sometimes                                      37
                           Yes, frequently                                   22
                              No, never                                       3

dtype: int64
```

```
# Extract values and labels directly
values = df["Have you ever felt that a survey question was designed to push a specific answer?"].value_counts()
labels = values.index  # Get unique response labels

# Create a pie chart
plt.figure(figsize=(6, 6))
plt.pie(values, labels=labels, autopct="%1.1f%%", colors=["#66b3ff", "#99ff99", "#ff9999"])
plt.title("Have you ever felt that a survey question was designed to push a specific answer?")
plt.show()
```



**Second: The Quantitative data.**

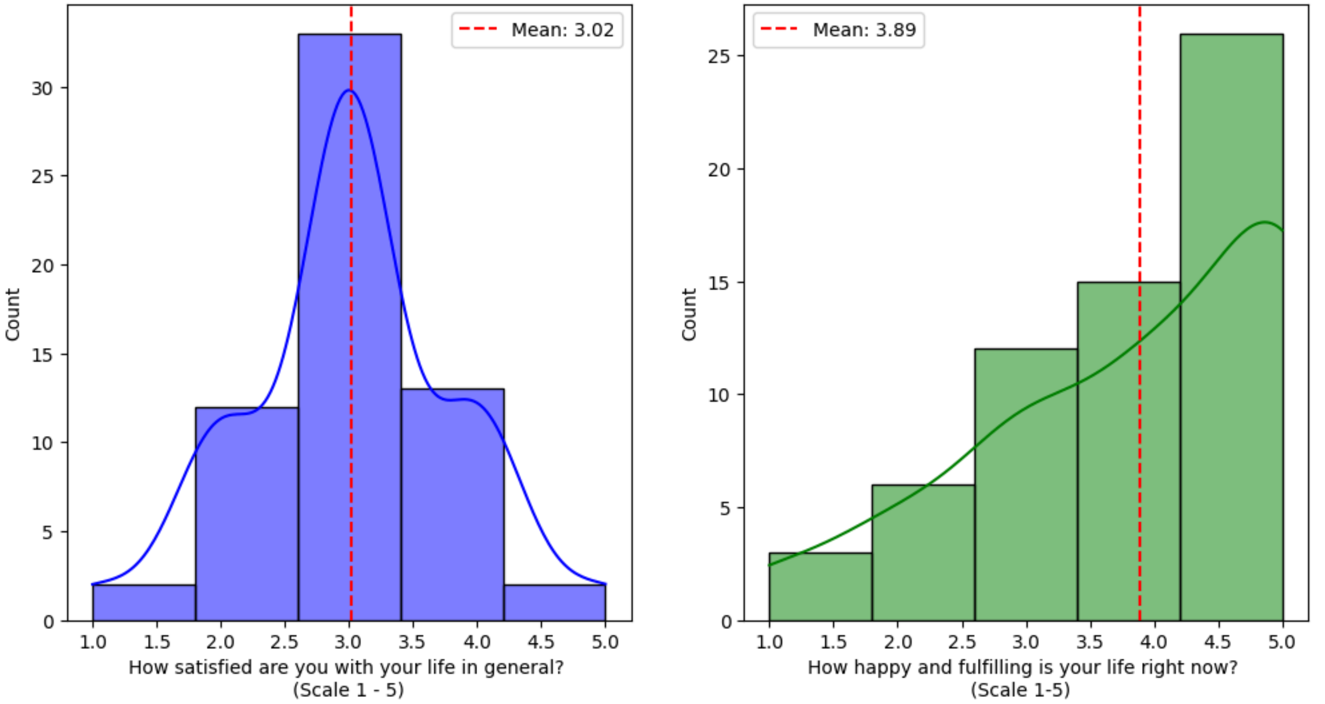
**How satisfied are you with your life in general?** vs **How happy and fulfilling is your life right now?**

```
# Calculate means
q_normal1 = df["How satisfied are you with your life in general? \n(Scale 1 - 5)"]
q_wordy1 = df["How happy and fulfilling is your life right now? \n(Scale 1-5)  "]
# Frequency Tables
freq_normal1 = q_normal1.value_counts().sort_index()
freq_wordy1 = q_wordy1.value_counts().sort_index()

print("First Table - Normal Question 1:\n", freq_normal1)
print(f"Mean: {q_normal1.mean():.2f}, Median: {q_normal1.median()}, Mode: {q_normal1.mode()[0]}")


print("\n\nSecond Table - Wordy Question 1:\n", freq_wordy1)
print(f"Mean: {q_wordy1.mean():.2f}, Median: {q_wordy1.median()}, Mode: {q_wordy1.mode()[0]}")
```

```
First Table - Normal Question 1:
 How satisfied are you with your life in general? \n(Scale 1 - 5)
1     2
2    12
3    33
4    13
5     2
Name: count, dtype: int64
Mean: 3.02, Median: 3.0, Mode: 3


Second Table - Wordy Question 1:
 How happy and fulfilling is your life right now? \n(Scale 1-5)
1     3
2     6
3    12
4    15
5    26
```

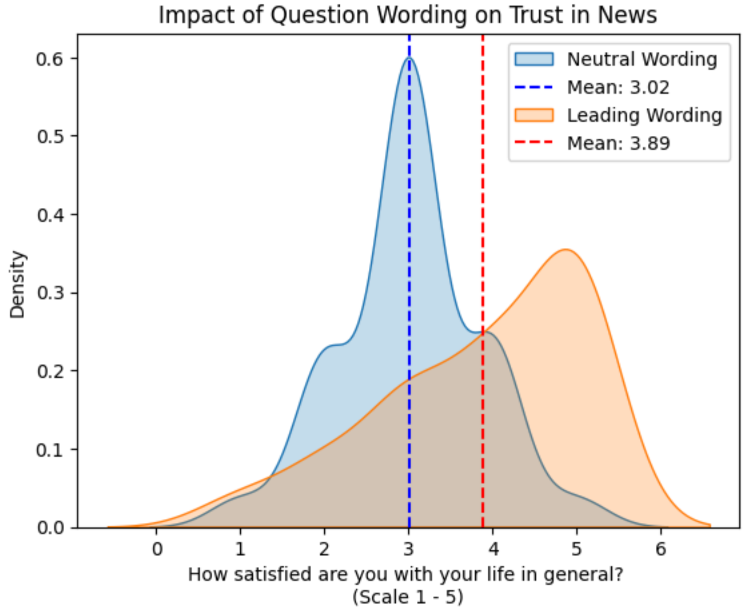
```
        Name: count, dtype: int64
        Mean: 3.89, Median: 4.0, Mode: 5
```

```
plt.figure(figsize=(12, 6))

plt.subplot(1, 2, 1)
sns.histplot(q_normal1, bins=5, kde=True, color='blue')
plt.axvline(q_normal1.mean(), color='red', linestyle='dashed', label=f'Mean: {q_normal1.mean():.2f}')
plt.legend()

plt.subplot(1, 2, 2)
sns.histplot(q_wordy1, bins=5, kde=True, color='green')
plt.axvline(q_wordy1.mean(), color='red', linestyle='dashed', label=f'Mean: {q_wordy1.mean():.2f}')
plt.legend()

plt.show()
```

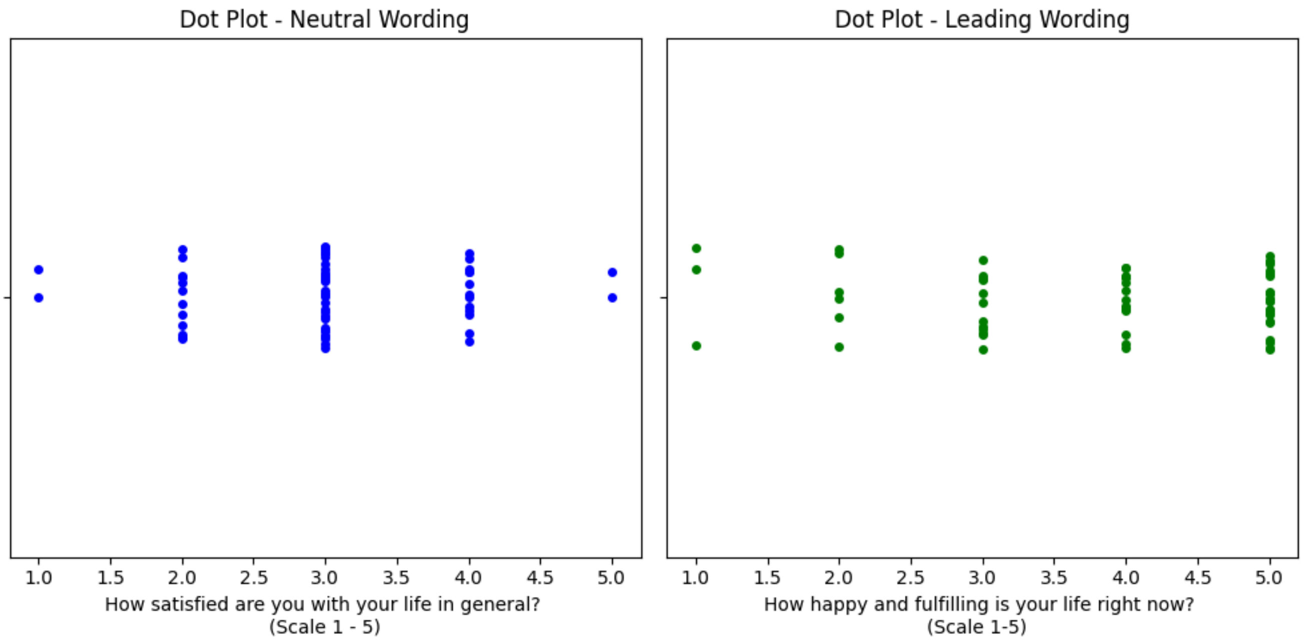


```
# Plot distributions of neutral vs. leading questions
sns.kdeplot(q_normal1, label="Neutral Wording", shade=True)
plt.axvline(q_normal1.mean(), color='blue', linestyle='dashed', label=f'Mean: {q_normal1.mean():.2f}')

sns.kdeplot(q_wordy1, label="Leading Wording", shade=True)
plt.axvline(q_wordy1.mean(), color='red', linestyle='dashed', label=f'Mean: {q_wordy1.mean():.2f}')


plt.title("Impact of Question Wording on Trust in News")
plt.legend()
plt.show()
```

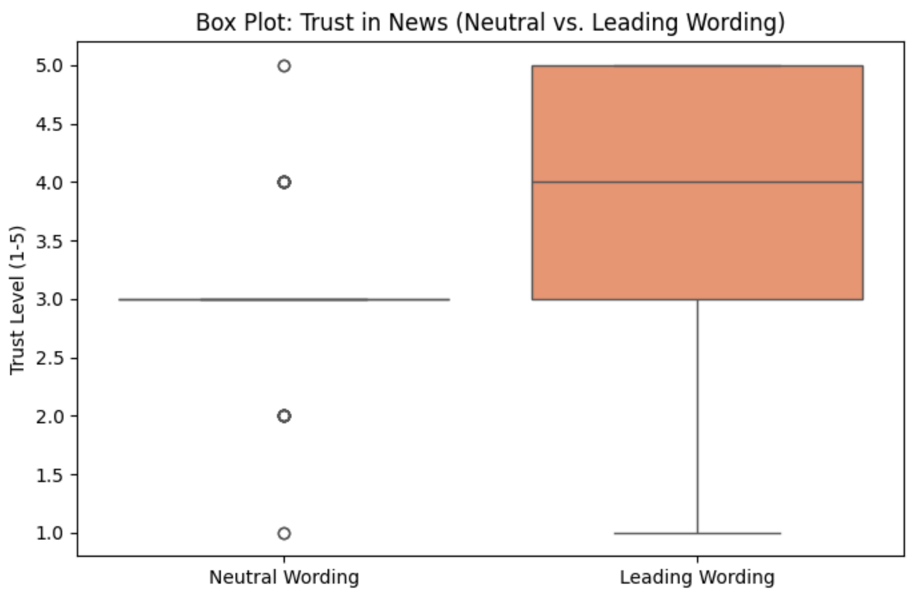
```
<ipython-input-35-2b79bc19483c>:2: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  sns.kdeplot(q_normal1, label="Neutral Wording", shade=True)
<ipython-input-35-2b79bc19483c>:5: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  sns.kdeplot(q_wordy1, label="Leading Wording", shade=True)
```



```
plt.figure(figsize=(10, 5))  # Increased width for better visibility

plt.subplot(1, 2, 1)
sns.stripplot(x = q_normal1, color='blue')
plt.title("Dot Plot - Neutral Wording")

plt.subplot(1, 2, 2)
sns.stripplot(x = q_wordy1, color='green')
plt.title("Dot Plot - Leading Wording")

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



```
# Box plot comparing neutral vs. leading wording
plt.figure(figsize=(8, 5))
sns.boxplot(data=(q_normal1, q_wordy1), palette="Set2")
plt.title("Box Plot: Trust in News (Neutral vs. Leading Wording)")
plt.ylabel("Trust Level (1-5)")
plt.xticks([0, 1], ["Neutral Wording", "Leading Wording"])
plt.show()
```



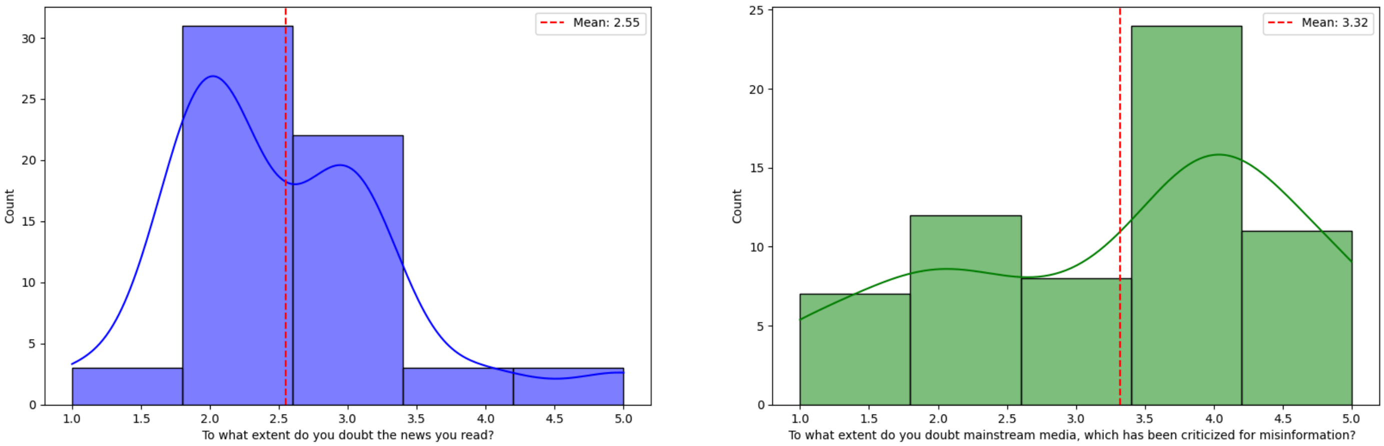
**To what extent do you doubt the news you read?** vs **To what extent do you doubt mainstream media, which has been criticized for misinformation?**

```
# Calculate means
q_normal2 = df["To what extent do you doubt the news you read?"]
q_wordy2 = df["To what extent do you doubt mainstream media, which has been criticized for misinformation?  "]
# Frequency Tables
freq_normal2 = q_normal2.value_counts().sort_index()
freq_wordy2 = q_wordy2.value_counts().sort_index()

print("First Table - Normal Question 2:\n", freq_normal2)
print(f"Mean: {q_normal2.mean():.2f}, Median: {q_normal2.median()}, Mode: {q_normal2.mode()[0]}")


print("\n\nSecond Table - Wordy Question 2:\n", freq_wordy2)
print(f"Mean: {q_wordy2.mean():.2f}, Median: {q_wordy2.median()}, Mode: {q_wordy2.mode()[0]}")
```

```
First Table - Normal Question 2:
 To what extent do you doubt the news you read?
1     3
2    31
3    22
4     3
5     3
Name: count, dtype: int64
Mean: 2.55, Median: 2.0, Mode: 2


Second Table - Wordy Question 2:
 To what extent do you doubt mainstream media, which has been criticized for misinformation?
1     7
2    12
3     8
4    24
5    11
Name: count, dtype: int64
Mean: 3.32, Median: 4.0, Mode: 4
```

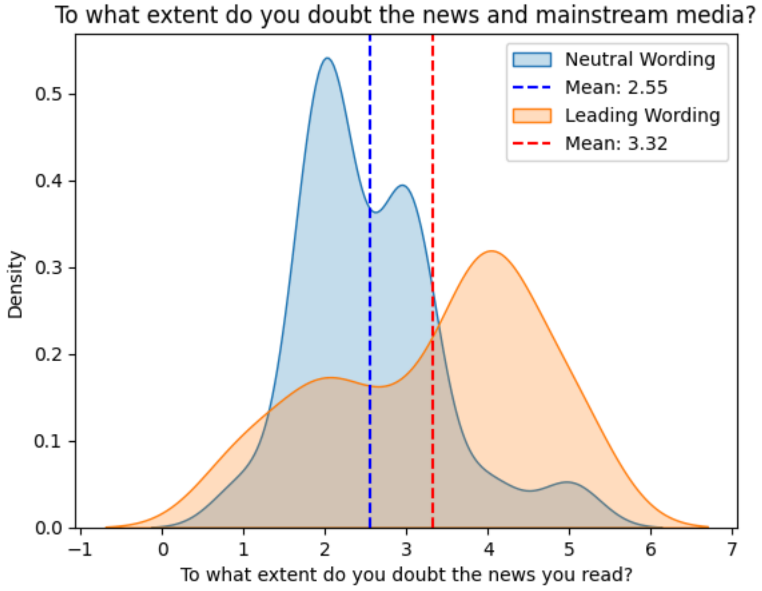
```
plt.figure(figsize=(20, 6))

plt.subplot(1, 2, 1)
```

```
sns.histplot(q_normal2, bins=5, kde=True, color='blue')
plt.axvline(q_normal2.mean(), color='red', linestyle='dashed', label=f'Mean: {q_normal2.mean():.2f}')
plt.legend()

plt.subplot(1, 2, 2)
sns.histplot(q_wordy2, bins=5, kde=True, color='green')
plt.axvline(q_wordy2.mean(), color='red', linestyle='dashed', label=f'Mean: {q_wordy2.mean():.2f}')
plt.legend()

plt.show()
```

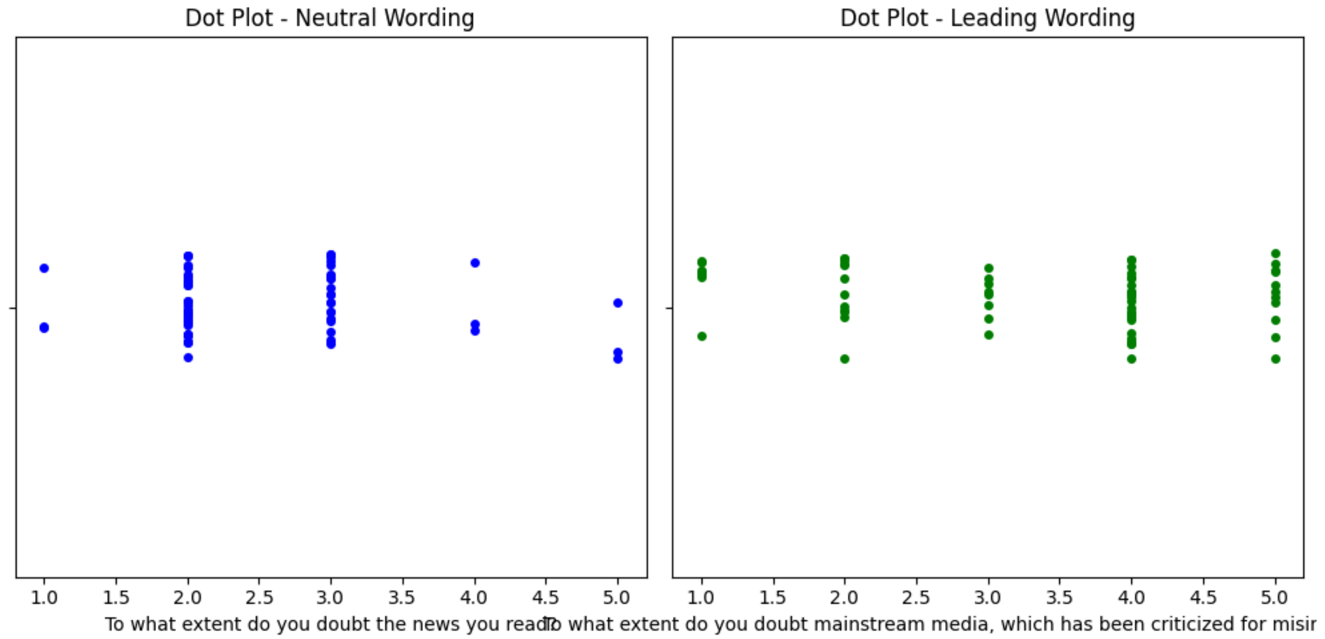


```
# Plot distributions of neutral vs. leading questions
sns.kdeplot(q_normal2, label="Neutral Wording", shade=True)
plt.axvline(q_normal2.mean(), color='blue', linestyle='dashed', label=f'Mean: {q_normal2.mean():.2f}')

sns.kdeplot(q_wordy2, label="Leading Wording", shade=True)
plt.axvline(q_wordy2.mean(), color='red', linestyle='dashed', label=f'Mean: {q_wordy2.mean():.2f}')


plt.title("To what extent do you doubt the news and mainstream media?")
plt.legend()
plt.show()
```

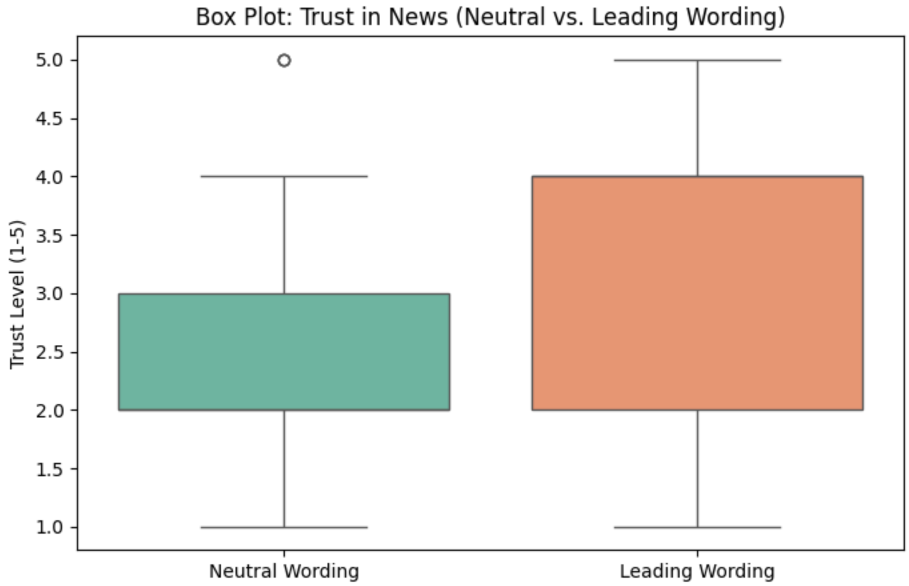
```
<ipython-input-40-386b1c1cd2ea>:2: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  sns.kdeplot(q_normal2, label="Neutral Wording", shade=True)
<ipython-input-40-386b1c1cd2ea>:5: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  sns.kdeplot(q_wordy2, label="Leading Wording", shade=True)
```



```
plt.figure(figsize=(10, 5))  # Increased width for better visibility

plt.subplot(1, 2, 1)
sns.stripplot(x = q_normal2, color='blue')
plt.title("Dot Plot - Neutral Wording")

plt.subplot(1, 2, 2)
sns.stripplot(x = q_wordy2, color='green')
plt.title("Dot Plot - Leading Wording")

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



```
# Box plot comparing neutral vs. leading wording
plt.figure(figsize=(8, 5))
sns.boxplot(data=df[["To what extent do you doubt the news you read?", "To what extent do you doubt mainstream media, which has been critici
plt.title("Box Plot: Trust in News (Neutral vs. Leading Wording)")
plt.ylabel("Trust Level (1-5)")
plt.xticks([0, 1], ["Neutral Wording", "Leading Wording"])
plt.show()
```



**In your opinion, how many hours does the average person spend on social media daily?**

vs

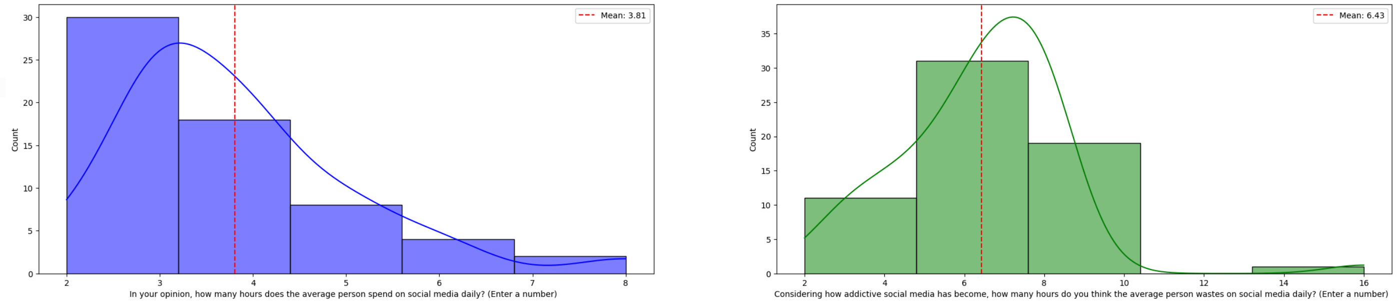
**Considering how addictive social media has become, how many hours do you think the average person wastes on social media daily?**

```
# Calculate means
q_normal3 = df["In your opinion, how many hours does the average person spend on social media daily? (Enter a number)  "]
q_wordy3 = df["Considering how addictive social media has become, how many hours do you think the average person wastes on social media dail
# Frequency Tables
freq_normal3 = q_normal3.value_counts().sort_index()
freq_wordy3 = q_wordy3.value_counts().sort_index()

print("First Table - Normal Question 3:\n", freq_normal3)
print(f"Mean: {q_normal3.mean():.2f}, Median: {q_normal3.median():.2f}, Mode: {q_normal3.mode()[0]}")


print("\n\nSecond Table - Wordy Question 3:\n", freq_wordy3)
print(f"Mean: {q_wordy3.mean():.2f}, Median: {q_wordy3.median():.2f}, Mode: {q_wordy3.mode()[0]}")
```

```
First Table - Normal Question 3:
 In your opinion, how many hours does the average person spend on social media daily? (Enter a number)
2.000000     5
3.000000    25
3.807018     5
4.000000    13
5.000000     8
6.000000     4
8.000000     2
Name: count, dtype: int64
Mean: 3.81, Median: 3.81, Mode: 3.0


Second Table - Wordy Question 3:
 Considering how addictive social media has become, how many hours do you think the average person wastes on social media daily? (Enter
2.000000      1
3.000000      5
4.000000      5
5.000000      7
6.000000      7
6.428571      6
```

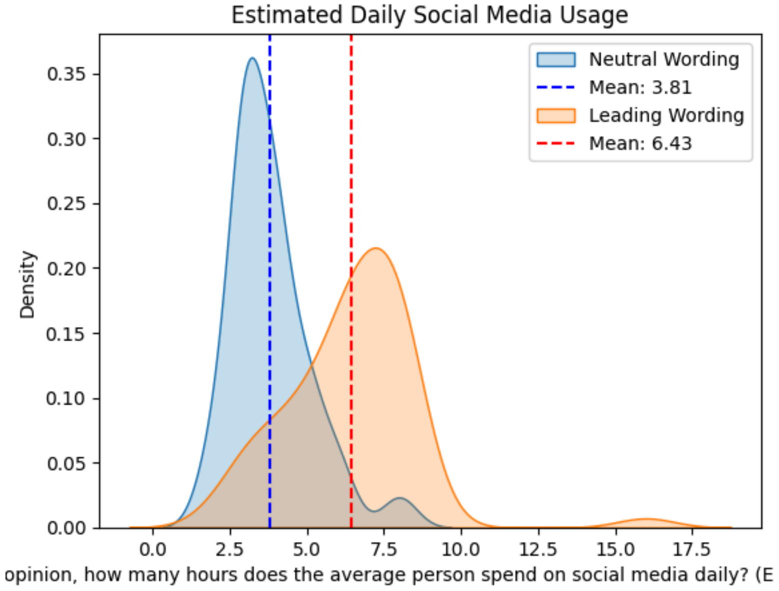
```
7.000000     11
8.000000     18
9.000000      1
16.000000     1
Name: count, dtype: int64
Mean: 6.43, Median: 6.71, Mode: 8.0
```

```
plt.figure(figsize=(30, 6))

plt.subplot(1, 2, 1)
sns.histplot(q_normal3, bins=5, kde=True, color='blue')
plt.axvline(q_normal3.mean(), color='red', linestyle='dashed', label=f'Mean: {q_normal3.mean():.2f}')
plt.legend()

plt.subplot(1, 2, 2)
sns.histplot(q_wordy3, bins=5, kde=True, color='green')
plt.axvline(q_wordy3.mean(), color='red', linestyle='dashed', label=f'Mean: {q_wordy3.mean():.2f}')
plt.legend()

plt.show()
```

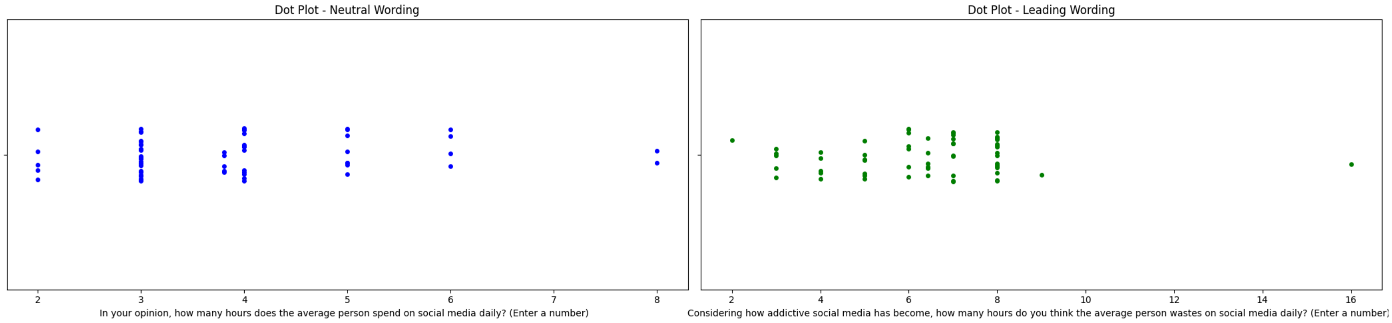


```
# Plot distributions of neutral vs. leading questions
sns.kdeplot(q_normal3, label="Neutral Wording", shade=True)
plt.axvline(q_normal3.mean(), color='blue', linestyle='dashed', label=f'Mean: {q_normal3.mean():.2f}')

sns.kdeplot(q_wordy3, label="Leading Wording", shade=True)
plt.axvline(q_wordy3.mean(), color='red', linestyle='dashed', label=f'Mean: {q_wordy3.mean():.2f}')


plt.title("Estimated Daily Social Media Usage")
plt.legend()
plt.show()
```

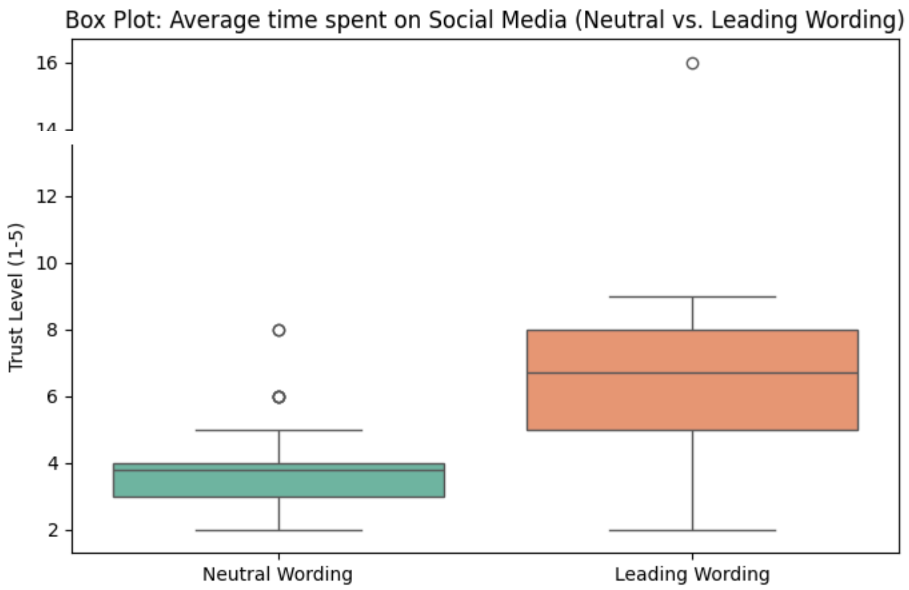
```
<ipython-input-45-4118ca365681>:2: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  sns.kdeplot(q_normal3, label="Neutral Wording", shade=True)
<ipython-input-45-4118ca365681>:5: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  sns.kdeplot(q_wordy3, label="Leading Wording", shade=True)
```



```
plt.figure(figsize=(21, 5))  # Increased width for better visibility

plt.subplot(1, 2, 1)
sns.stripplot(x = q_normal3, color='blue')
plt.title("Dot Plot - Neutral Wording")

plt.subplot(1, 2, 2)
sns.stripplot(x = q_wordy3, color='green')
plt.title("Dot Plot - Leading Wording")

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



```
# Box plot comparing neutral vs. leading wording
plt.figure(figsize=(8, 5))
sns.boxplot(data=df[["In your opinion, how many hours does the average person spend on social media daily? (Enter a number)  ", "Considering
plt.title("Box Plot: Average time spent on Social Media (Neutral vs. Leading Wording)")
plt.ylabel("Trust Level (1-5)")
plt.xticks([0, 1], ["Neutral Wording", "Leading Wording"])
plt.show()
```



**Climate change is a global issue that requires attention.**

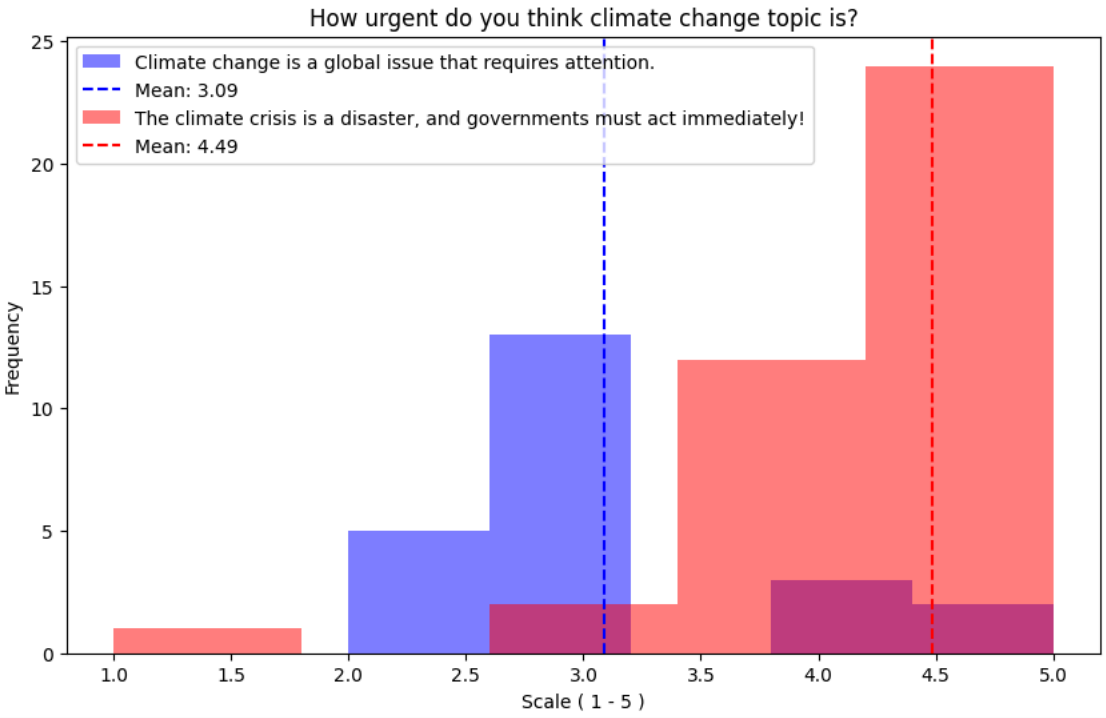
vs

**The climate crisis is a disaster, and governments must act immediately!**

```
# Define the statements
statement_1_q1 = "Climate change is a global issue that requires attention."
statement_2_q1 = "The climate crisis is a disaster, and governments must act immediately!"

# Filter the data
group1_question1 = df[df["Which statement do you agree with more?  "] == statement_1_q1]  # People who chose the first statement
group2_question1 = df[df["Which statement do you agree with more?  "] == statement_2_q1]  # People who chose the second statement



# Calculate mean, median, and mode for Group 1
group1_mean_question1 = group1_question1["How urgent do you think climate change topic is?  \n(Scale 1-5)  "].mean()
group1_median_question1 = group1_question1["How urgent do you think climate change topic is?  \n(Scale 1-5)  "].median()
group1_mode_question1 = group1_question1["How urgent do you think climate change topic is?  \n(Scale 1-5)  "].mode().values

# Calculate mean, median, and mode for Group 2
group2_mean_question1 = group2_question1["How urgent do you think climate change topic is?  \n(Scale 1-5)  "].mean()
group2_median_question1 = group2_question1["How urgent do you think climate change topic is?  \n(Scale 1-5)  "].median()
group2_mode_question1 = group2_question1["How urgent do you think climate change topic is?  \n(Scale 1-5)  "].mode().values

# Print the results
print("Group 1 (Climate change is a global issue that requires attention.) - Mean:", group1_mean_question1)
print("Group 2 (The climate crisis is a disaster, and governments must act immediately!) - Mean:", group2_mean_question1)

print("Group 1 (Climate change is a global issue that requires attention.) - Median:", group1_median_question1)
print("Group 2 (The climate crisis is a disaster, and governments must act immediately!) - Median:", group2_median_question1)

print("Group 1 (Climate change is a global issue that requires attention.) - Mode:", group1_mode_question1)
print("Group 2 (The climate crisis is a disaster, and governments must act immediately!) - Mode:", group2_mode_question1)
```

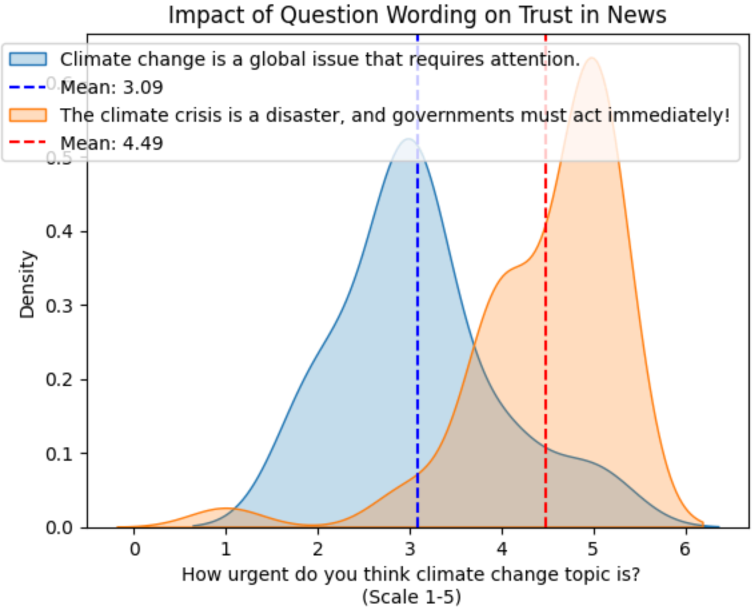
```
Group 1 (Climate change is a global issue that requires attention.) - Mean: 3.0869565217391304
Group 2 (The climate crisis is a disaster, and governments must act immediately!) - Mean: 4.487179487179487
Group 1 (Climate change is a global issue that requires attention.) - Median: 3.0
Group 2 (The climate crisis is a disaster, and governments must act immediately!) - Median: 5.0
Group 1 (Climate change is a global issue that requires attention.) - Mode: [3]
Group 2 (The climate crisis is a disaster, and governments must act immediately!) - Mode: [5]
```

```
urgency_col = "How urgent do you think climate change topic is?  \n(Scale 1-5)  "

plt.figure(figsize=(10, 6))
plt.hist(group1_question1[urgency_col], bins=5, alpha=0.5, label= statement_1_q1, color="blue")
plt.axvline(group1_question1[urgency_col].mean(), color='blue', linestyle='dashed', label=f'Mean: {group1_mean_question1:.2f}')
plt.hist(group2_question1[urgency_col], bins=5, alpha=0.5, label= statement_2_q1 , color="red")
plt.axvline(group2_question1[urgency_col].mean(), color='red', linestyle='dashed', label=f'Mean: {group2_mean_question1:.2f}')

plt.ylabel("Frequency")
plt.xlabel("Scale ( 1 - 5 )")
plt.title("How urgent do you think climate change topic is?")
plt.legend()
plt.show()
```



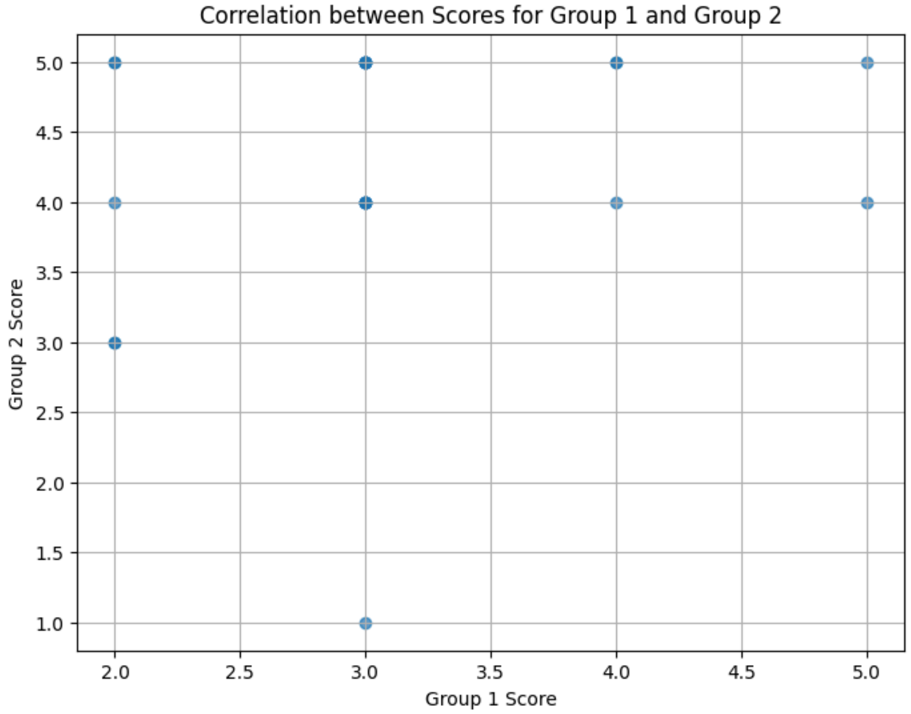
```
# Plot distributions of neutral vs. leading questions
sns.kdeplot(group1_question1[urgency_col], label= statement_1_q1, shade=True)
plt.axvline(group1_mean_question1, color='blue', linestyle='dashed', label=f'Mean: {group1_mean_question1:.2f}')

sns.kdeplot(group2_question1[urgency_col], label= statement_2_q1, shade=True)
plt.axvline(group2_mean_question1, color='red', linestyle='dashed', label=f'Mean: {group2_mean_question1:.2f}')


plt.title("Impact of Question Wording on Trust in News")
plt.legend()
plt.show()
```

```
<ipython-input-51-32848384032f>:2: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  sns.kdeplot(group1_question1[urgency_col], label= statement_1_q1, shade=True)
<ipython-input-51-32848384032f>:5: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  sns.kdeplot(group2_question1[urgency_col], label= statement_2_q1, shade=True)
```



```
min_size = len(group1_question1)
group1_q1 = group1_question1["How urgent do you think climate change topic is?  \n(Scale 1-5)  "][:min_size]
group2_q1 = group2_question1["How urgent do you think climate change topic is?  \n(Scale 1-5)  "][:min_size]

# Calculate Pearson correlation
correlation, p_value = stats.pearsonr(group1_q1, group2_q1)

print(f"Correlation coefficient: {correlation:.2f}")
print(f"P-value: {p_value:.3f}")

# Visualize the relationship  a scatter plot
plt.figure(figsize=(8, 6))
plt.scatter(group1_q1, group2_q1, alpha=0.7)
plt.xlabel("Group 1 Score")
plt.ylabel("Group 2 Score")
plt.title("Correlation between Scores for Group 1 and Group 2")
plt.grid()
plt.show()
```

```
Correlation coefficient: 0.19
P-value: 0.377
```



**How reliable is scientific research?**

vs

**How much do you trust scientific studies, given that many are influenced by corporate funding?**

```
# Define the statements
statement_1_q2 = "How reliable is scientific research?"
statement_2_q2 = "How much do you trust scientific studies, given that many are influenced by corporate funding?"

# Filter the data
group1_question2 = df[df["Which statement do you agree with more? "] == statement_1_q2]  # People who chose the first statement
group2_question2 = df[df["Which statement do you agree with more? "] == statement_2_q2]  # People who chose the second statement



# Calculate mean, median, and mode for Group 1
group1_mean_question2 = group1_question2["How reliable do you think scientific research is?  \n(Scale 1-5)  "].mean()
group1_median_question2 = group1_question2["How reliable do you think scientific research is?  \n(Scale 1-5)  "].median()
group1_mode_question2 = group1_question2["How reliable do you think scientific research is?  \n(Scale 1-5)  "].mode().values

# Calculate mean, median, and mode for Group 2
group2_mean_question2 = group2_question2["How reliable do you think scientific research is?  \n(Scale 1-5)  "].mean()
group2_median_question2 = group2_question2["How reliable do you think scientific research is?  \n(Scale 1-5)  "].median()
group2_mode_question2 = group2_question2["How reliable do you think scientific research is?  \n(Scale 1-5)  "].mode().values

# Print the results
print("Group 1 (How reliable is scientific research?) - Mean:", group1_mean_question2)
print("Group 2 (How much do you trust scientific studies, given that many are influenced by corporate funding?) - Mean:", group2_mean_questi

print("Group 1 (How reliable is scientific research?) - Median:", group1_median_question2)
print("Group 2 (How much do you trust scientific studies, given that many are influenced by corporate funding?) - Median:", group2_median_qu

print("Group 1 (How reliable is scientific research?) - Mode:", group1_mode_question2)
print("Group 2 (How much do you trust scientific studies, given that many are influenced by corporate funding?) - Mode:", group2_mode_questi
```

```
Group 1 (How reliable is scientific research?) - Mean: 3.588235294117647
Group 2 (How much do you trust scientific studies, given that many are influenced by corporate funding?) - Mean: 3.7333333333333334
Group 1 (How reliable is scientific research?) - Median: 4.0
Group 2 (How much do you trust scientific studies, given that many are influenced by corporate funding?) - Median: 4.0
Group 1 (How reliable is scientific research?) - Mode: [3]
Group 2 (How much do you trust scientific studies, given that many are influenced by corporate funding?) - Mode: [4]
```

```
urgency_col = "How reliable do you think scientific research is?  \n(Scale 1-5)  "

plt.figure(figsize=(10, 6))
plt.hist(group1_question2[urgency_col], bins=5, alpha=0.5, label= statement_1_q2, color="blue")
plt.axvline(group1_question2[urgency_col].mean(), color='blue', linestyle='dashed', label=f'Mean: {group1_mean_question2:.2f}')
plt.hist(group2_question2[urgency_col], bins=5, alpha=0.5, label= statement_2_q2 , color="red")
```