```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
df=pd.read_csv("weather-check.csv")
df
```

```
     RespondentID Check_Weather_Daily  \
0      3887201482                 Yes
1      3887159451                 Yes
2      3887152228                 Yes
3      3887145426                 Yes
4      3887021873                 Yes
..            ...                 ...
914    3877568315                 Yes
915    3877568116                  No
916    3877568054                 Yes
917    3877568053                 Yes
918    3877566926                 Yes

                                    Check_Weather_Method  \
0                The default weather app on your phone
1                The default weather app on your phone
2                The default weather app on your phone
3                The default weather app on your phone
4    A specific website or app (please provide the ...
..                                                 ...
914                                The Weather Channel
915                                    Internet search
916                                The Weather Channel
917              The default weather app on your phone
918                                      Local TV News

    Weather_Smartwatch_Likelihood      Age  Gender
Household_Income  \
0                      Very likely  30 - 44    Male    $50,000 to
$74,999
1                      Very likely  18 - 29    Male  Prefer not to
answer
2                      Very likely  30 - 44    Male  $100,000 to
$124,999
3                   Somewhat likely  30 - 44    Male  Prefer not to
answer
4                      Very likely  30 - 44    Male  $150,000 to
$174,999
..                             ...      ...     ...                    .
..
914                  Very unlikely  30 - 44  Female    $25,000 to
$49,999
915                  Very unlikely      60+  Female  $100,000 to
```

```
$124,999
916                   Very likely  45 - 59  Female  Prefer not to
answer
917                   Very likely  30 - 44  Female  Prefer not to
answer
918                Somewhat likely      60+  Female  Prefer not to
answer

               US Region
0         South Atlantic
1         South Atlantic
2        Middle Atlantic
3         South Atlantic
4        Middle Atlantic
..                   ...
914       South Atlantic
915  West South Central
916             Pacific
917       South Atlantic
918             Pacific

[919 rows x 8 columns]
```

```
df.isna().sum()
```

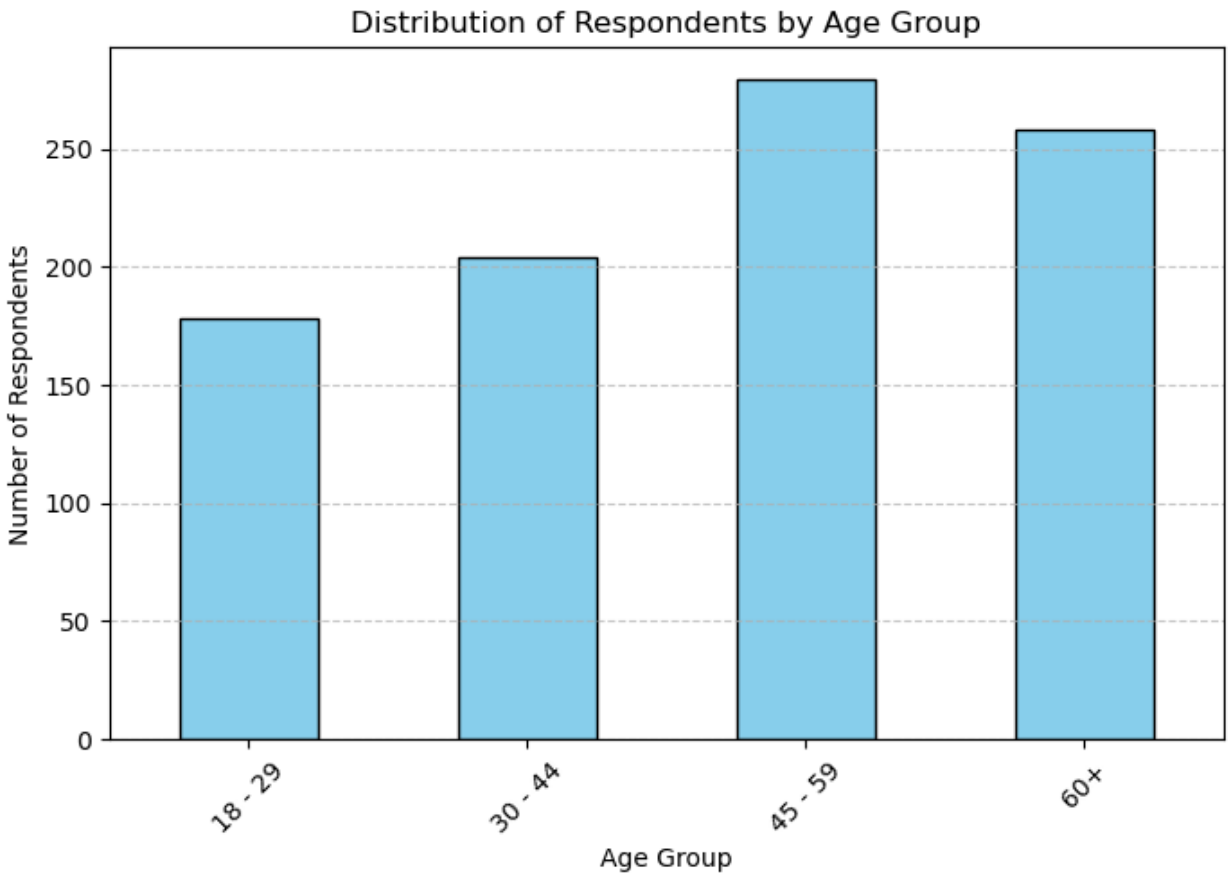
```
RespondentID                      0
Check_Weather_Daily               0
Check_Weather_Method              0
Weather_Smartwatch_Likelihood     0
Age                               0
Gender                            0
Household_Income                  0
US Region                         0
dtype: int64
```

```
df.duplicated().sum()
```

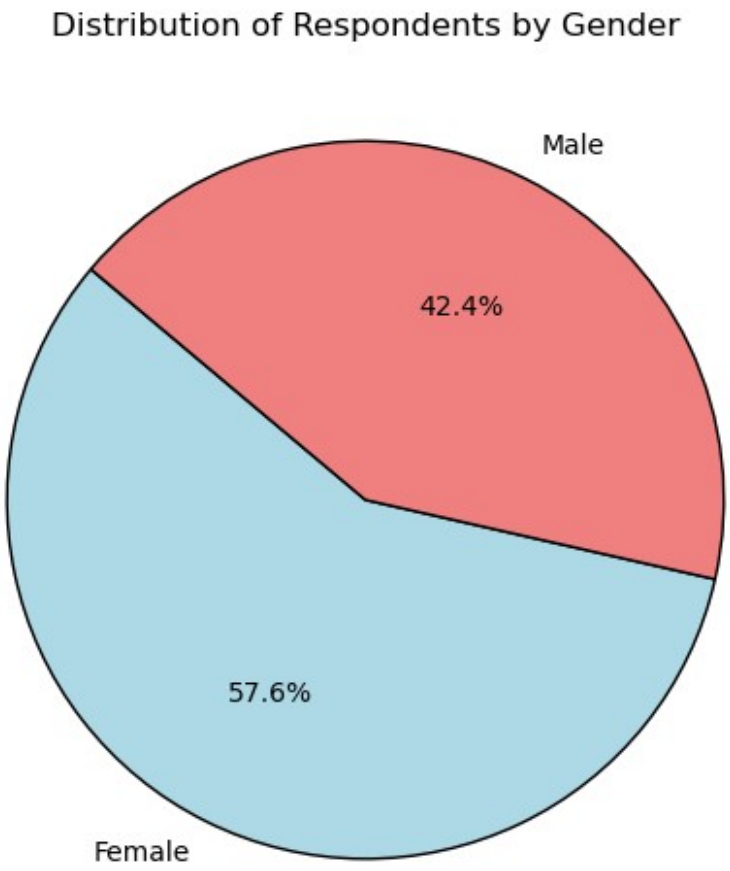
```
0
```

```
#Q1: What is the distribution of respondents by age group? Plot: Bar
chart

age_distribution = df["Age"].value_counts().sort_index()
plt.figure(figsize=(8, 5))
age_distribution.plot(kind="bar", color="skyblue", edgecolor="black")

plt.xlabel("Age Group")
plt.ylabel("Number of Respondents")
plt.title("Distribution of Respondents by Age Group")
plt.xticks(rotation=45)
plt.grid(axis="y", linestyle="--", alpha=0.7)
```

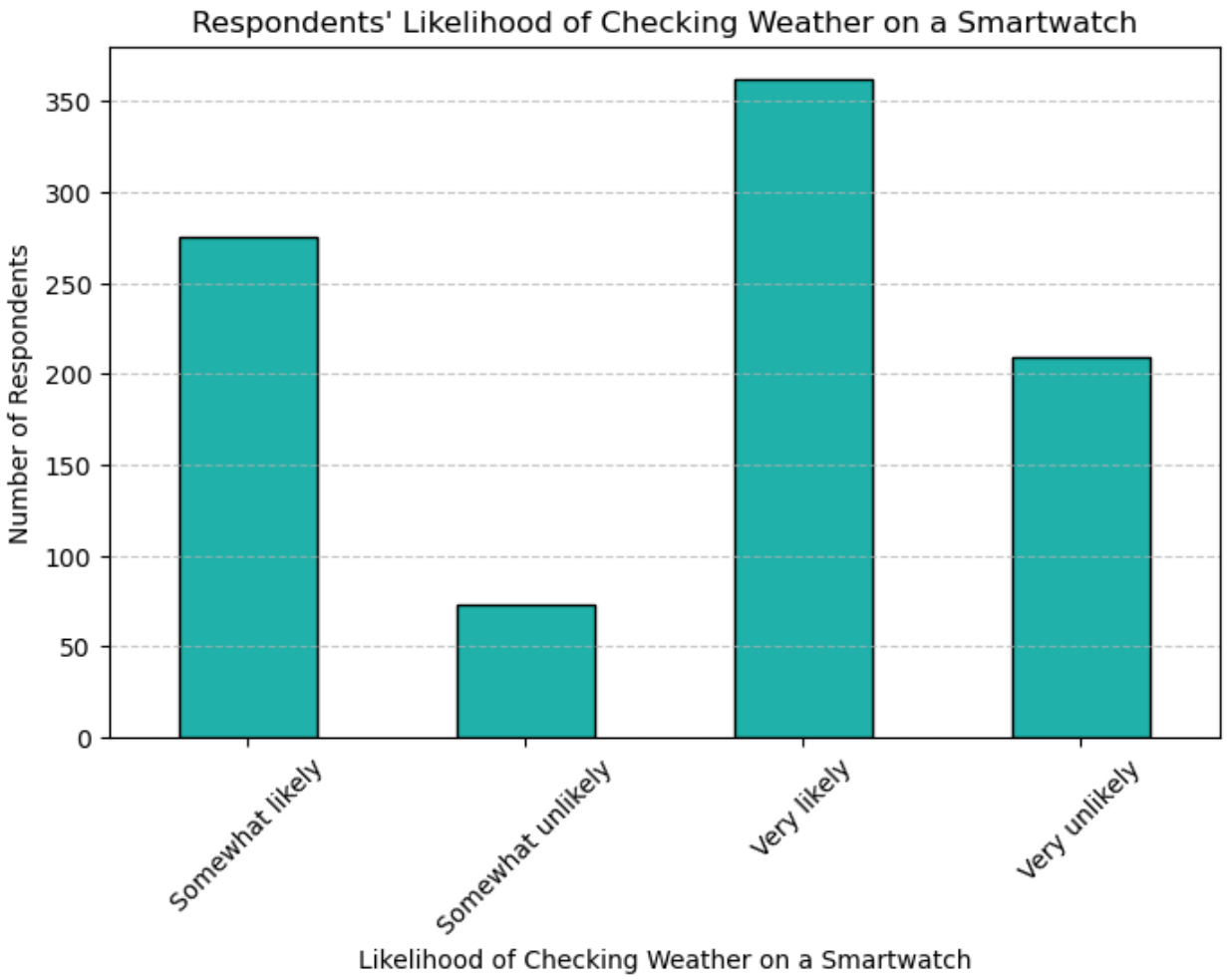
```
plt.show()
```



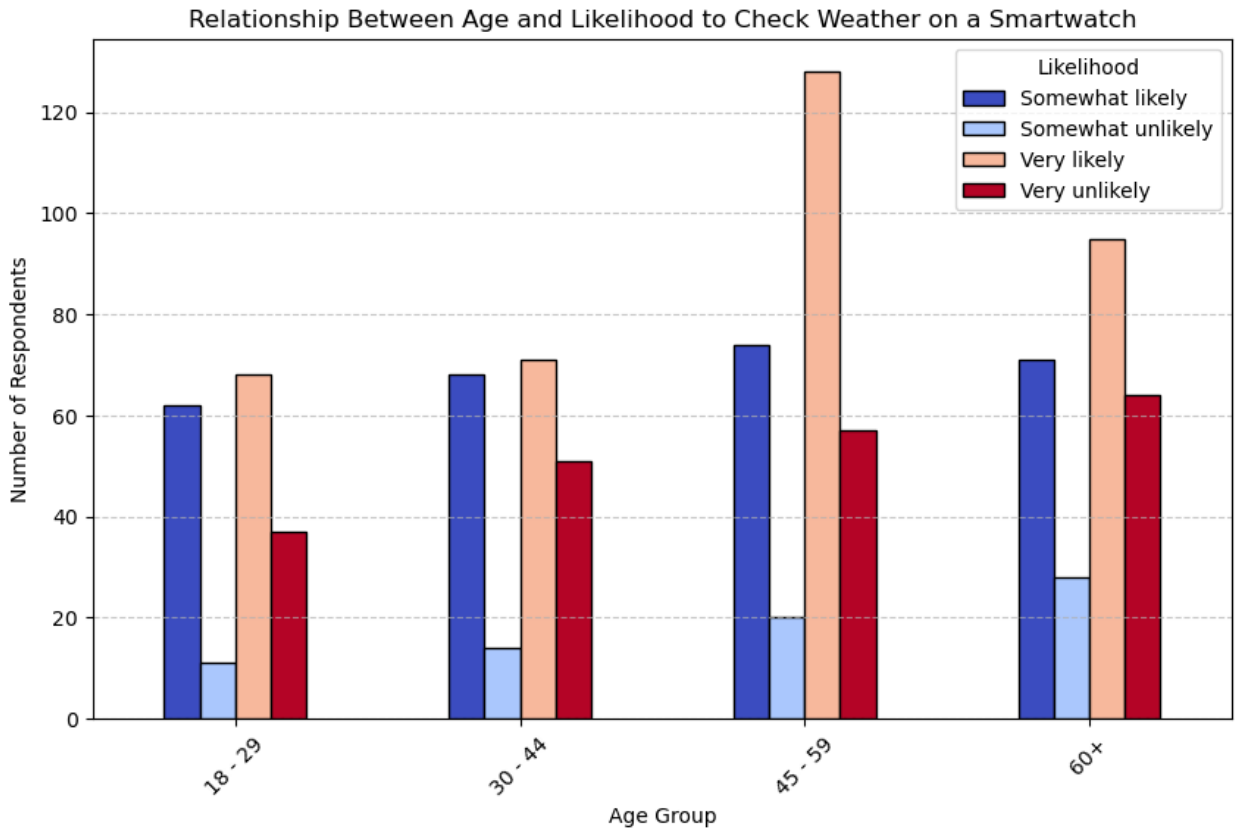

```
#Q2: What is the distridution of respondents by gender? Plot: Pie chart

gender_distribution = df["Gender"].value_counts()

plt.figure(figsize=(6, 6))
plt.pie(
    gender_distribution,
    labels=gender_distribution.index,
    autopct="%1.1f%%",
    colors=["lightblue", "lightcoral", "lightgreen"],
    startangle=140,
    wedgeprops={"edgecolor": "black"}
)

plt.title("Distribution of Respondents by Gender")

plt.show()
```




```
#Q3: How likely are respondents to check the weather on a smartwatch?
Plot: Bar chart

smartwatch_likelihood =
df["Weather_Smartwatch_Likelihood"].value_counts().sort_index()

plt.figure(figsize=(8, 5))
smartwatch_likelihood.plot(kind="bar", color="lightseagreen",
edgecolor="black")

plt.xlabel("Likelihood of Checking Weather on a Smartwatch")
plt.ylabel("Number of Respondents")
plt.title("Respondents' Likelihood of Checking Weather on a
Smartwatch")
plt.xticks(rotation=45)
plt.grid(axis="y", linestyle="--", alpha=0.7)

plt.show()
```



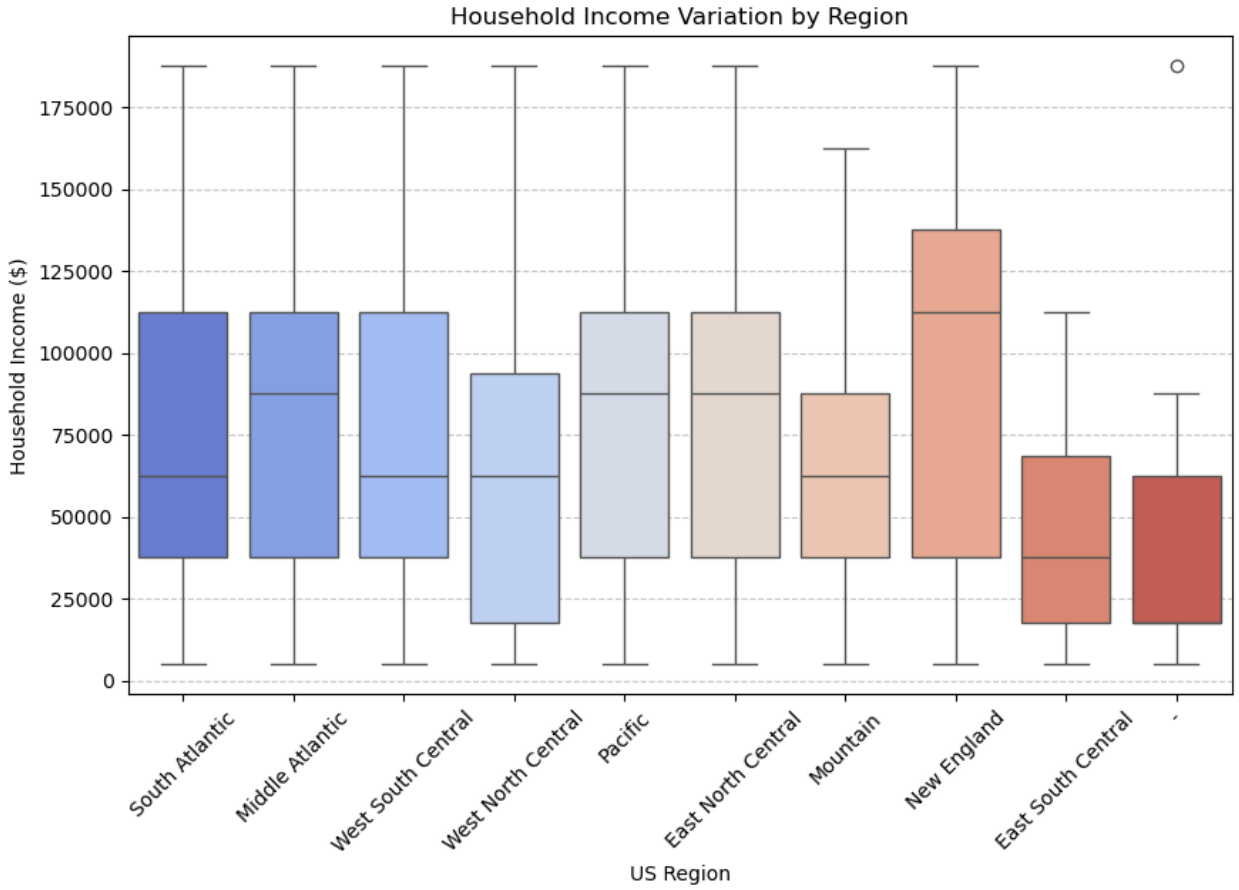

```
#Q4: What is the relationship between age and likelihood to check the
weather on a smartwatch Grouped bar chart.

age_smartwatch_counts = df.groupby(["Age",
"Weather_Smartwatch_Likelihood"]).size().unstack()
age_smartwatch_counts.plot(kind="bar", figsize=(10, 6),
colormap="coolwarm", edgecolor="black")

plt.xlabel("Age Group")
plt.ylabel("Number of Respondents")
plt.title("Relationship Between Age and Likelihood to Check Weather on
a Smartwatch")
plt.xticks(rotation=45)
plt.legend(title="Likelihood")
plt.grid(axis="y", linestyle="--", alpha=0.7)

plt.show()
```



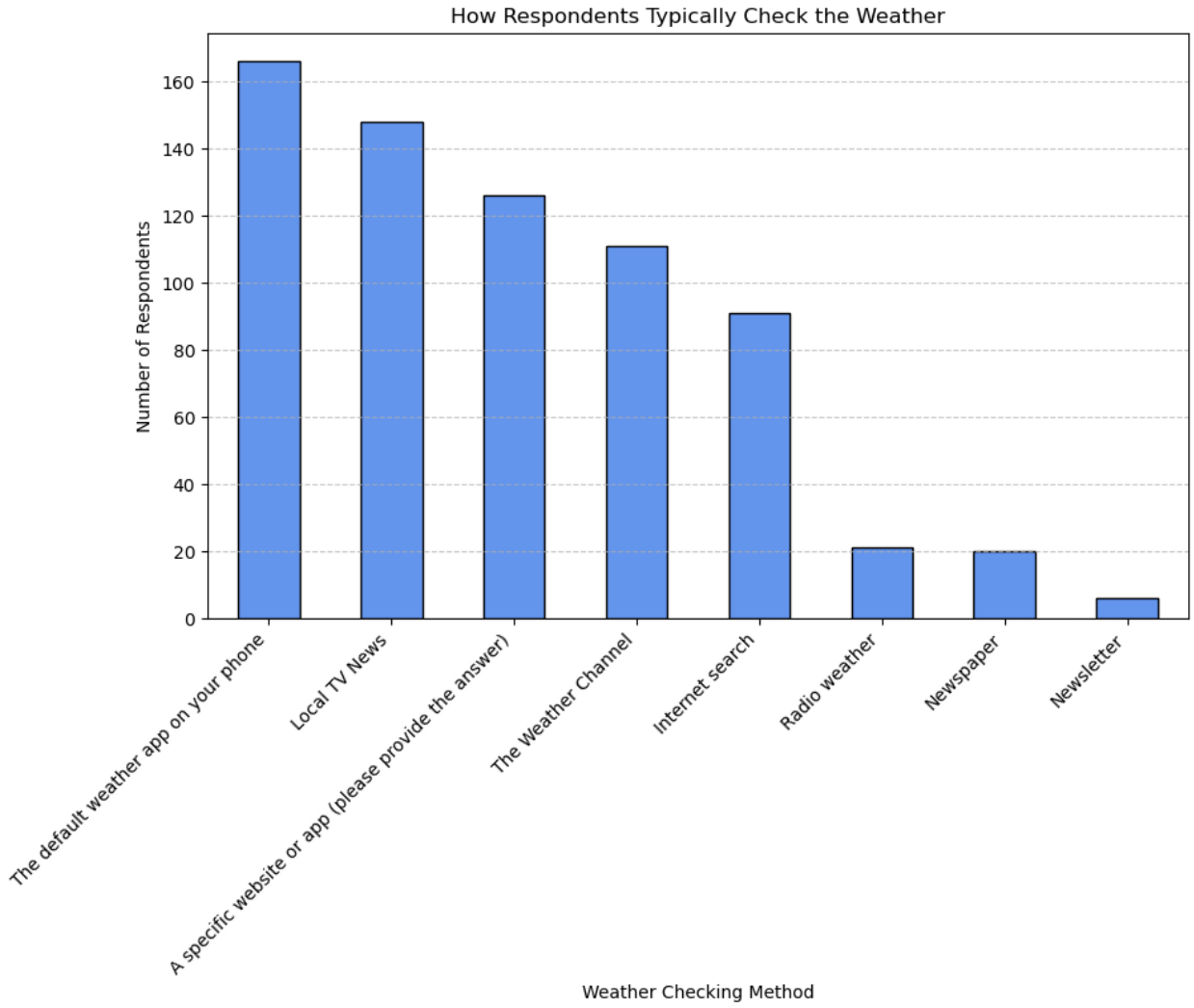

```
#Q5: How does househo;d income vary by the region? Plot: Bax Plot

df = df[df["Household_Income"] != "Prefer not to answer"]

def extract_midpoint(income_range):
    """Convert income range (e.g., "$50,000 to $74,999") to its
midpoint."""
    if isinstance(income_range, str) and " to " in income_range:
        low, high = income_range.replace("$", "").replace(",",
"").split(" to ")
        return (float(low) + float(high)) / 2
    return np.nan

df["Household_Income"] =
df["Household_Income"].apply(extract_midpoint)

df = df.dropna(subset=["Household_Income"])

plt.figure(figsize=(10, 6))
sns.boxplot(x="US Region", y="Household_Income", data=df,
palette="coolwarm")

plt.xlabel("US Region")
plt.ylabel("Household Income ($)")
```

```
plt.title("Household Income Variation by Region")
plt.xticks(rotation=45)
plt.grid(axis="y", linestyle="--", alpha=0.7)

plt.show()
```

```
C:\Users\SANDRA B\AppData\Local\Temp\ipykernel_13832\4208640107.py:12:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df["Household_Income"] =
df["Household_Income"].apply(extract_midpoint)
C:\Users\SANDRA B\AppData\Local\Temp\ipykernel_13832\4208640107.py:17:
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=False` for the same effect.

  sns.boxplot(x="US Region", y="Household_Income", data=df,
palette="coolwarm")
```

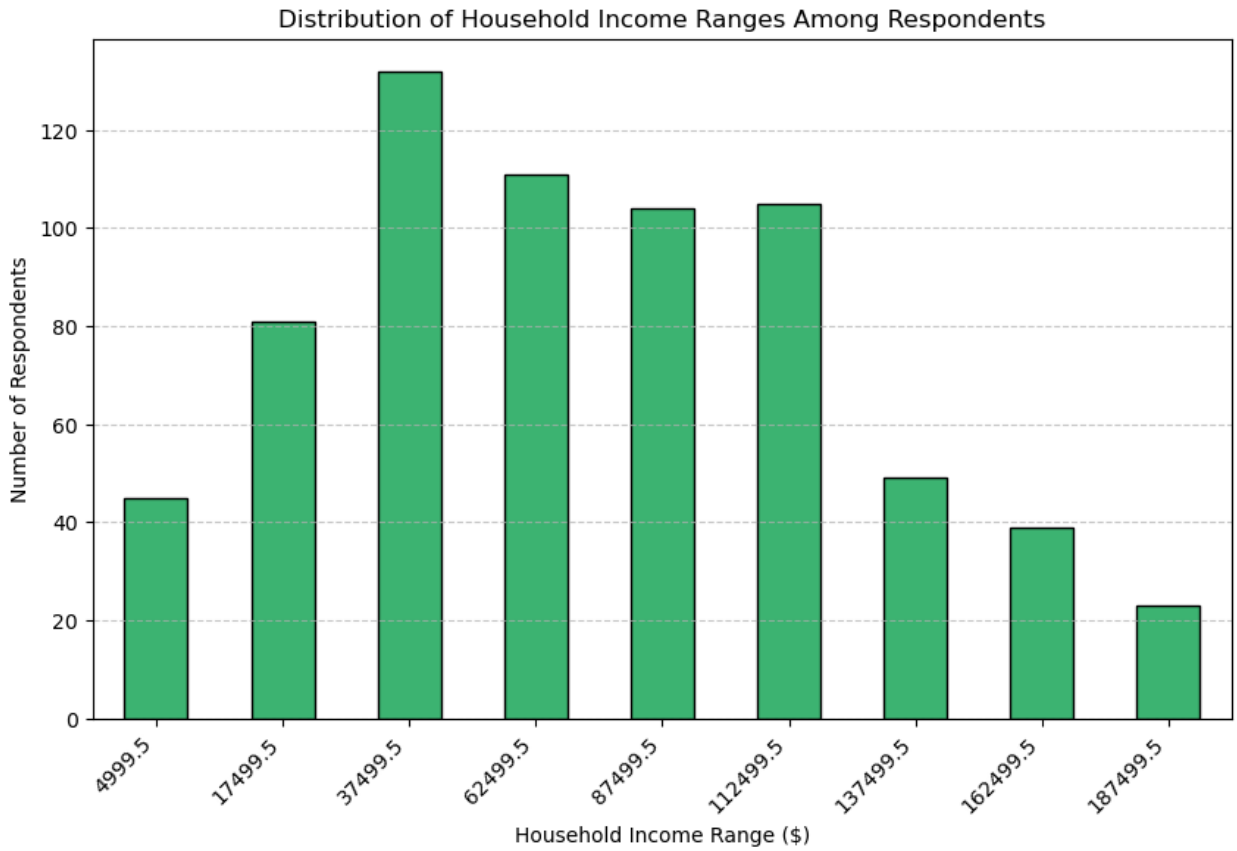



```
#Q6: How do respondents typically check the weather? Plot: Bar Chart.

weather_methods = df["Check_Weather_Method"].value_counts()

plt.figure(figsize=(10, 6))
weather_methods.plot(kind="bar", color="cornflowerblue",
edgecolor="black")

plt.xlabel("Weather Checking Method")
plt.ylabel("Number of Respondents")
plt.title("How Respondents Typically Check the Weather")
plt.xticks(rotation=45, ha="right")
plt.grid(axis="y", linestyle="--", alpha=0.7)

plt.show()
```

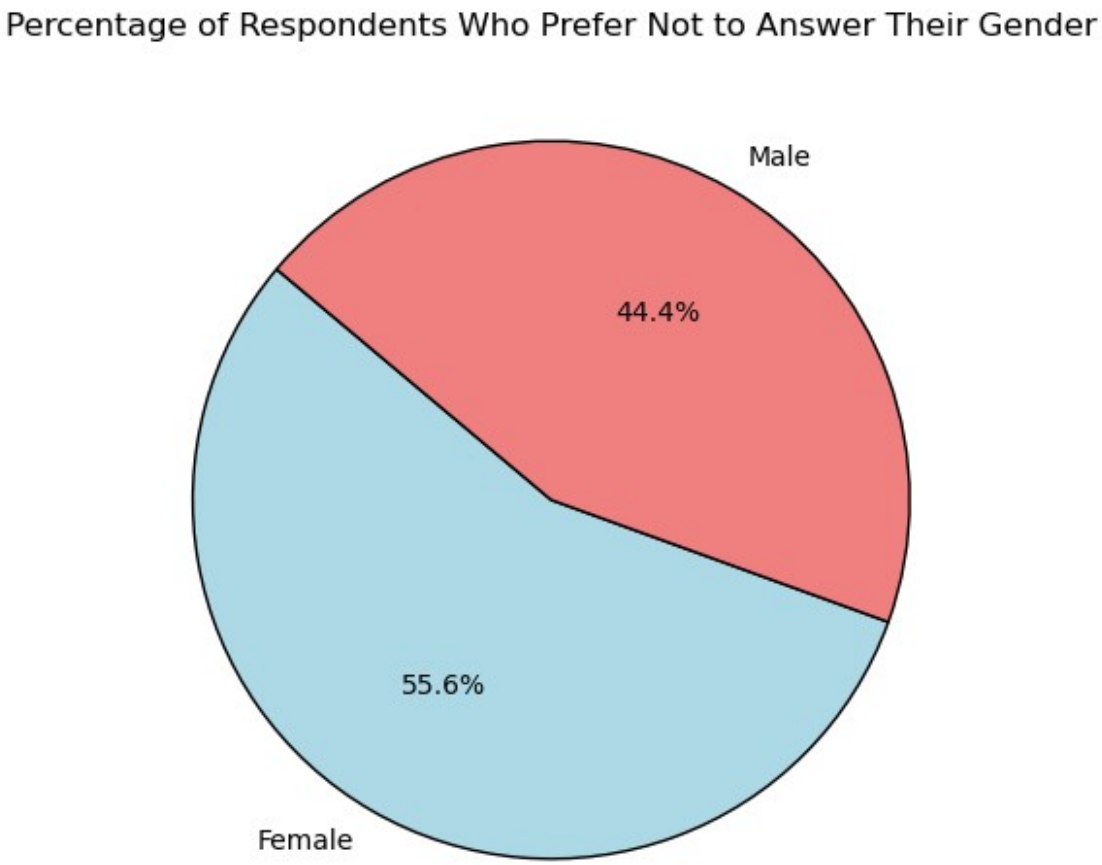



```
#Q7: What is the distribution of household income ranges among
respondents? Plot: Bar Chart.


df = df[df["Household_Income"] != "Prefer not to answer"]

income_distribution =
df["Household_Income"].value_counts().sort_index()

plt.figure(figsize=(10, 6))
income_distribution.plot(kind="bar", color="mediumseagreen",
edgecolor="black")

plt.xlabel("Household Income Range ($)")
plt.ylabel("Number of Respondents")
plt.title("Distribution of Household Income Ranges Among Respondents")
plt.xticks(rotation=45, ha="right")
plt.grid(axis="y", linestyle="--", alpha=0.7)
```

```
plt.show()
```

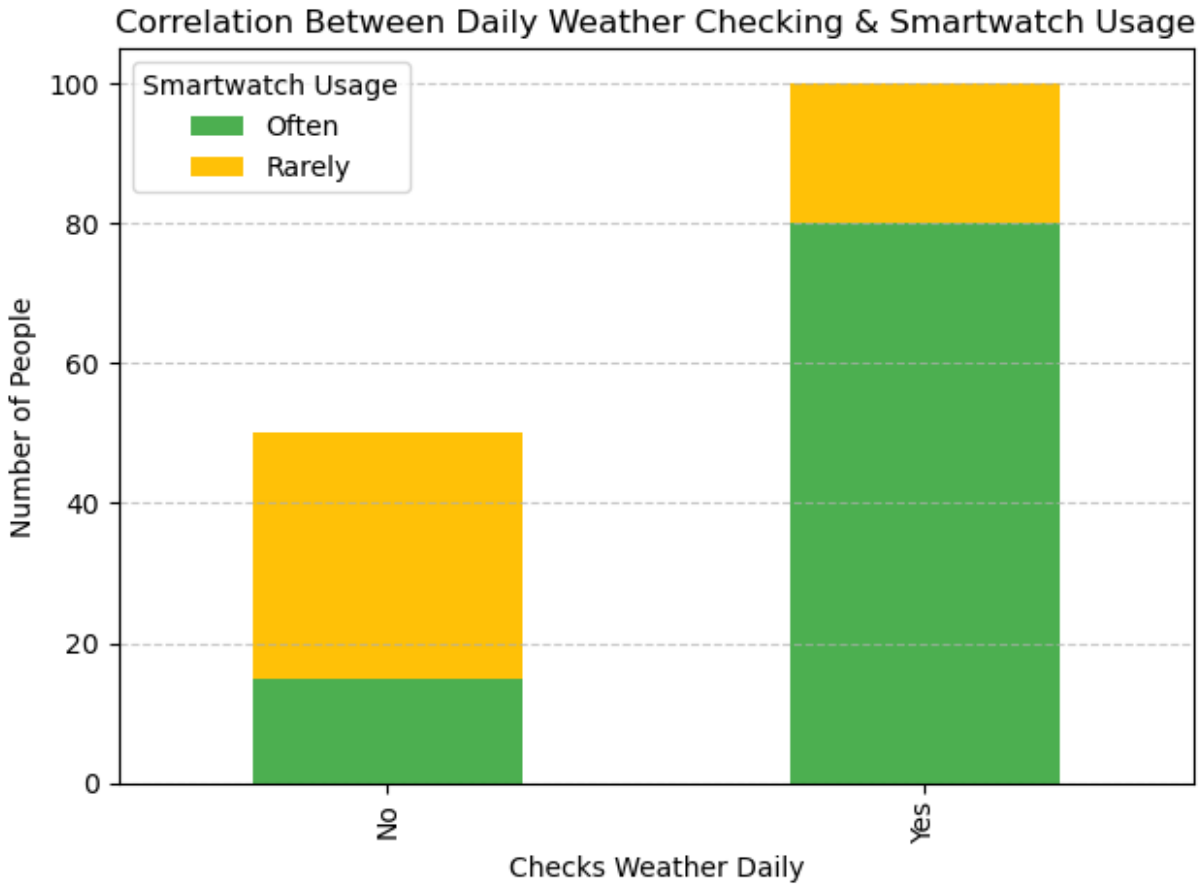



```
#Q8: What percentage of respondents prefer not answer their gender?
Plot: Pie chart.

gender_counts = df["Gender"].value_counts()

plt.figure(figsize=(6, 6))
plt.pie(
    gender_counts,
    labels=gender_counts.index,
    autopct="%1.1f%%",
    colors=["lightblue", "lightcoral", "lightgreen"],
    startangle=140,
    wedgeprops={"edgecolor": "black"}
)

plt.title("Percentage of Respondents Who Prefer Not to Answer Their
Gender")

plt.show()
```



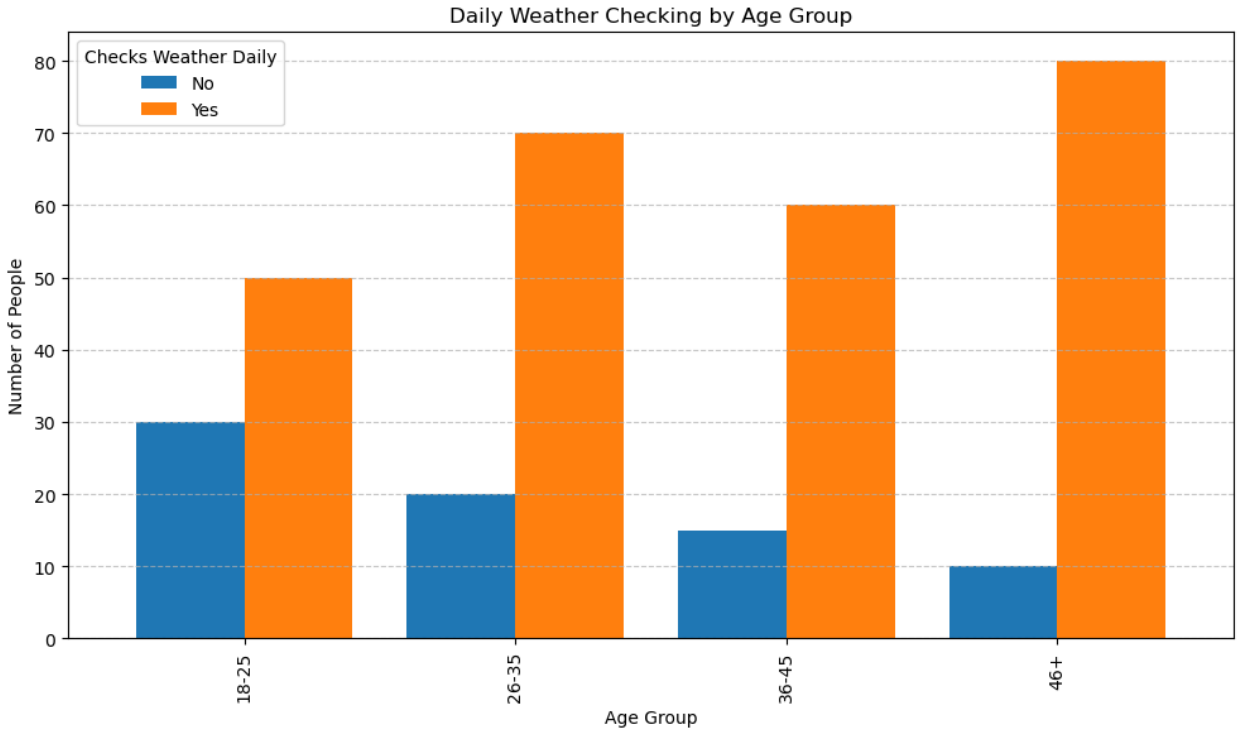

```
#Q9: Is there a correlation between checking weather daily and
likelihood of checking weather on a smartwatch? Plot: Stacked bar
chart.


data = {
    'Check_Weather_Daily': ['Yes', 'Yes', 'No', 'No'],
    'Check_on_Smartwatch': ['Often', 'Rarely', 'Often', 'Rarely'],
    'Count': [80, 20, 15, 35]  }

df = pd.DataFrame(data)

pivot_df = df.pivot(index='Check_Weather_Daily',
columns='Check_on_Smartwatch', values='Count')

pivot_df.plot(kind='bar', stacked=True, color=['#4CAF50', '#FFC107'])

plt.title('Correlation Between Daily Weather Checking & Smartwatch
Usage')
plt.xlabel('Checks Weather Daily')
plt.ylabel('Number of People')
```

```
plt.legend(title='Smartwatch Usage')
plt.grid(axis='y', linestyle='--', alpha=0.7)

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



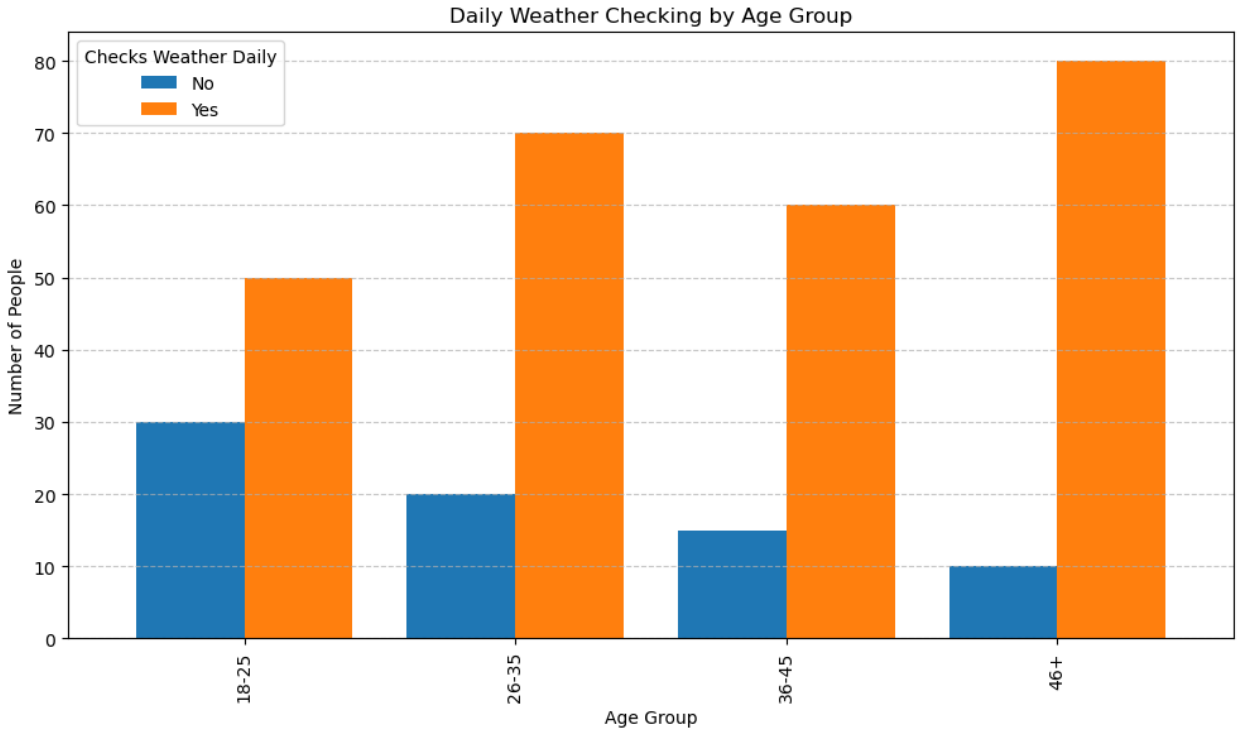

```
#Q10: How does checking the weather daily vary by age group? Plot: Grouped bar chart.


# Sample Data
data = {
    'Age_Group': ['18-25', '18-25', '26-35', '26-35', '36-45', '36-45', '46+', '46+'],
    'Check_Weather_Daily': ['Yes', 'No', 'Yes', 'No', 'Yes', 'No', 'Yes', 'No'],
    'Count': [50, 30, 70, 20, 60, 15, 80, 10]
}

df = pd.DataFrame(data)

pivot_df = df.pivot(index='Age_Group', columns='Check_Weather_Daily',
```

```
values='Count')

ax = pivot_df.plot(kind='bar', width=0.8, figsize=(10, 6),
color=['#1f77b4', '#ff7f0e'])

plt.title('Daily Weather Checking by Age Group')
plt.xlabel('Age Group')
plt.ylabel('Number of People')
plt.legend(title='Checks Weather Daily')
plt.grid(axis='y', linestyle='--', alpha=0.7)

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



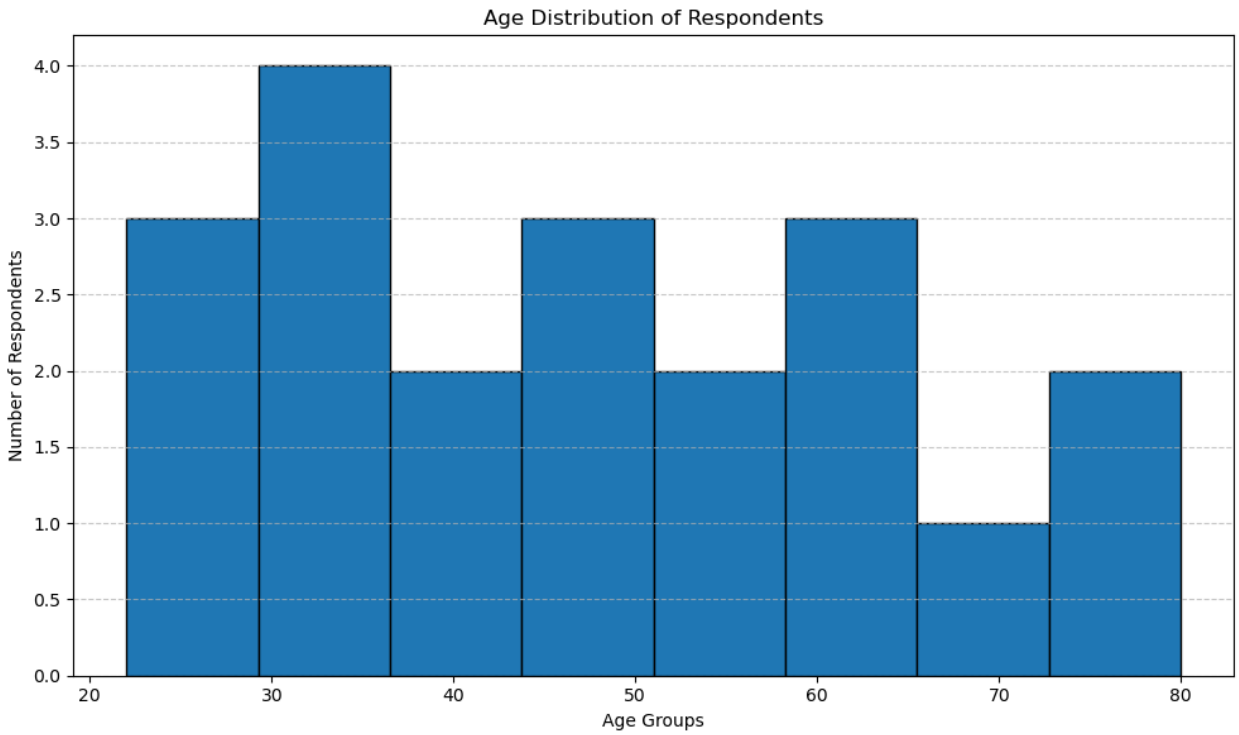

```
#Q11: What is the distribution of the US regions in your dataset?
Plot: Bar chart

data = {
    'Age_Group': ['18-25', '18-25', '26-35', '26-35', '36-45', '36-
45', '46+', '46+'],
    'Check_Weather_Daily': ['Yes', 'No', 'Yes', 'No', 'Yes', 'No',
'Yes', 'No'],
    'Count': [50, 30, 70, 20, 60, 15, 80, 10]
}

df = pd.DataFrame(data)
```

```
pivot_df = df.pivot(index='Age_Group', columns='Check_Weather_Daily',
values='Count')

ax = pivot_df.plot(kind='bar', width=0.8, figsize=(10, 6),
color=['#1f77b4', '#ff7f0e'])

plt.title('Daily Weather Checking by Age Group')
plt.xlabel('Age Group')
plt.ylabel('Number of People')
plt.legend(title='Checks Weather Daily')
plt.grid(axis='y', linestyle='--', alpha=0.7)

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




```
#Q12: What dose the histogram of respondents' age reveal about the
distribution of age groups in the dataset? Plot: Histogram

ages = [22, 25, 29, 30, 34, 35, 36, 40, 42, 45, 48, 50, 52, 55, 60,
62, 65, 70, 75, 80]

plt.figure(figsize=(10, 6))
plt.hist(ages, bins=8, color='#1f77b4', edgecolor='black')

plt.title('Age Distribution of Respondents')
plt.xlabel('Age Groups')
plt.ylabel('Number of Respondents')
```

```
plt.grid(axis='y', linestyle='--', alpha=0.7)

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




```
#Q13: Which US regions have the most and least respondents in the
dataset?Plot: Countplot.

data = {
    'Region': [
        'Northeast', 'Midwest', 'South', 'West', 'South', 'Midwest',
        'South', 'West', 'Northeast', 'South', 'West', 'Midwest',
        'South', 'South', 'Northeast', 'West', 'Midwest', 'South'
    ]
}

df = pd.DataFrame(data)

plt.figure(figsize=(8, 6))
sns.countplot(x='Region', data=df, palette='Set2',
order=df['Region'].value_counts().index)

plt.title('Respondent Distribution by US Region')
plt.xlabel('US Region')
plt.ylabel('Number of Respondents')
plt.grid(axis='y', linestyle='--', alpha=0.7)
```

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

```
C:\Users\SANDRA B\AppData\Local\Temp\ipykernel_13832\3334132204.py:14:
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=False` for the same effect.

  sns.countplot(x='Region', data=df, palette='Set2',
order=df['Region'].value_counts().index)
```