CSCI3022 Summer 23

Final Project

Alberto Espinosa
ID: 109749564.
Dr. Osita Onyejekwe
Data Science 3022
Summer 2023

Introduction/Background

I will be conducting a thorough analysis of a data set using statistical modeling techniques learned in class.

The data set is made up of survey data on smoking habits of people from the United Kingdom. It contains the demographic characteristics of smokers, some of their habits regarding smoking, and gross income. The data frame has *1691* observations on *12* variables. This data was collected from STEM Learning Website (https://www.stem.org.uk/resources/elibrary/resource/28452/large-datasets-stats4schools) compiled from the responses given by over 1,500 people to a survey. The data set was found using Kaggle (https://www.kaggle.com/datasets/utkarshx27/smoking-dataset-from-uk). It is an observational dataset since it is a survey that was handed out to people in the UK. The type of sampling done for the survey was not specified by the organization. This data was intended for student who are learning data science, and more specifically, to draw relationships regarding who is more likely to smoke in the UK.

By analyzing this data set I hope to gain a deeper understadning regarding smoking habits of people so that it may help me during my own personal journey to stop smoking and eventually live a healthier life.

Loading and Cleaning Data

In [2]:
```
import numpy as np
import pandas as pd
import seaborn as sns
from scipy import stats
import matplotlib.pylab as plt
%matplotlib inline
    # 'inline' puts your graph in the cell versus popup window
```

Description of the data set

Column	Description
gender	Gender of the participant
age	Age of the participant
marital_status	Marital status (Divorced, Married, Separated, Single and Widowed).
highest_qualification	Highest education level (A Levels, Degree, GCSE/CSE, GCSE/O Level, Higher/Sub Degree, No Qualification, ONC/BTEC and Other/Sub Degree).
nationality	Nationality (British, English, Irish, Scottish, Welsh, Other, Refused and Unknown).
ethnicity	Ethnicity (Asian, Black, Chinese, Mixed, White and Refused Unknown).
gross_income	Gross income (Under 2,600, 2,600 to 5,200, 5,200 to 10,400, 10,400 to 15,600, 15,600 to 20,800, 20,800 to 28,600, 28,600 to 36,400, Above 36,400, Refused and Unknown).
region	Region (London, Midlands & East Anglia, Scotland, South East, South West, The North and Wales).
smoke	Smoking status (No and Yes).
amt_weekends	Number of cigarettes smoked per day on weekends.
amt_weekdays	Number of cigarettes smoked per day on weekdays.
type	Type of cigarettes smoked (Packets, Hand-Rolled, Both/Mainly Packets and Both/Mainly Hand-Rolled.)

**This description of the data set was retrieved from [Kaggle (https://www.kaggle.com/datasets/utkarshx27/smoking-dataset-from-uk)](https://www.kaggle.com/datasets/utkarshx27/smoking-dataset-from-uk)*

Load the data from the file "smoking.csv"

In [3]:
```
# Loading the data set into JupyterNotebook
dfSmokingDirty = pd.read_csv("smoking.csv")
```

In [43]:
```
# How many rown in the data set
total_rows = len(dfSmokingDirty)
print("There are {} rows in the data set.".format(total_rows))

null_type_rows = dfSmokingDirty['type'].isnull()
null_count = null_entries_amt_weekends.sum()

print("There are only {} 'type' entries in the data set.".format(null_count
```

```
There are 1691 rows in the data set.
There are only 1270 'type' entries in the data set.
```

In [5]:
```
# Display the info of the data frame to uderstand it better
dfSmokingDirty.info()
# Display the first few rows to understand the nature of the variables
dfSmokingDirty.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1691 entries, 0 to 1690
Data columns (total 13 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Unnamed: 0             1691 non-null   int64
 1   gender                 1691 non-null   object
 2   age                    1691 non-null   int64
 3   marital_status         1691 non-null   object
 4   highest_qualification  1691 non-null   object
 5   nationality            1691 non-null   object
 6   ethnicity              1691 non-null   object
 7   gross_income           1691 non-null   object
 8   region                 1691 non-null   object
 9   smoke                  1691 non-null   object
 10  amt_weekends           421 non-null    float64
 11  amt_weekdays           421 non-null    float64
 12  type                   421 non-null    object
dtypes: float64(2), int64(2), object(9)
memory usage: 171.9+ KB
```

Out[5]:

	Unnamed: 0	gender	age	marital_status	highest_qualification	nationality	ethnicity	gross_income
0	1	Male	38	Divorced	No Qualification	British	White	2,600 to 5,200
1	2	Female	42	Single	No Qualification	British	White	Under 2,600
2	3	Male	40	Married	Degree	English	White	28,600 to 36,400
3	4	Female	40	Married	Degree	English	White	10,400 to 15,600
4	5	Female	39	Married	GCSE/O Level	British	White	2,600 to 5,200

Data Cleaning

In [6]:
```
dfSmoking = dfSmokingDirty
```

In [7]:
```
#Function is adapted from CSCI3022_S23_HW2.ipynb
def fix_amt_weekends(val):
    # check for null values and set to 0
    if pd.isnull(val):
        return int(0)
    return int(val)

def fix_amt_weekdays(val):
    # check for null values and set to 0
    if pd.isnull(val):
        return int(0)
    return int(val)
```

In [8]:
```
dfSmoking.loc[:, "amt_weekends"] = dfSmoking.loc[:, "amt_weekends"].apply(f
dfSmoking.loc[:, "amt_weekdays"] = dfSmoking.loc[:, "amt_weekdays"].apply(f
# dfSmoking.head(10)
```

In [9]:
```
dfSmoking = dfSmoking.drop(['Unnamed: 0', 'type'], axis = 1)
```

In [10]:
```
dfSmoking.head(10)
```

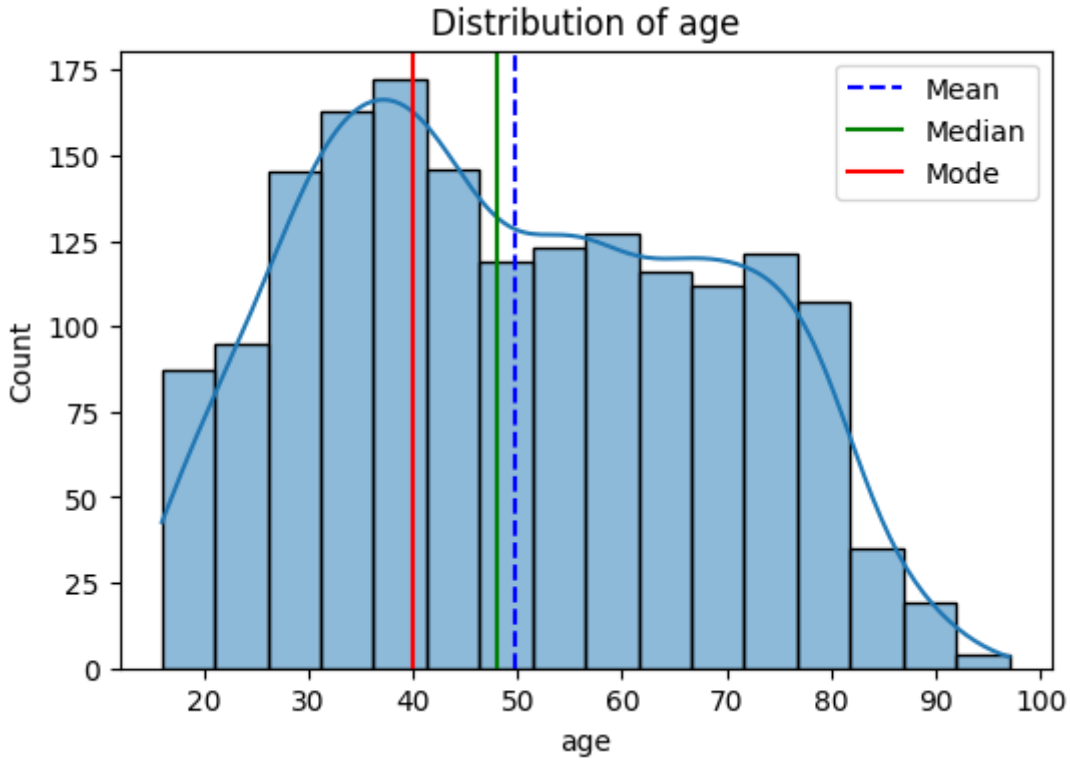
Out[10]:

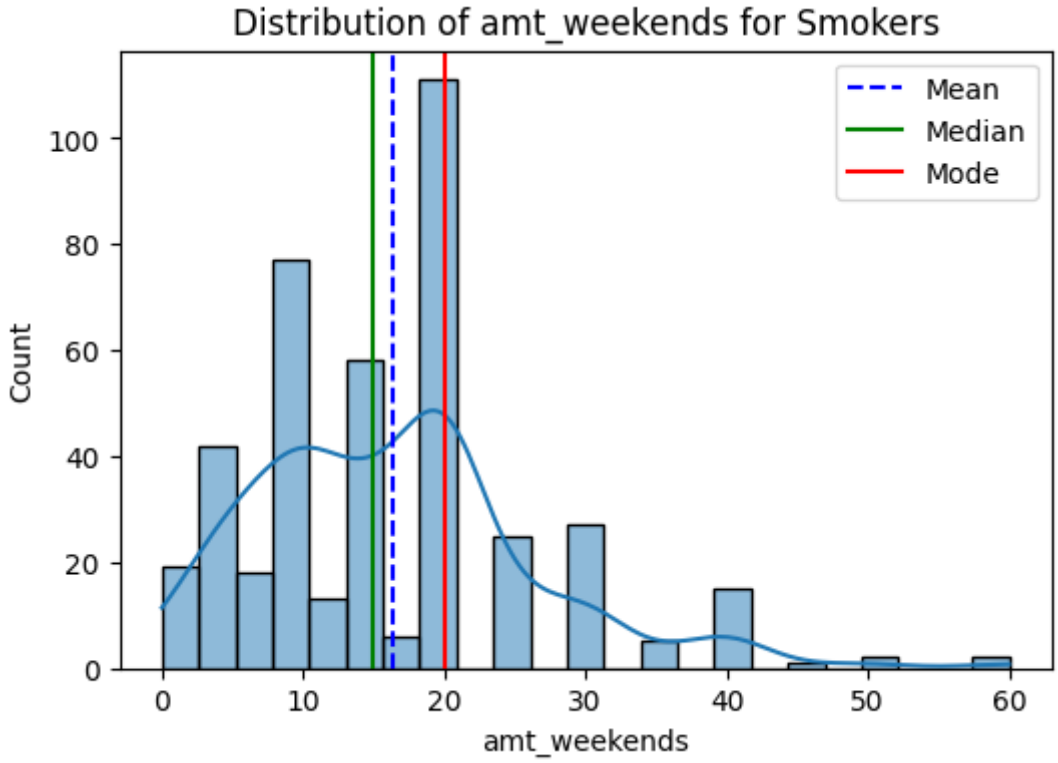
	gender	age	marital_status	highest_qualification	nationality	ethnicity	gross_income	region	sm
0	Male	38	Divorced	No Qualification	British	White	2,600 to 5,200	The North	
1	Female	42	Single	No Qualification	British	White	Under 2,600	The North	
2	Male	40	Married	Degree	English	White	28,600 to 36,400	The North	
3	Female	40	Married	Degree	English	White	10,400 to 15,600	The North	
4	Female	39	Married	GCSE/O Level	British	White	2,600 to 5,200	The North	
5	Female	37	Married	GCSE/O Level	British	White	15,600 to 20,800	The North	
6	Male	53	Married	Degree	British	White	Above 36,400	The North	
7	Male	44	Single	Degree	English	White	10,400 to 15,600	The North	
8	Male	40	Single	GCSE/CSE	English	White	2,600 to 5,200	The North	
9	Female	41	Married	No Qualification	English	White	5,200 to 10,400	The North	

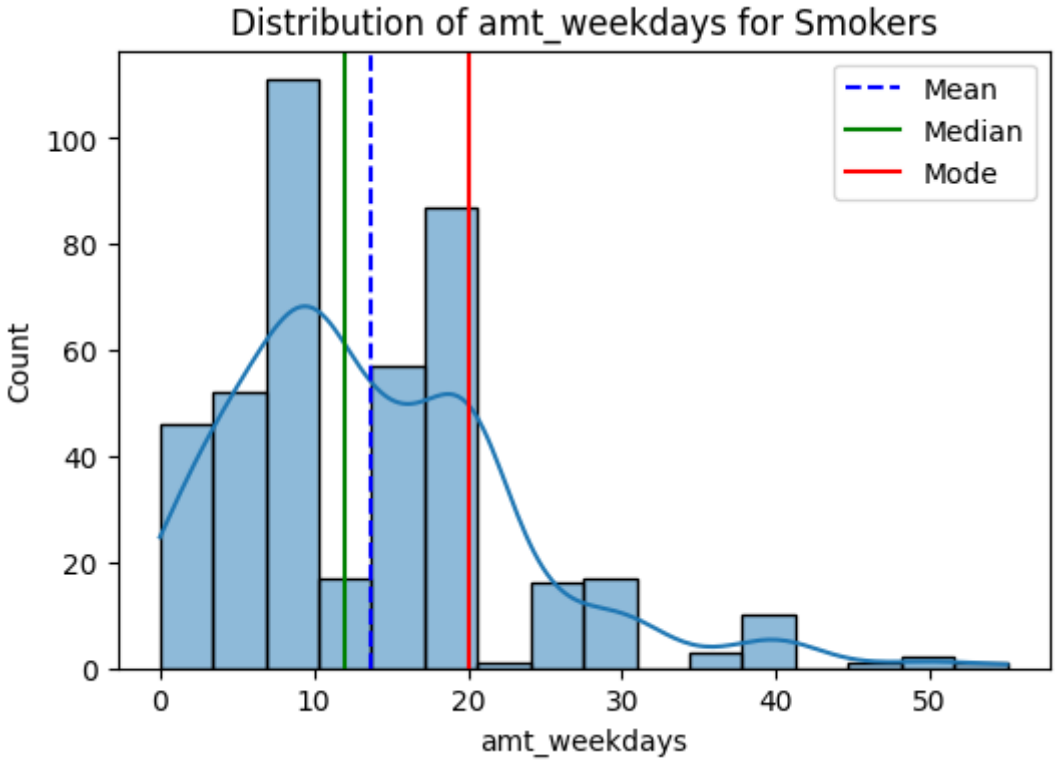
Exploratory Data Analysis

Visualize General data to draw more meaningful conclusions.

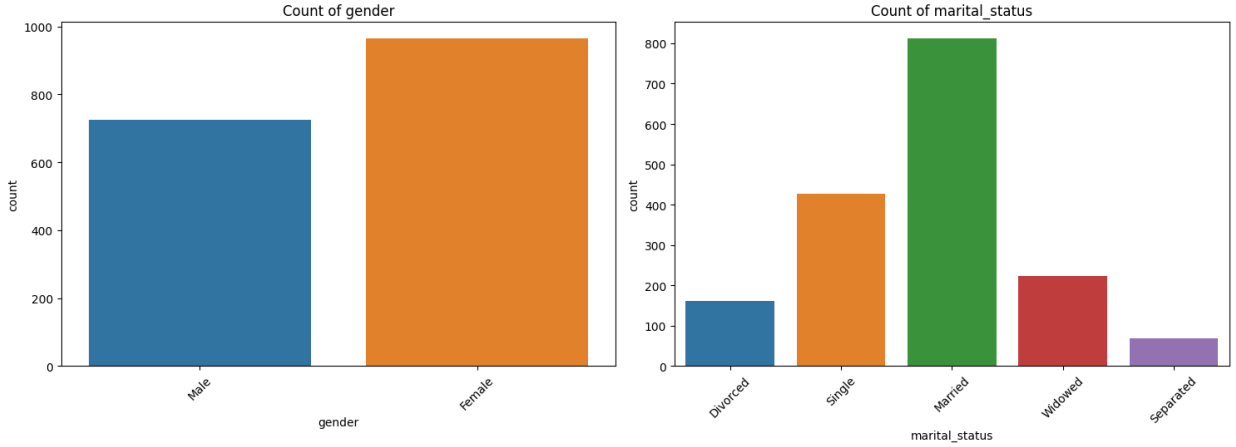
In [48]:
```
# Plot the variables that consist only of float64/Ints
# female_survived=len(dfTitanic.loc[(dfTitanic["Sex"] =='female') & (dfTita

#calculate the mean, median, mode
mean_col = dfSmoking.loc[:, 'age']
median = dfSmoking.loc[:, 'age'].median()
mode = dfSmoking.loc[:, 'age'].mode().tolist()[0]
#set figure size
plt.figure(figsize=(6, 4))
# plot
sns.histplot(data=dfSmoking, x='age', kde=True)
plt.title(f'Distribution of age')
#lines for the data
plt.axvline(x=np.mean(mean_col), color='b', linestyle='--', label='Mean')
plt.axvline(median, color='g', linestyle='-', label='Median')
plt.axvline(mode, color='r', linestyle='-', label='Mode')
plt.legend()
plt.show()

#only for smokers
# dfSmokers = dfSmoking[dfSmoking['smoke'] == 'Yes']
#survived_age = dfTitanic.loc[(dfTitanic["Survived"] ==1), ["Age"]]
dfSmokers = dfSmoking.loc[(dfSmoking["smoke"] == 'Yes'), :]

floats = ['amt_weekends', 'amt_weekdays']

#Iterate throught all 3 of them
for var in floats:
    # Calculate the mean median and mode for the smokers
    median = dfSmokers.loc[:, var].median()
    mode = dfSmokers.loc[:, var].mode().tolist()[0]
    mean_col = dfSmokers.loc[:, var]
    #create the standard figure size
    plt.figure(figsize=(6, 4))
    # use seaborn to plot
    sns.histplot(data=dfSmokers, x=var, kde=True)
    #Plot the Data
    plt.title(f'Distribution of {var} for Smokers')
    #plot the mean of the data
    plt.axvline(x=np.mean(mean_col), color='b', linestyle='--', label='Mean
    plt.axvline(median, color='g', linestyle='-', label='Median')
    plt.axvline(mode, color='r', linestyle='-', label='Mode')
    #show the plot
    plt.legend()
    plt.show()

# References:
# https://seaborn.pydata.org/generated/seaborn.countplot.html
```


Distribution of age
Mean
Median
Mode
Count
age
0
25
50
75
100
125
150
175
20
30
40
50
60
70
80
90
100

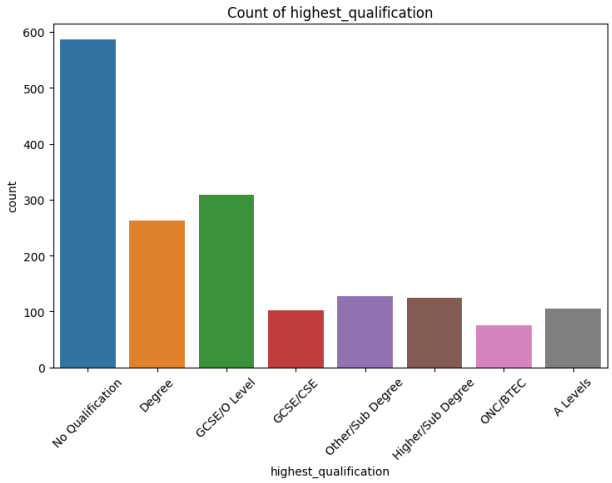

Distribution of amt_weekends for Smokers
Mean
Median
Mode
Count
amt_weekends
0
20
40
60
80
100
0
10
20
30
40
50
60

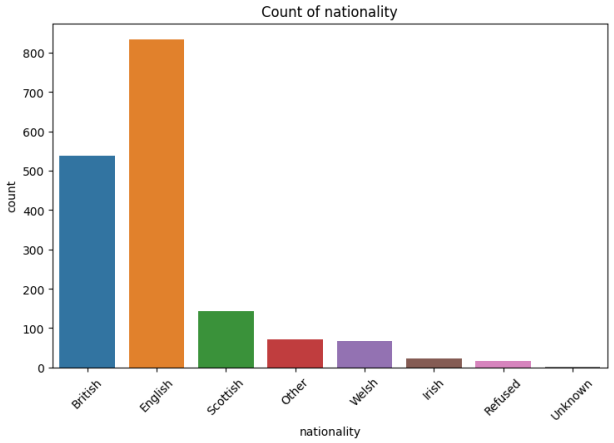

Distribution of amt_weekdays for Smokers
Mean
Median
Mode
Count
amt_weekdays
0
20
40
60
80
100
0
10
20
30
40
50

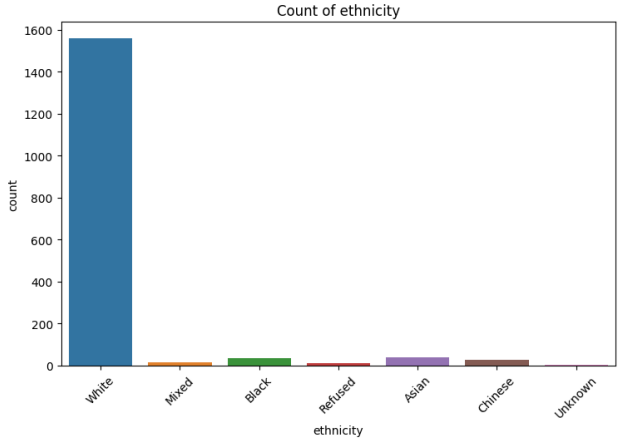
In [12]:
```
# # Initialize figure and axis
# fig, ax = plt.subplots(figsize=(8,4))   # ax is just a variable name of y
# # Plot histogram
# dfSmoking.hist(column="age", ax=ax);         # The first ax is a keyword

#Object variables
object_vars = ['gender', 'marital_status', 'highest_qualification', 'nation
cols = 2
rows = 4

# print(f"number of rows is {rows}")

plt.figure(figsize=(15, rows * 6))  # Adjust the height based on the number

for idx, var in enumerate(object_vars):
    plt.subplot(rows, cols, idx + 1)
    sns.countplot(data=dfSmoking, x=var)
    plt.title(f'Count of {var}')
    plt.xticks(rotation=45)

plt.tight_layout()
plt.show()

# Reference for code:
# https://joserzapata.github.io/courses/python-ciencia-datos/visualizacion/
# https://seaborn.pydata.org/tutorial/distributions.html
```

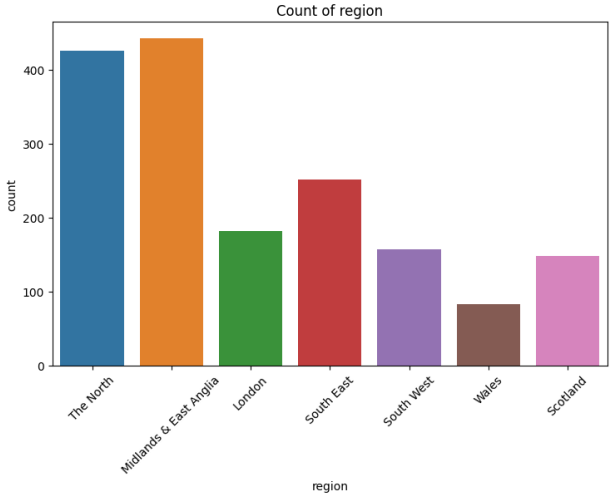

Count of gender
count
gender
Male
Female

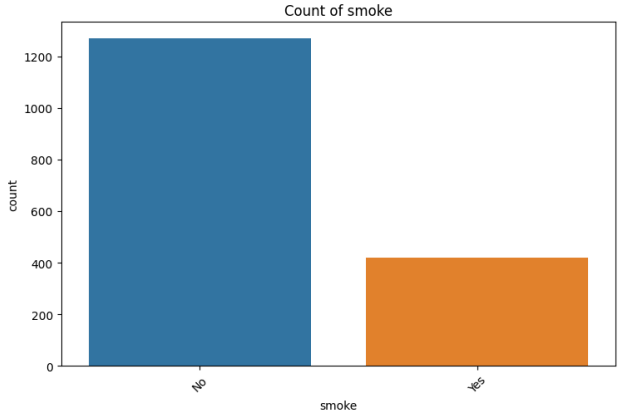
Count of marital_status
count
marital_status
Divorced
Single
Married
Widowed
Separated

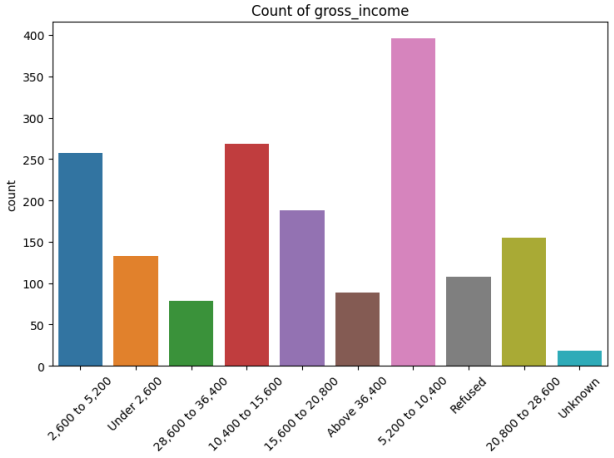

Count of highest_qualification
count
highest_qualification
No Qualification
Degree
GCSE/O Level
GCSE/CSE
Other/Sub Degree
Higher/Sub Degree
ONC/BTEC
A Levels


Count of nationality
count
nationality
British
English
Scottish
Other
Welsh
Irish
Refused
Unknown


Count of ethnicity
count
ethnicity
White
Mixed
Black
Refused
Asian
Chinese
Unknown


Count of region
count
region
The North
Midlands & East Anglia
London
South East
South West
Wales
Scotland

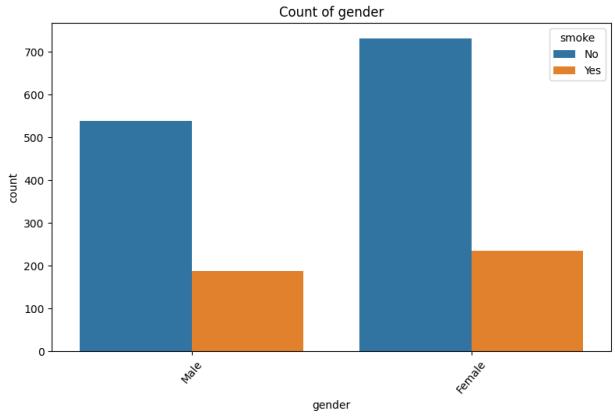

Count of smoke
count
smoke
No
Yes

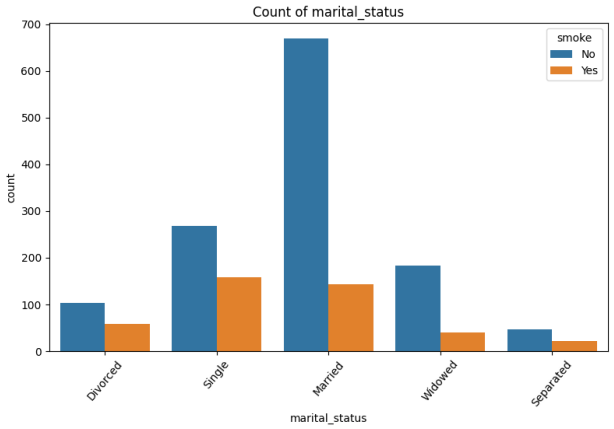

Count of gross_income
count
2,600 to 5,200
Under 2,600
28,600 to 36,400
10,400 to 15,600
15,600 to 20,800
Above 36,400
5,200 to 10,400
Refused
20,800 to 28,600
Unknown

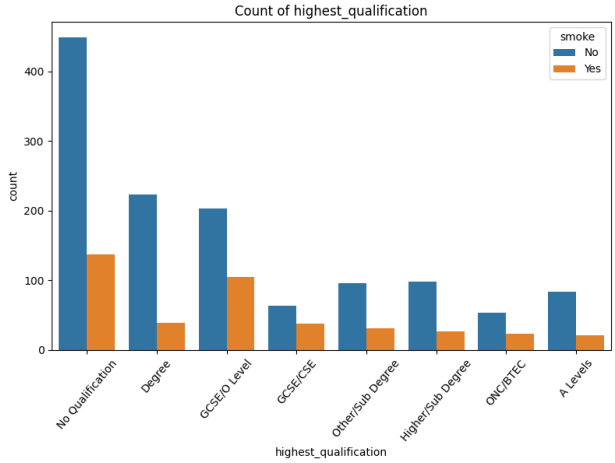
gross_income

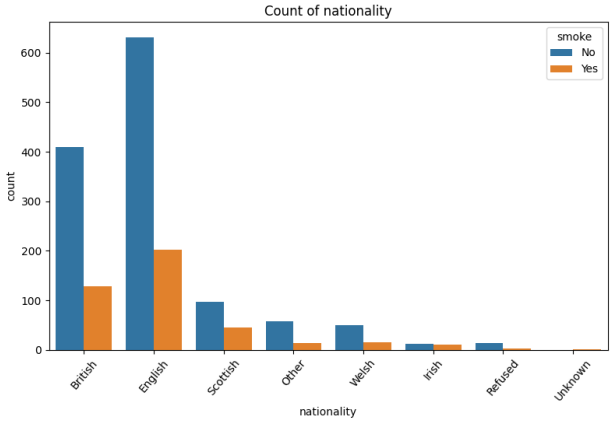
Visualize Relationshipts between variables

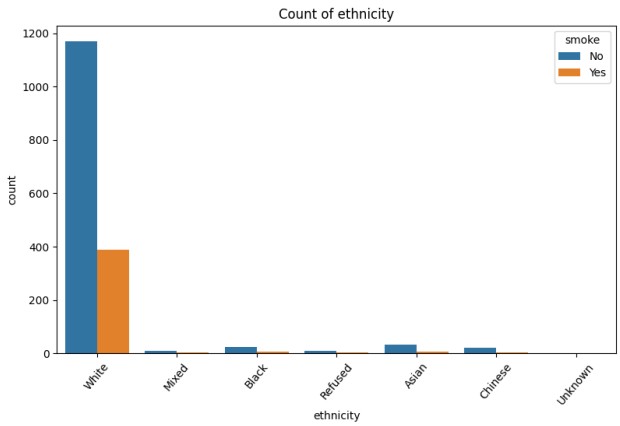
In [13]:
```
# Plot relationships between categorical vars and smokers
# object_vars = ['gender', 'marital_status', 'highest_qualification', 'nati
# for var in object_vars:
#     sns.catplot(x=var, hue="smoke", kind="count", data=dfSmoking)
#     plt.show()

object_vars = ['gender', 'marital_status', 'highest_qualification', 'nation
cols = 2
rows = 4

# print(f"number of rows is {rows}")

plt.figure(figsize=(16, rows * 6))  # Adjust the height based on the number

for idx, var in enumerate(object_vars, start=1):  # Start subplot indexing
    plt.subplot(rows, cols, idx)
    sns.countplot(data=dfSmoking, x=var, hue="smoke")
    plt.title(f'Count of {var}')
    plt.xticks(rotation=50)

plt.tight_layout()
plt.show()

sns.catplot(x="smoke", y="age", kind="box", data=dfSmoking)
plt.show()

#References:
# https://seaborn.pydata.org/generated/seaborn.countplot.html
#https://github.com/clair513/Seaborn-Tutorial/blob/master/Seaborn%20-%20Bar
```

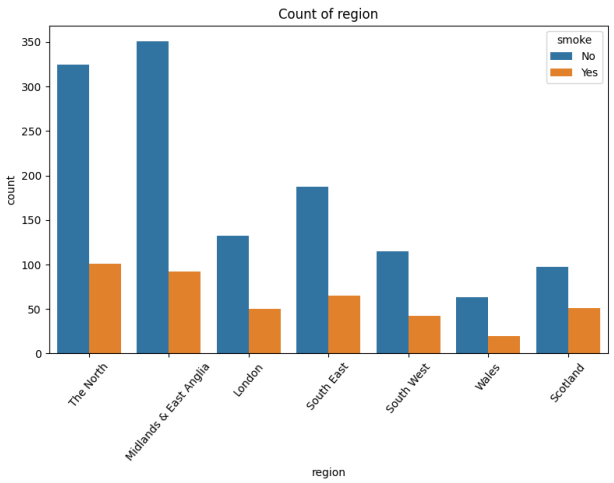

Count of gender
smoke
No
Yes
count
Male
Female
gender

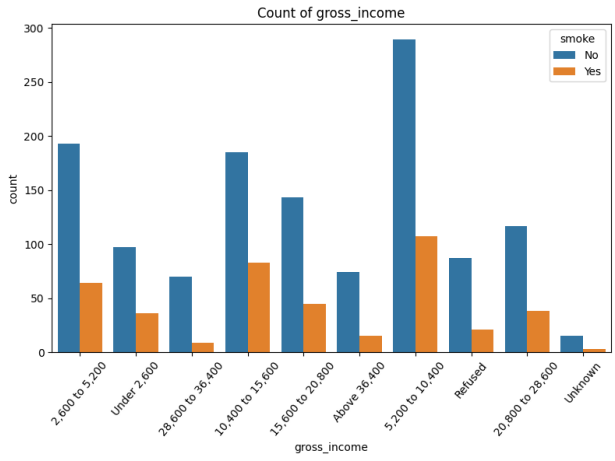

Count of marital_status
smoke
No
Yes
count
Divorced
Single
Married
Widowed
Separated
marital_status

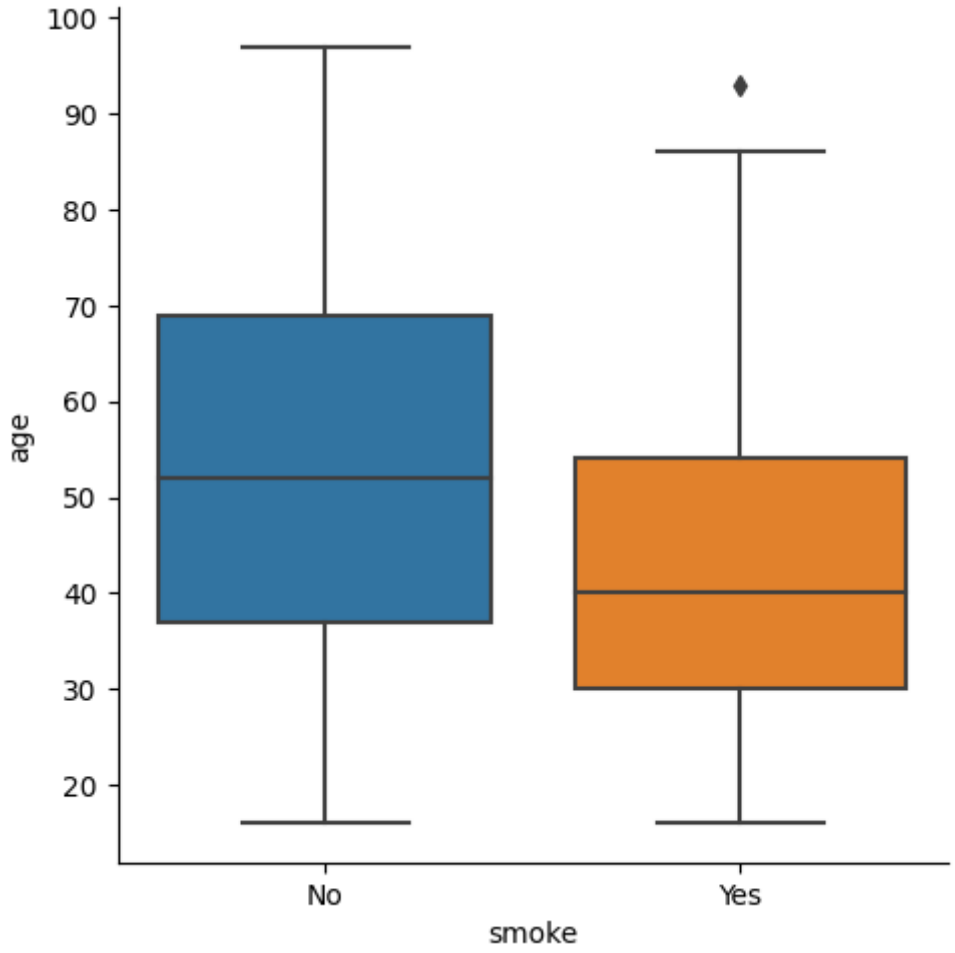

Count of highest_qualification
smoke
No
Yes
count
No Qualification
Degree
GCSE/O Level
GCSE/CSE
Other/Sub Degree
Higher/Sub Degree
ONC/BTEC
A Levels
highest_qualification


Count of nationality
smoke
No
Yes
count
British
English
Scottish
Other
Welsh
Irish
Refused
Unknown
nationality


Count of ethnicity
smoke
No
Yes
count
White
Mixed
Black
Refused
Asian
Chinese
Unknown
ethnicity

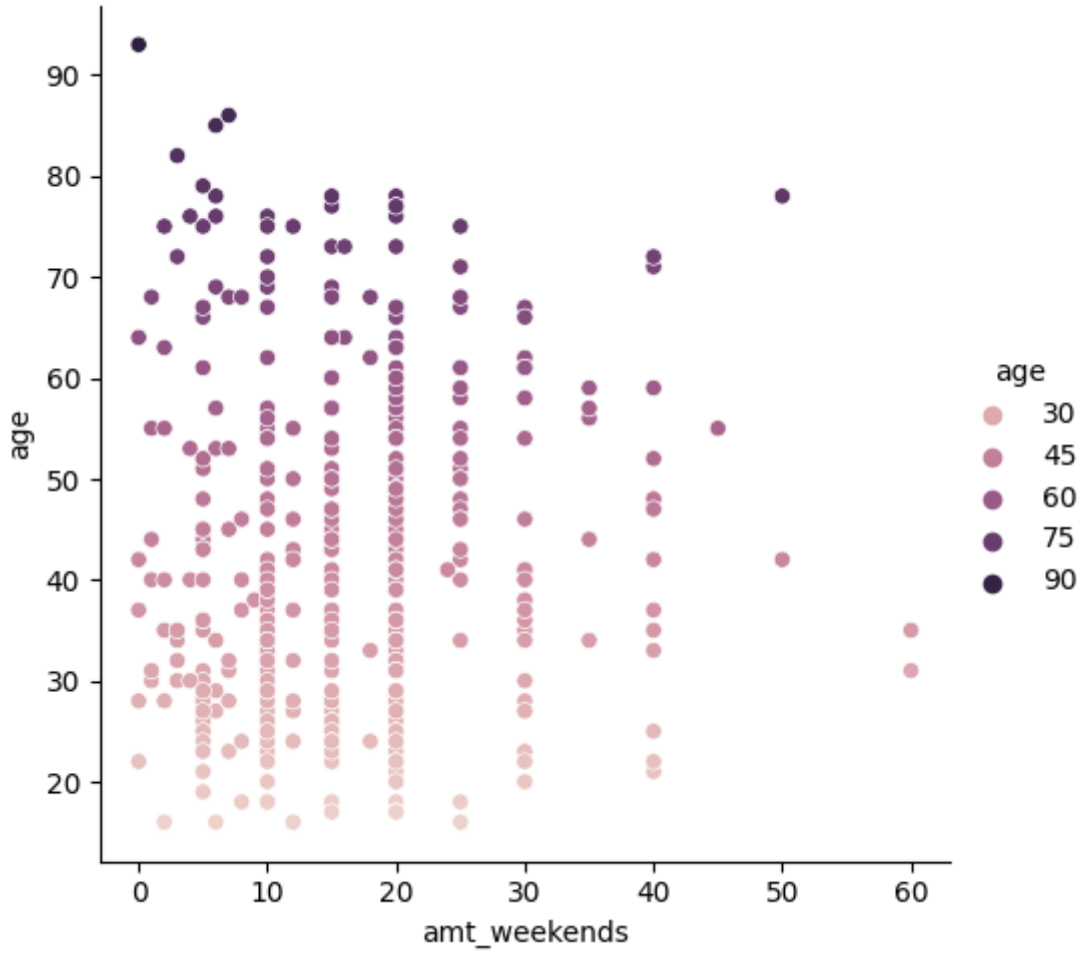

Count of region
smoke
No
Yes
count
The North
Midlands & East Anglia
London
South East
South West
Wales
Scotland
region

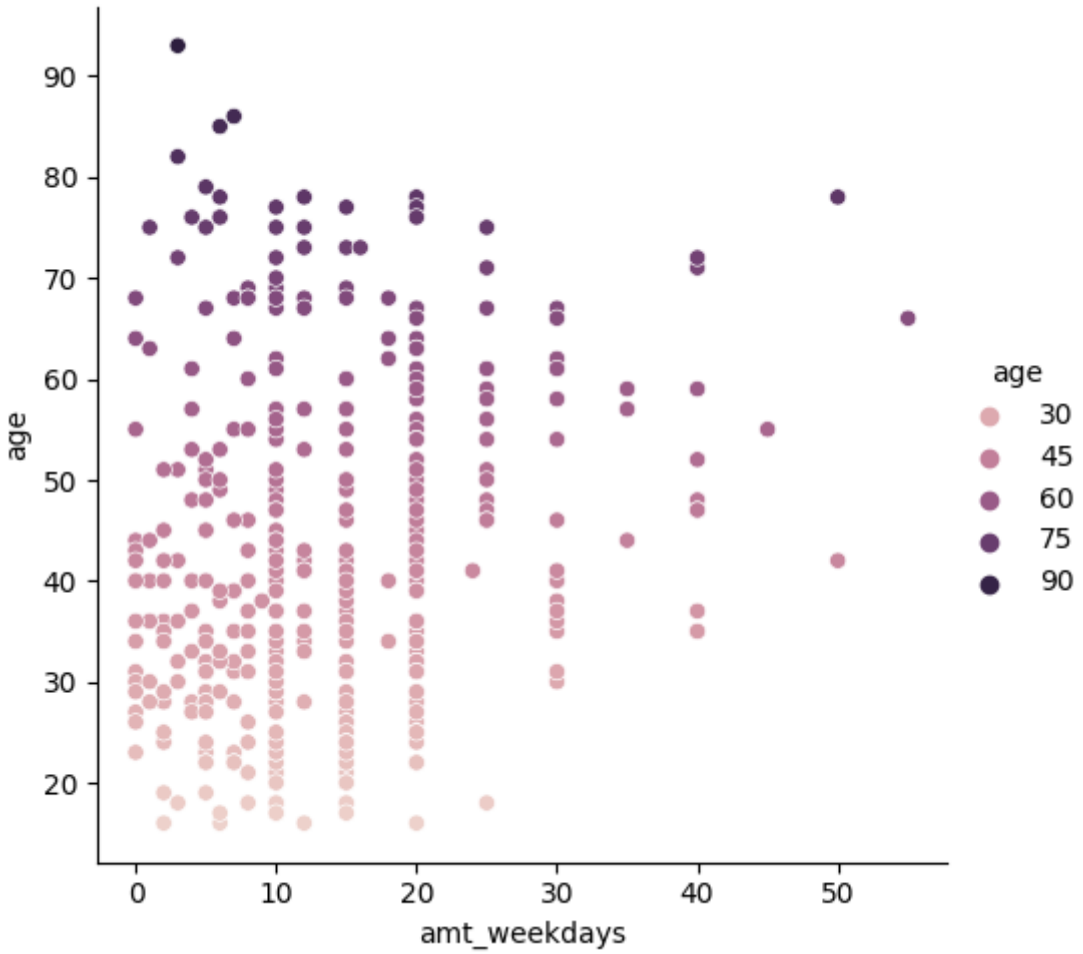

Count of gross_income
smoke
No
Yes
count
2,600 to 5,200
Under 2,600
28,600 to 36,400
10,400 to 15,600
15,600 to 20,800
Above 36,400
5,200 to 10,400
Refused
20,800 to 28,600
Unknown
gross_income


100
90
80
70
60
50
40
30
20
age
No
Yes
smoke

In [14]:
```
# sns.relplot(data=dfSmoking, x="gender", y="age", hue="smoke") highest_qua
sns.relplot(
    data=dfSmokers,
    x="amt_weekends", y="age", hue="age")

sns.relplot(
    data=dfSmokers,
    x="amt_weekdays", y="age", hue="age")

# sns.relplot(
#     data=dfSmokers,
#     x="age", y="gross_income", hue="gross_income")

#References:
# https://seaborn.pydata.org/generated/seaborn.countplot.html
#https://github.com/clair513/Seaborn-Tutorial/blob/master/Seaborn%20-%20Bar
#https://github.com/datacamp/COVID-19-EDA-tutorial/blob/master/notebooks/1-
```

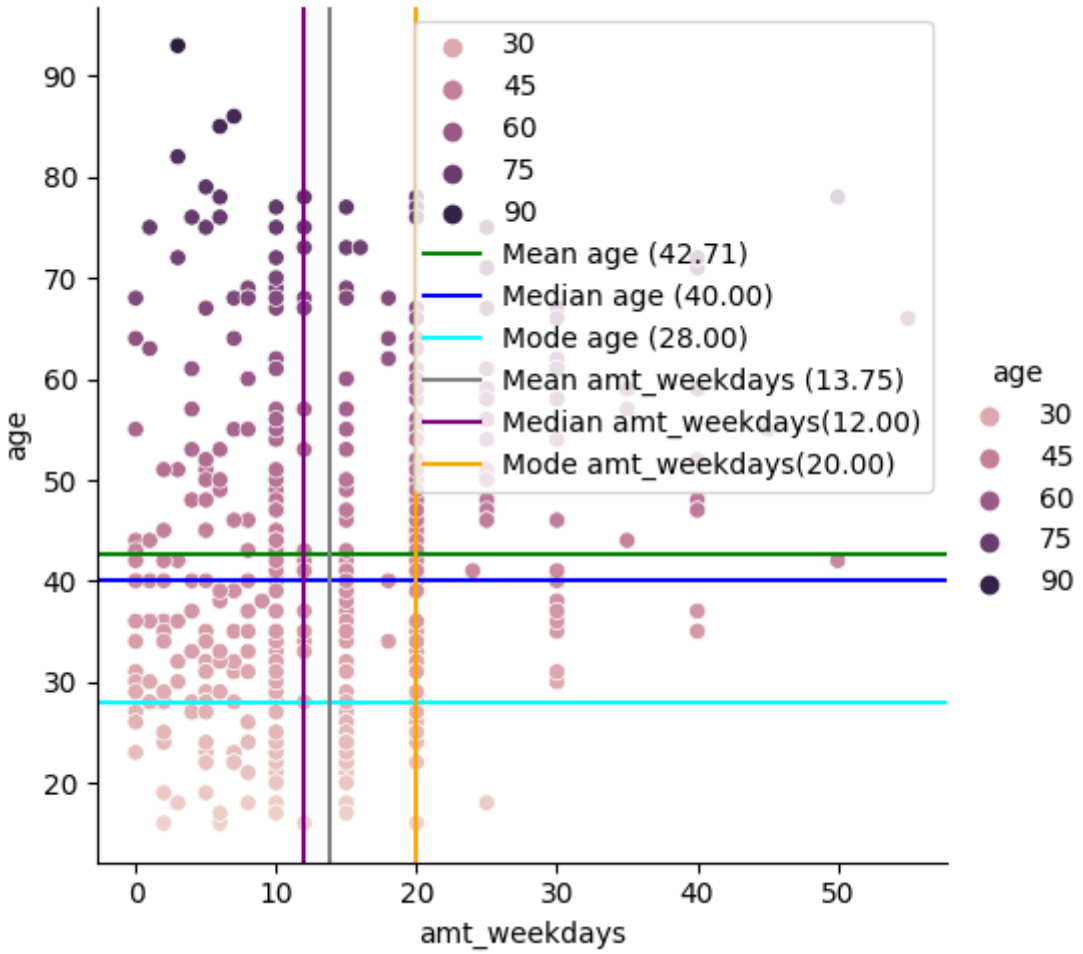
Out[14]: <seaborn.axisgrid.FacetGrid at 0x7f7f9a690d90>




90
80
70
60
50
40
30
20
age
0
10
20
30
40
50
amt_weekdays
age
30
45
60
75
90

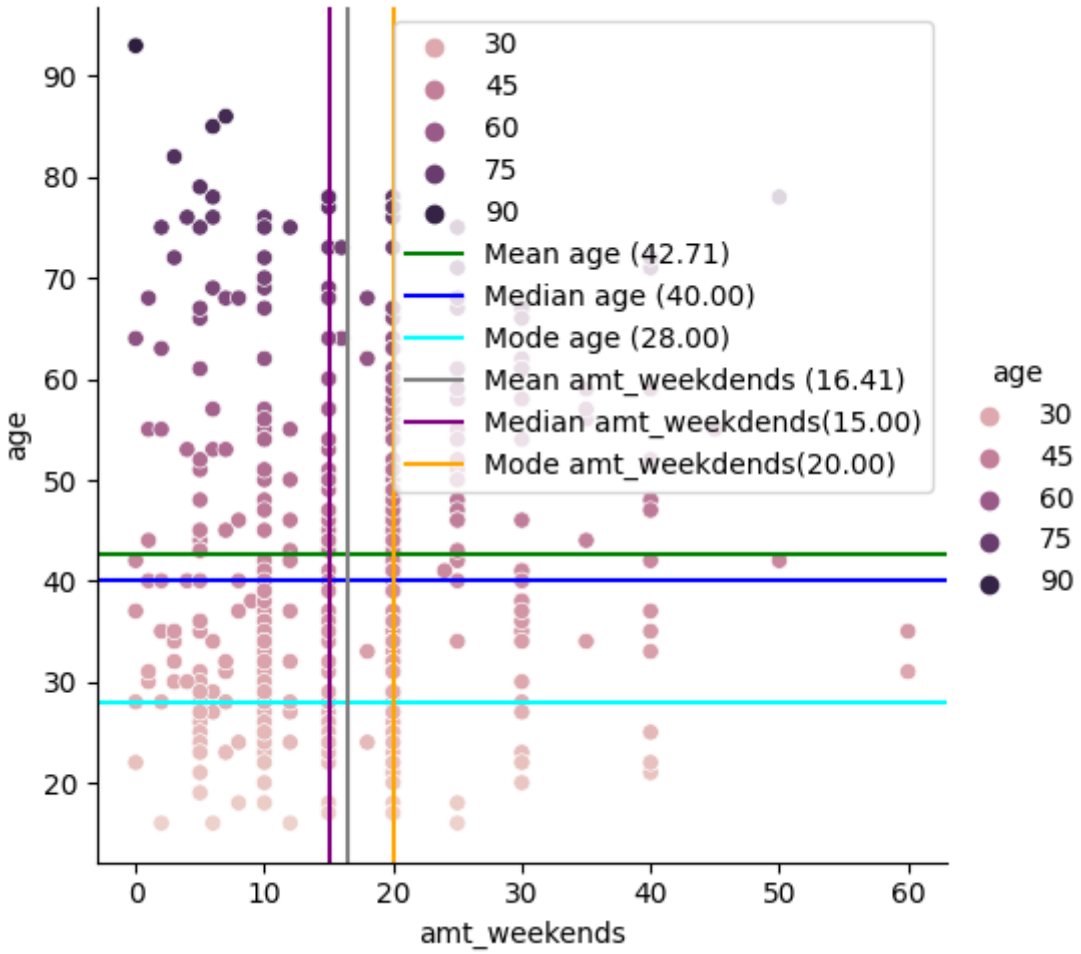
In [23]:

```
#Calc mean, median, and mode
mean_age = dfSmokers['age'].mean()
median_age = dfSmokers['age'].median()
mode_age = dfSmokers["age"].mode().iloc[0]
#repeat for the amount of ciggarettes per weekday
mean_amt_weekdays = dfSmokers['amt_weekdays'].mean()
median_amt_weekdays = dfSmokers['amt_weekdays'].median()
mode_amt_weekdays = dfSmokers["amt_weekdays"].mode().iloc[0]


# plot
scatter_plot = sns.relplot(
    data=dfSmokers,
    x="amt_weekdays", y="age", hue="age")

# Add lines for mean, median, mode
plt.axhline(mean_age, color='green', linestyle='-', label=f'Mean age ({mean
plt.axhline(median_age, color='blue', linestyle='-', label=f'Median age ({m
plt.axhline(mode_age, color='cyan', linestyle='-', label=f'Mode age ({mode_

plt.axvline(mean_amt_weekdays, color='grey', linestyle='-', label=f'Mean am
plt.axvline(median_amt_weekdays, color='purple', linestyle='-', label=f'Med
plt.axvline(mode_amt_weekdays, color='orange', linestyle='-', label=f'Mode


# Show the plot
scatter_plot.ax.legend()
plt.show()
```


30
45
60
75
90
Mean age (42.71)
Median age (40.00)
Mode age (28.00)
Mean amt_weekdays (13.75)
Median amt_weekdays(12.00)
Mode amt_weekdays(20.00)
age
amt_weekdays
0
10
20
30
40
50
20
30
40
50
60
70
80
90
age
30
45
60
75
90

In [51]:

```
#Calc mean, median, and mode
mean_age = dfSmokers['age'].mean()
median_age = dfSmokers['age'].median()
mode_age = dfSmokers["age"].mode().iloc[0]
#repeat for the amount of ciggarettes per weekday
mean_amt_weekdays = dfSmokers['amt_weekends'].mean()
median_amt_weekdays = dfSmokers['amt_weekends'].median()
mode_amt_weekdays = dfSmokers["amt_weekends"].mode().iloc[0]


# plot
scatter_plot = sns.relplot(
    data=dfSmokers,
    x="amt_weekends", y="age", hue="age")

# Add lines for mean, median, mode
plt.axhline(mean_age, color='green', linestyle='-', label=f'Mean age ({mean
plt.axhline(median_age, color='blue', linestyle='-', label=f'Median age ({m
plt.axhline(mode_age, color='cyan', linestyle='-', label=f'Mode age ({mode_

plt.axvline(mean_amt_weekdays, color='grey', linestyle='-', label=f'Mean am
plt.axvline(median_amt_weekdays, color='purple', linestyle='-', label=f'Med
plt.axvline(mode_amt_weekdays, color='orange', linestyle='-', label=f'Mode


# Show the plot
scatter_plot.ax.legend()
plt.show()
```



Descriptive Statistics

Determine meaningful numerical statistics regarding this data set in order to draw relationships.

Measure of disperstion:

- Range,
- Percentiles,
- IQR,
- Variance,
- Std Dev.

In [32]:
```
# List of columns you want to calculate statistics for
cols = ['age', 'amt_weekends', 'amt_weekdays']

print("Statistics for people that smoke in the data set \n")

# Loop through each column
for col in cols:

    col_data = dfSmokers[col]

    # Calculate the statistics
    col_range = col_data.max() - col_data.min()
    col_perc = col_data.describe(percentiles=[0.25, 0.50, 0.75])
    col_iqr = column_perc['75%'] - column_perc['25%']
    col_var = col_data.var()
    col_std = col_data.std()

    # Print the results
    print(f"Statistics for smokers column '{col}':")
    print("Range:", col_range)
    print("Percentiles:")
    print(col_perc)
    print("IQR:", col_iqr)
    print("Variance:", col_var)
    print("Std Dev.:", col_std)
    print("\n")
```

```
Statistics for people that smoke in the data set

Statistics for smokers column 'age':
Range: 77
Percentiles:
count    421.000000
mean      42.714964
std       16.179631
min       16.000000
25%       30.000000
50%       40.000000
75%       54.000000
max       93.000000
Name: age, dtype: float64
IQR: 24.0
Variance: 261.7804660106323
Std Dev.: 16.179631207497664


Statistics for smokers column 'amt_weekends':
Range: 60.0
Percentiles:
count    421.000000
mean      16.410926
std        9.892988
min        0.000000
25%       10.000000
50%       15.000000
75%       20.000000
max       60.000000
Name: amt_weekends, dtype: float64
IQR: 24.0
Variance: 97.8712136636127
Std Dev.: 9.892988105906765


Statistics for smokers column 'amt_weekdays':
Range: 55.0
Percentiles:
count    421.000000
mean      13.750594
std        9.388292
min        0.000000
25%        7.000000
50%       12.000000
75%       20.000000
max       55.000000
Name: amt_weekdays, dtype: float64
IQR: 24.0
Variance: 88.14002940843797
Std Dev.: 9.388292145456381
```

In [ ]: