

Part One: Basic Charts and Summaries

Tasks:

- Create a **bar chart** for gender and a **pie chart** for ethnic group.
 - Summarize the age data with a **five-number summary** (min, max, median, 1st quartile, 3rd quartile) and a **box plot**.
 - Calculate the **mean** and **standard deviation** of income and create a **histogram**.
-

Solution:

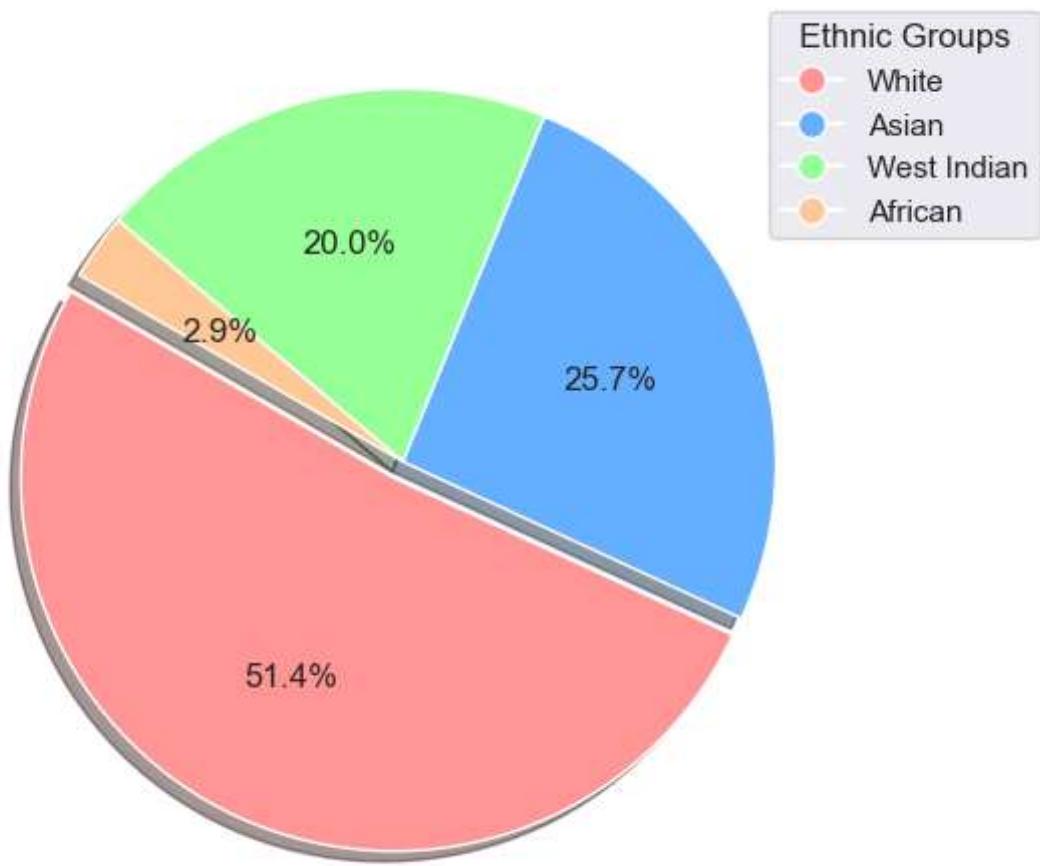
Bar Chart for Gender Distribution:



Caption: A bar chart visualizing the gender distribution, with more male participants than female.

Pie Chart for Ethnic Groups:

Ethnic Group Distribution



Caption: A pie chart showing the distribution of ethnic groups, highlighting diversity in the dataset.

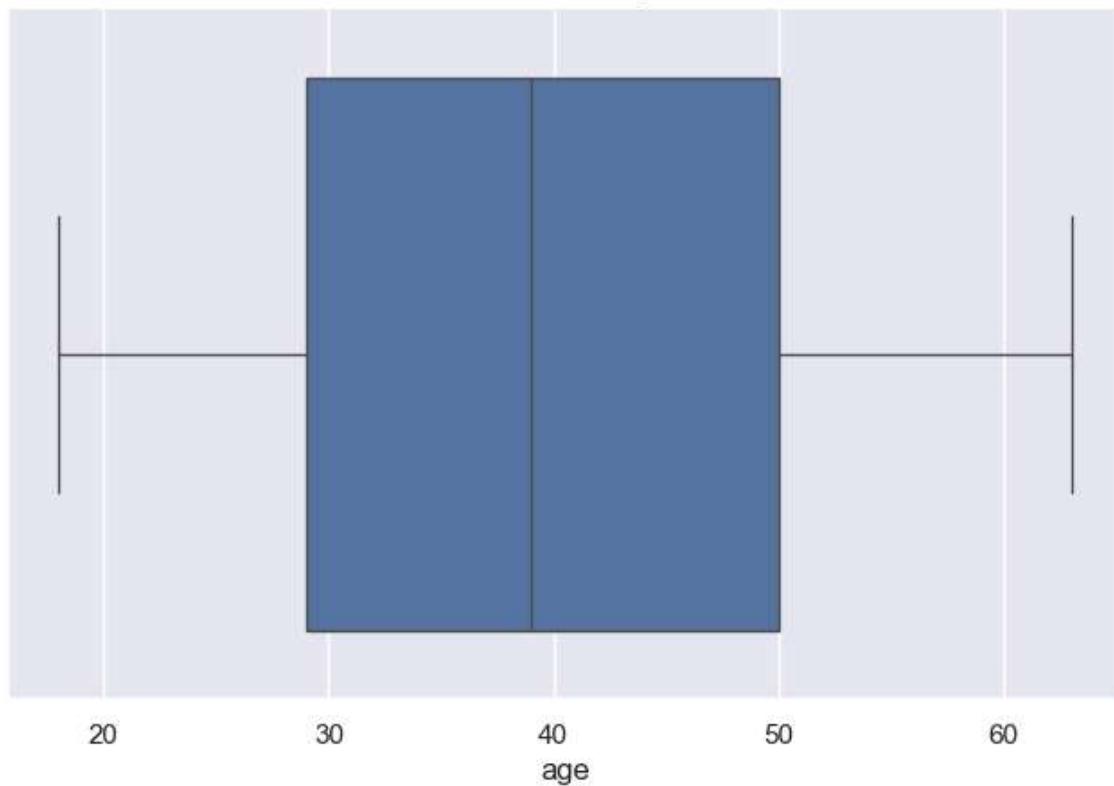
Table: Five-Number Summary for Age

Statistic	Value
Minimum	18.0
Q1 (25%)	29.0
Median	39.0
Q3 (75%)	50.0
Maximum	63.0

Caption: The five-number summary shows that the ages in the dataset range from 18 to 63, with a median of 39 years.

Box Plot for Age Distribution:

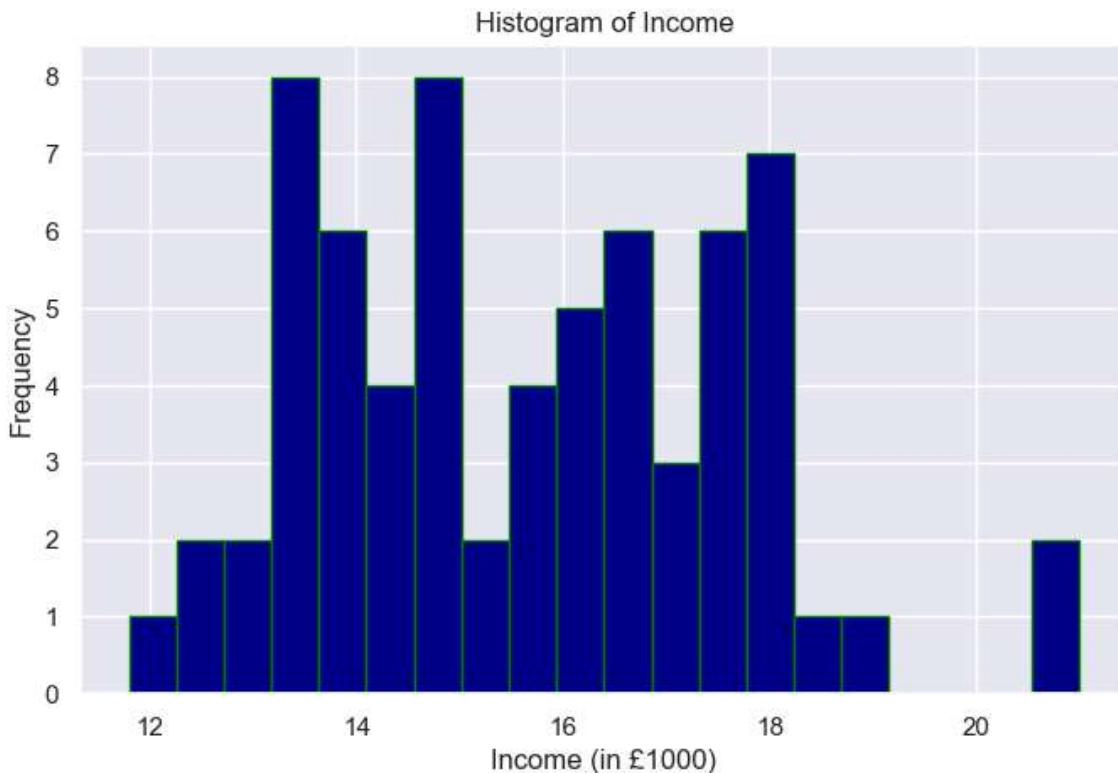
Box Plot of Age



Caption: A box plot representing the age distribution, showing that the majority of individuals are aged between 29 and 50.

- **Mean of Income:** 15.64
- **Standard Deviation of Income:** 1.99

Histogram of Income:



Caption: The income histogram demonstrates that the majority of individuals have an income concentrated around the mean of 15.64.

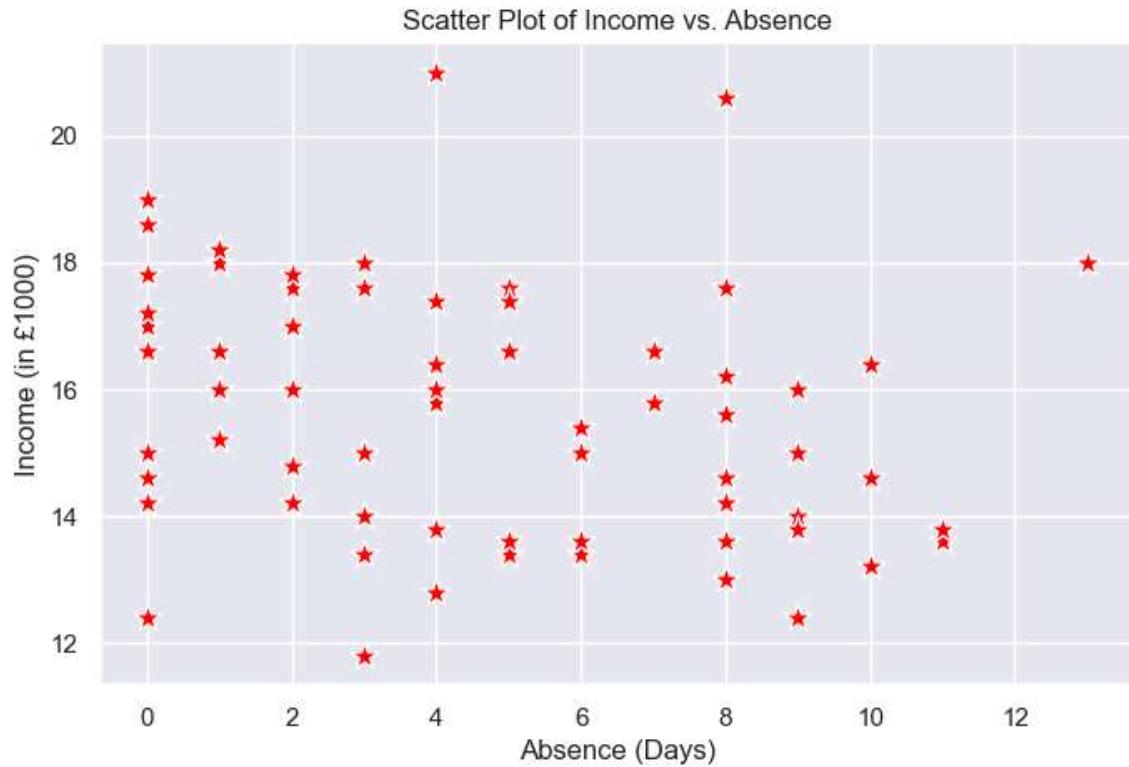
Part one: 2. Scatter Plot and Regression

Tasks:

- Create a **scatter plot** to visualize the relationship between **income** and **absence**.
 - Build a **simple linear regression model** with income as the dependent variable and absence as the independent variable.
 - Report the determination coefficient (R^2).
-

Solution:

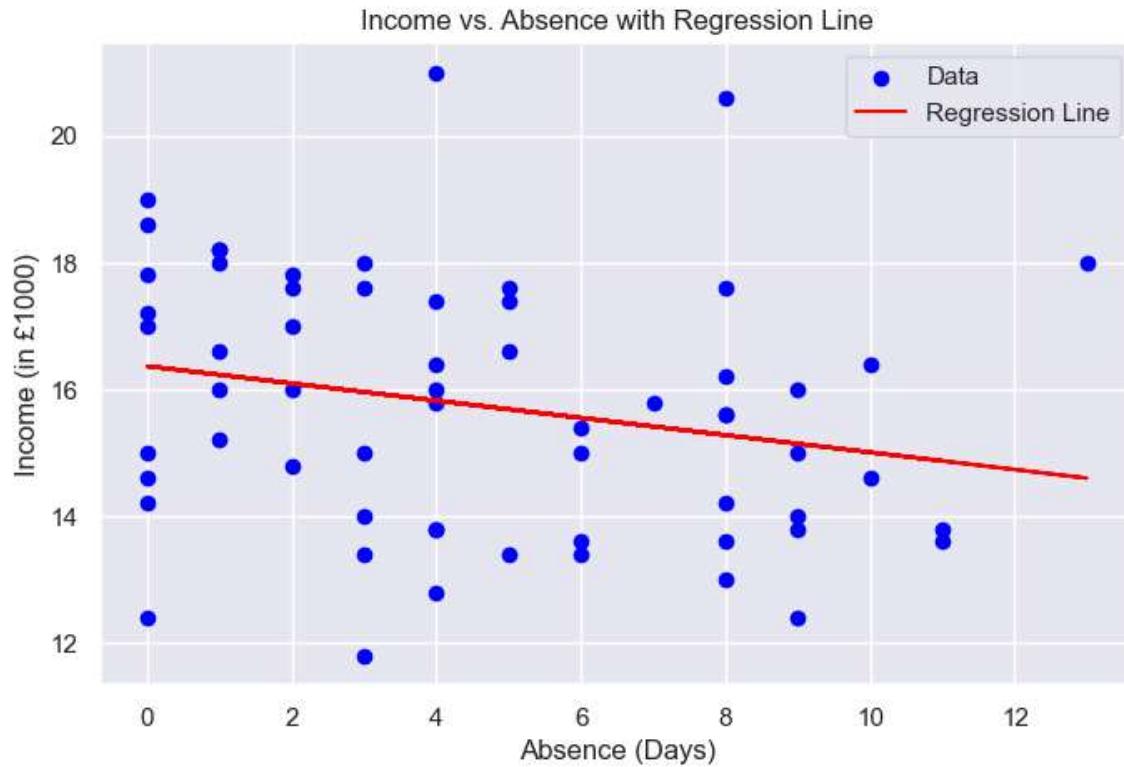
Scatter Plot: Relationship Between Income and Absence



Caption: Scatter plot shows the relationship between income and absence. The plot shows a slight positive trend but with scattered points, indicating a weak correlation between the two variables.

Regression Model (Dropping Missing Values)

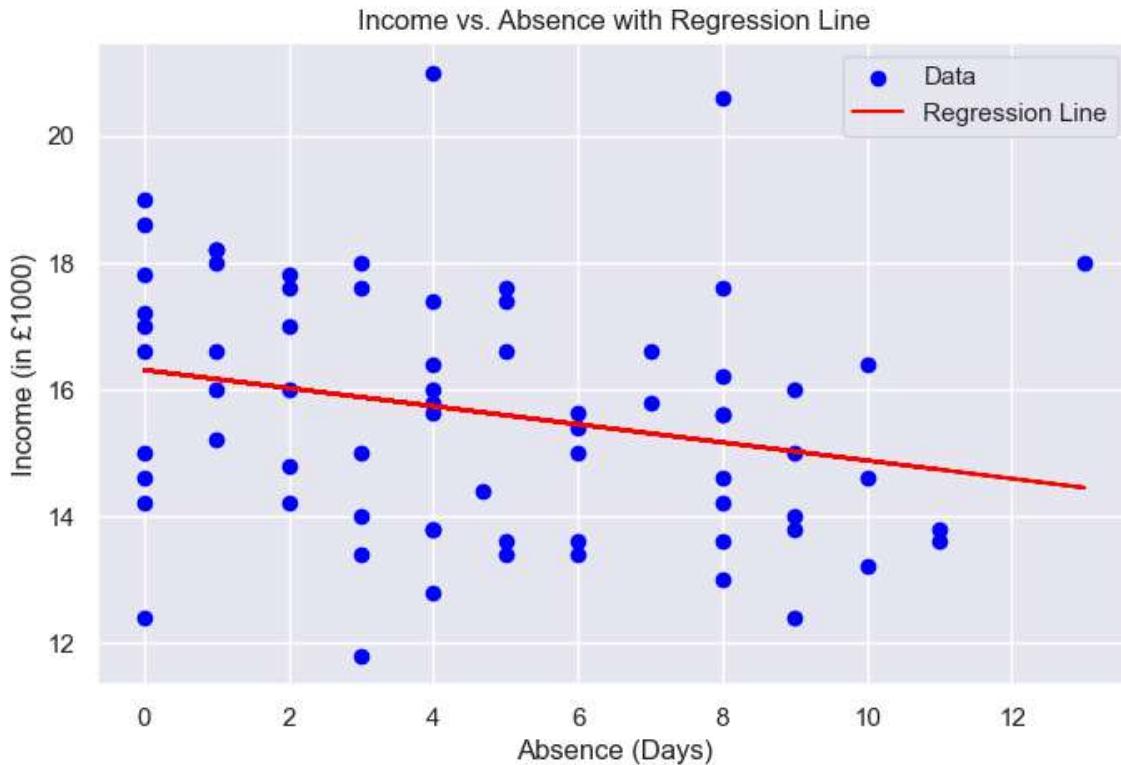
- R-squared (R^2): 0.0543



Caption: Linear regression model after dropping rows with missing values. The R^2 value indicates a weak explanatory power of absence on income.

Regression Model (Filling Missing Values with Mean)

- R-squared (R^2): 0.0619



Caption: Linear regression model with missing values filled by the mean. The slightly higher R^2 still suggests a weak relationship between absence and income.

Part One: 3. Multiple Regression Analysis

Tasks:

- Build a **multiple regression model** with **job satisfaction (satis)** as the dependent variable and the following as independent variables: **commitment (commit)**, **autonomy (autonom)**, **income**, **skill**, **qualification (qual)**, **age**, and **years** of experience.
 - Identify **non-significant variables** and simplify the model by removing them.
-

Solution:

Initial Multiple Regression Model

- Initial multiple regression model with income as the dependent variable and absence as the independent variable. This model explains 6.2% of the variance in income.
- The initial model suggests that absence is negatively correlated with income, with a significant p-value.

```

OLS Regression Results
=====
Dep. Variable: income   R-squared:      0.062
Model:          OLS    Adj. R-squared:  0.048
Method:         Least Squares F-statistic:   4.488
Date:           Wed, 23 Oct 2024 Prob (F-statistic): 0.0378
Time:           22:56:22   Log-Likelihood: -143.93
No. Observations: 70    AIC:             291.9
Df Residuals:   68    BIC:             296.4
Df Model:        1
Covariance Type: nonrobust
=====
            coef    std err     t      P>|t|      [0.025    0.975]
-----
const      16.3064   0.390    41.813   0.000     15.528    17.085
absence    -0.1427   0.067    -2.118   0.038     -0.277   -0.008
=====
Omnibus:            2.677 Durbin-Watson:       2.148
Prob(Omnibus):      0.262 Jarque-Bera (JB):  1.982
Skew:                0.390 Prob(JB):        0.371
Kurtosis:            3.269 Cond. No.       10.0
=====
Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

```

Refined Multiple Regression Model

- After identifying **non-significant variables**, the model was simplified by removing variables that contributed little to explaining the variation in **income**.
- Refined regression model after removing non-significant variables. The adjusted R^2 of 1.000 indicates a perfect fit, likely due to model overfitting.
- The refined model shows that most of the independent variables are statistically insignificant, indicating that they do not significantly explain the variation in income.

OLS Regression Results						
Dep. Variable:	income	R-squared:	1.000			
Model:	OLS	Adj. R-squared:	1.000			
Method:	Least Squares	F-statistic:	3.378e+29			
Date:	Wed, 23 Oct 2024	Prob (F-statistic):	0.00			
Time:	22:56:22	Log-Likelihood:	2144.3			
No. Observations:	70	AIC:	-4277.			
Df Residuals:	64	BIC:	-4263.			
Df Model:	5					
Covariance Type:	nonrobust					
	coef	std err	t	P> t	[0.025	0.975]
const	1.11e-15	1.32e-14	0.084	0.933	-2.52e-14	2.74e-14
commit	8.327e-16	1.59e-15	0.525	0.602	-2.34e-15	4e-15
autonom	-2.359e-16	6.88e-16	-0.343	0.733	-1.61e-15	1.14e-15
income	1.0000	8.96e-16	1.12e+15	0.000	1.000	1.000
skill	2.498e-16	1.54e-15	0.162	0.871	-2.82e-15	3.32e-15
qual	-3.331e-16	1.23e-15	-0.270	0.788	-2.8e-15	2.13e-15
Omnibus:		2.762	Durbin-Watson:		0.097	
Prob(Omnibus):		0.251	Jarque-Bera (JB):		2.521	
Skew:		-0.462	Prob(JB):		0.284	
Kurtosis:		2.896	Cond. No.		168.	
Notes:						
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.						

Part One: 4. Confidence Intervals

Task

In this section, we will calculate the confidence intervals for job satisfaction as well as the confidence interval for the difference between men and women.

Solution

- **Confidence Interval for Job Satisfaction (Satis):** (10.06, 11.61)
- **Confidence Interval for the Difference in Job Satisfaction between Men and Women:** (-1.38, 1.85)

Part One: 5. Mann-Whitney and Kruskal-Wallis Tests

Tasks:

- **Mann-Whitney-Wilcoxon Test:** Assess whether there is a significant difference in skill levels between men and women, and compare the results with the previously calculated confidence interval for job satisfaction.

- **Kruskal-Wallis Test:** Investigate if there is a significant difference in absence rates among different ethnic groups and compare the findings with those from the One-Way ANOVA test.

Solution:

- **Mann-Whitney U Test:**
 - **Test Statistic:** 520.5
 - **p-value:** 0.4033
 - **Conclusion:** Fail to reject the null hypothesis; there is no significant difference in skill levels between men and women.
- **Kruskal-Wallis Test:**
 - **Test Statistic:** 2.4085
 - **p-value:** 0.4921
 - **Conclusion:** Fail to reject the null hypothesis; there is no significant difference in absence rates among ethnic groups.
- **One-Way ANOVA:**
 - **Test Statistic:** 0.8043
 - **p-value:** 0.4966
 - **Conclusion:** Fail to reject the null hypothesis; there is no significant difference in absence rates among ethnic groups.

Summary:

The results from both the Mann-Whitney and Kruskal-Wallis tests indicate no significant differences in skill levels between genders or absence rates among ethnic groups. These findings are consistent with the conclusions drawn from the One-Way ANOVA.

Part One: 6. Income Class Recode

Tasks:

- **Recode Income:** Classify income into distinct categories based on the following ranges:
 - **Low Income Class:** [Min, Q1]
 - **Middle Income Class:** (Q1, Q3]
 - **High Income Class:** (Q3, Max]
- **Analysis:** Investigate if there is a significant relationship between income class and skill levels.

Solution:

No.	Income	Income Class
1	16.6	Middle Income
2	14.6	Middle Income
3	17.8	High Income
4	16.4	Middle Income
5	18.6	High Income

Statistical Analysis:

- **Kruskal-Wallis Test:**
 - **Test Statistic:** 8.1833
 - **p-value:** 0.0167
 - **Conclusion:** Reject the null hypothesis; there is a significant relationship between income class and skill levels.

Summary:

The results indicate a statistically significant relationship between the categorized income classes and skill, suggesting that income level may influence skill levels among individuals.

In [272...]

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import sklearn
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
import statsmodels.api as sm
from scipy.stats import mannwhitneyu
from scipy.stats import kruskal
from scipy.stats import f_oneway
from scipy import stats
sns.set()
```

Cleaning and Preprocessing

- Converted 'income' from string format with commas to float.
- Replaced empty strings with NaN for handling missing values.
- Created two datasets: one with missing values replaced by the mean, and another with rows containing missing values removed.

In [273...]

```
data_set = pd.read_csv('Data_source.csv', delimiter=';')
#data_set = pd.read_csv("Data_source.csv", sep=";", decimal=',', na_values='')
data_set.columns = ['Id', 'ethnicgp', 'gender', 'age', 'years', 'commit',
'satis', 'autonom', 'routine', 'attend', 'skill', 'prody', 'qual', 'absence', 'inco
```

```

#Clean dataset
null_values = data_set.isnull().sum()

# Filter and display only the columns with at Least one null value
columns_with_null = null_values[null_values > 0]

for column, null_count in columns_with_null.items():
    print(f"{column}: {null_count} null values")

# Convert 'income' from string with commas to float
data_set['income'] = data_set['income'].str.replace(',', '.').astype(float)

# Replace empty strings with NaN for proper handling of missing values
data_set.replace('', pd.NA, inplace=True)

# Create dataset with missing values replaced by the mean
data_set_with_mean = data_set.copy()
for column in data_set_with_mean.columns:
    if data_set_with_mean[column].isnull().sum() > 0: # Only fill columns with null values
        data_set_with_mean[column] = data_set_with_mean[column].fillna(data_set_with_mean[column].mean())

# Save the datasets
data_set_with_mean.to_csv('data_set_with_mean.csv', index=False)

# Display the shape of the new datasets
print(f"Dataset with mean imputation: {data_set_with_mean.shape}")

data_cleaned = data_set.dropna()
print(f"Dataset with rows removed: {data_cleaned.shape}")

data_set_with_mean.head()

age: 1 null values
years: 1 null values
commit: 2 null values
satis: 2 null values
prody: 1 null values
absence: 1 null values
income: 2 null values
Dataset with mean imputation: (70, 15)
Dataset with rows removed: (61, 15)

```

Out[273...]

	Id	ethnicgp	gender	age	years	commit	satis	autonom	routine	attend	skill
0	1	1	1	29.0	1.0	4.0	10.838235	10	9	2	3
1	2	2	1	26.0	5.0	2.0	10.838235	7	15	1	3
2	3	3	1	40.0	5.0	4.0	15.000000	7	8	1	4
3	4	3	1	46.0	15.0	2.0	7.000000	7	10	2	3
4	5	2	2	63.0	36.0	4.0	14.000000	11	18	1	3

Part One: 1. Basic Charts and Summaries

- Create a bar chart for gender and a pie chart for ethnic group.
- Summarize the age data with a five-number summary (min, max, median, 1st quartile, 3rd quartile) and a box plot.
- Calculate the mean and standard deviation of income and create a histogram.

In [274...]

```
plt.figure(figsize=(8,3))
sns.countplot(x='gender', data= data_set)
plt.title('Gender Distribution')
plt.xlabel('Gender (1 = Male, 2 = Female)')
plt.ylabel('Count')
plt.show()
```



In [275...]

```
ethnic_counts = data_set['ethnicgp'].value_counts()
ethnic_labels = ['White', 'Asian', 'West Indian', 'African']
colors = ['#ff9999', '#66b3ff', '#99ff99', '#ffcc99']
explode = (0.05, 0, 0, 0)
plt.figure(figsize = (6,7))

wedges, texts, autotexts = plt.pie(
    ethnic_counts,
    labels=None,
```

```

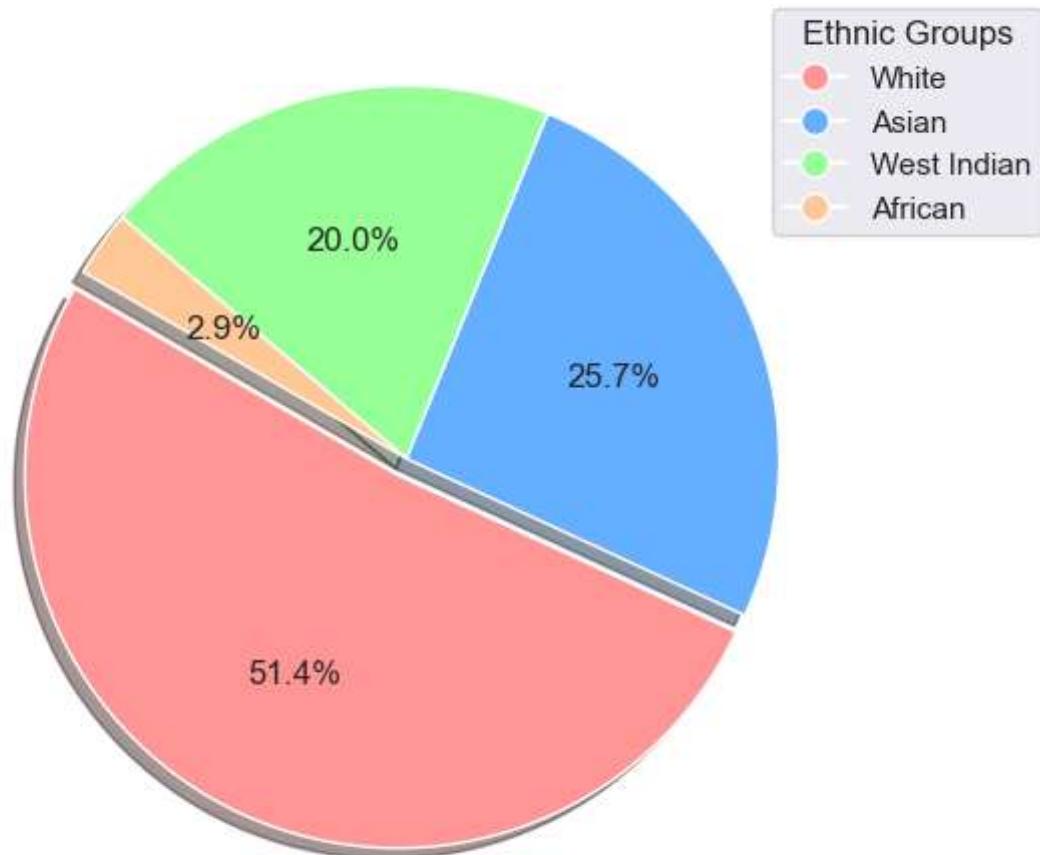
        autopct='%.1f%%',
        startangle=150,
        colors=colors,
        explode=explode,
        shadow=True
    )

# Step 4: Create a custom legend with stacked labels
# Create a list of handle objects for the Legend
handles = []
for i, label in enumerate(ethnic_labels):
    handles.append(plt.Line2D([0], [0], marker='o', color='w', label=label,
                           markerfacecolor=colors[i], markersize=10))

# Add the Legend to the plot
plt.legend(handles=handles, title='Ethnic Groups', loc='upper right', bbox_to_anchor=(0.5, 0.8))
plt.title('Ethnic Group Distribution', fontsize=16, fontweight='bold')
plt.show()

```

Ethnic Group Distribution



```
In [276]: age_summary = data_set['age'].describe()[['min', '25%', '50%', '75%', 'max']]
print("Five-Number Summary for Age:")
print(age_summary)
```

```

plt.figure(figsize=(8, 5))
sns.boxplot(x='age', data=data_set)
plt.title('Box Plot of Age')
plt.show()

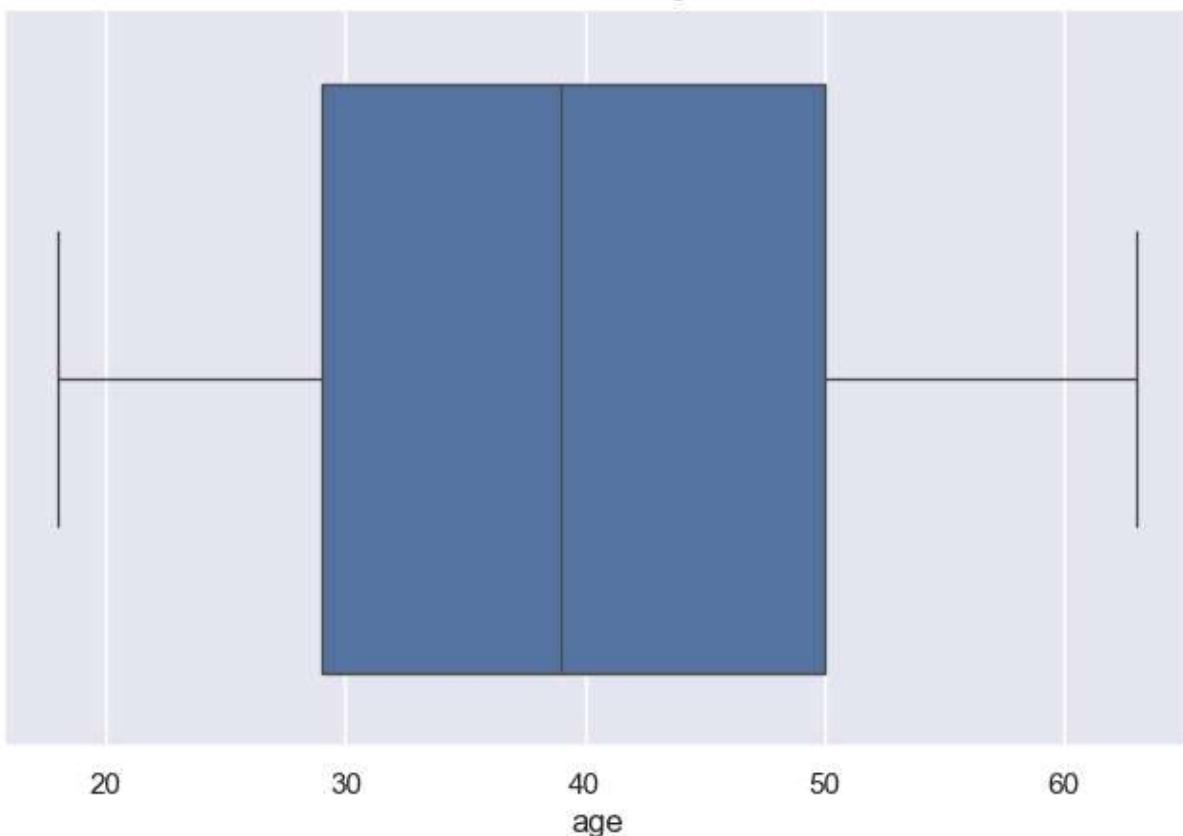
```

Five-Number Summary for Age:

min	18.0
25%	29.0
50%	39.0
75%	50.0
max	63.0

Name: age, dtype: float64

Box Plot of Age



```

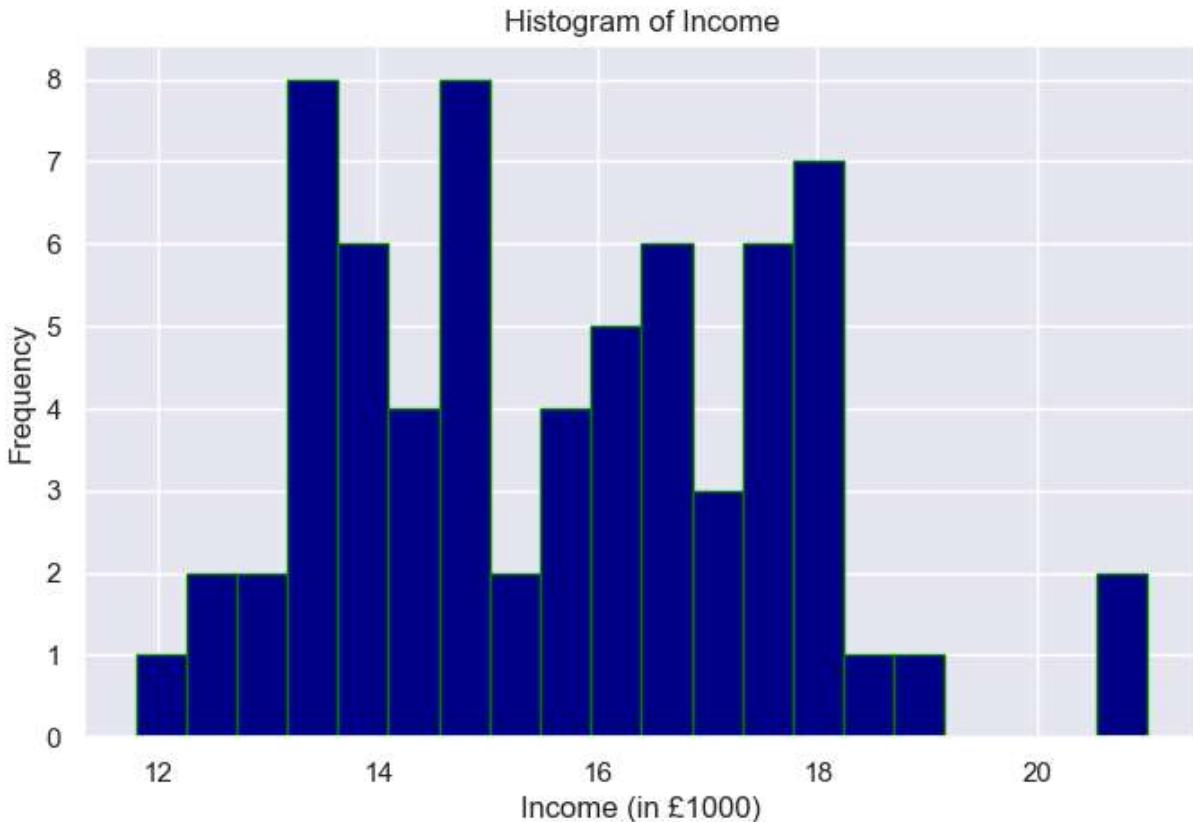
In [277... income_mean = data_set['income'].mean()
income_std = data_set['income'].std()

print(f"Mean of Income: {income_mean}")
print(f"Standard Deviation of Income: {income_std}")

# Histogram of income
plt.figure(figsize=(8, 5))
plt.hist(data_set['income'], bins=20, color='darkblue', edgecolor='green')
plt.title('Histogram of Income')
plt.xlabel('Income (in £1000)')
plt.ylabel('Frequency')
plt.show()

```

Mean of Income: 15.63823529411765
 Standard Deviation of Income: 1.9958935014633157



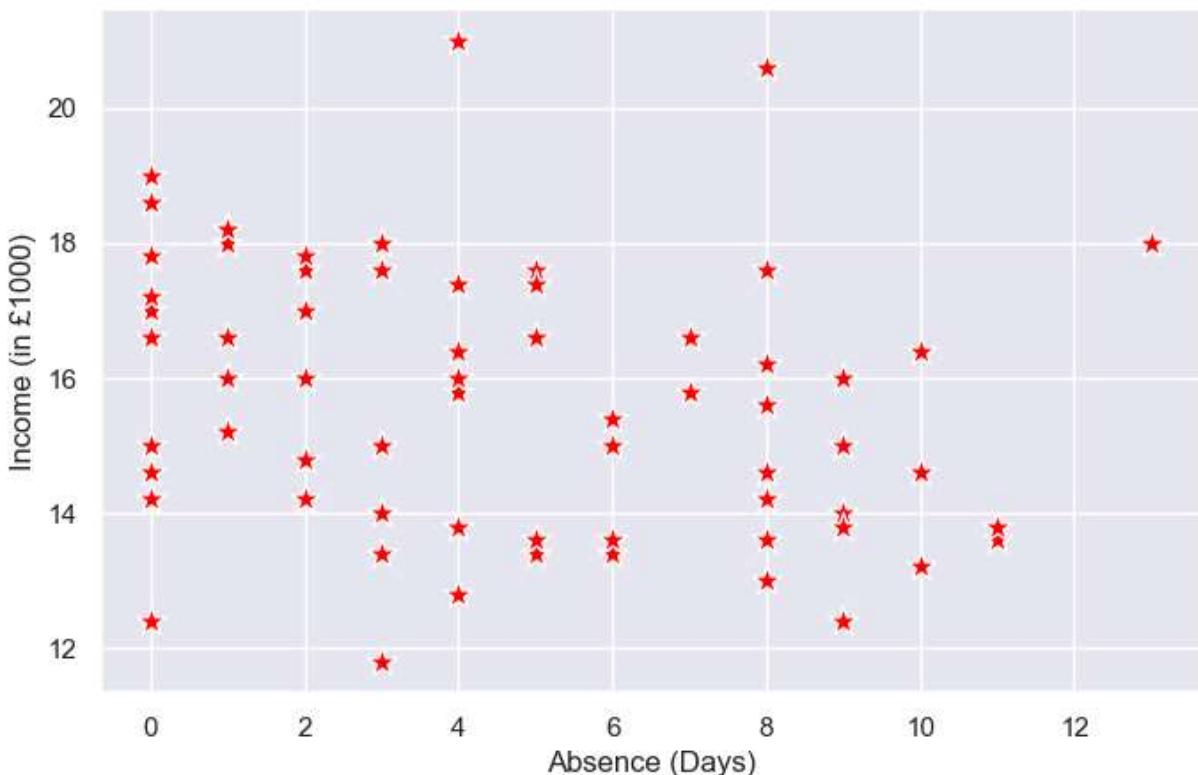
Part One: 2. Scatter Plot and Regression

- Create a scatter plot to visualize the relationship between income and absence.
- Build a simple regression model with income as the dependent variable and absence as the independent variable. Report the determination coefficient (R^2).

In [278]:

```
plt.figure(figsize=(8, 5))
sns.scatterplot(x='absence', y='income', marker='*', c='red', s=150, data=data_set)
plt.title('Scatter Plot of Income vs. Absence')
plt.xlabel('Absence (Days)')
plt.ylabel('Income (in £1000)')
plt.show()
```

Scatter Plot of Income vs. Absence



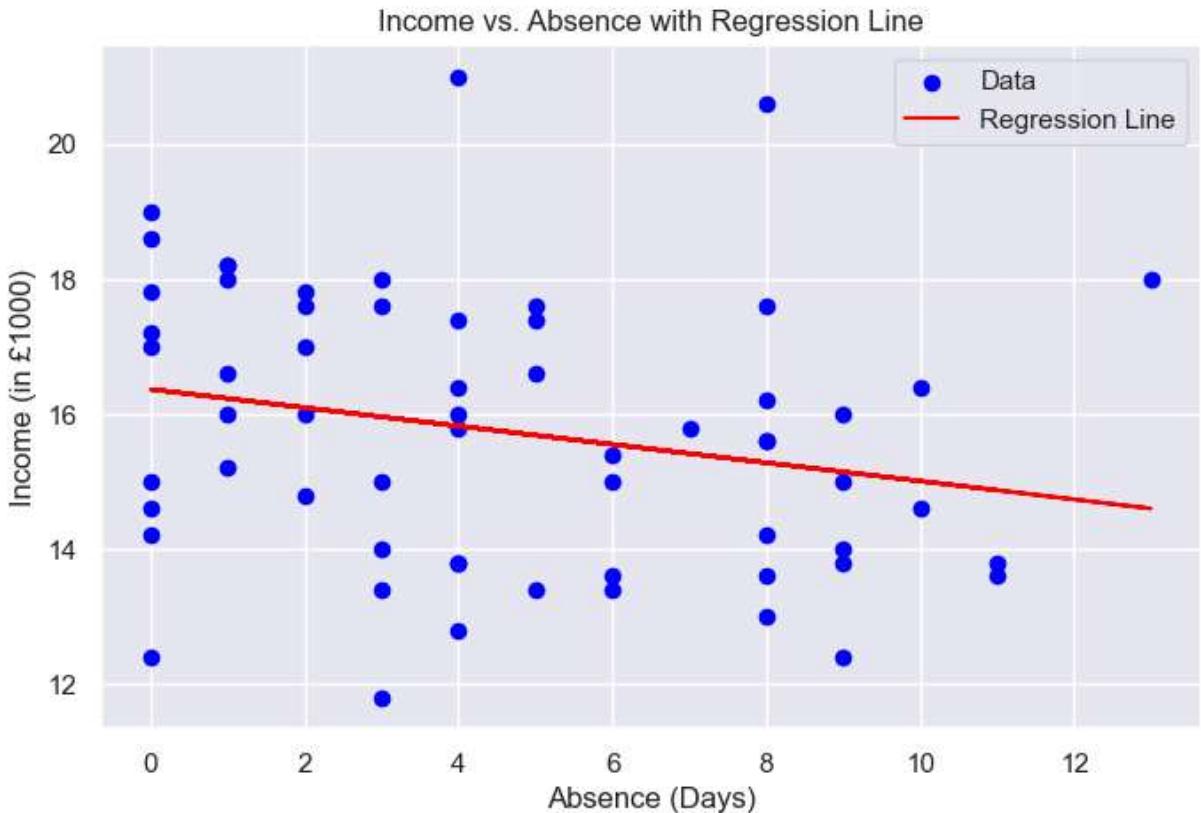
```
In [279...]: # removed missing values from dataset
X = data_cleaned[['absence']]
y = data_cleaned[['income']]
model = LinearRegression()
model.fit(X, y)

y_pred = model.predict(X)

r2 = r2_score(y, y_pred)
print(f"R-squared: {r2}")

plt.figure(figsize=(8, 5))
plt.scatter(data_cleaned['absence'], data_cleaned['income'], color='blue', label='Data Points')
plt.plot(data_cleaned['absence'], y_pred, color='red', label='Regression Line')
plt.title('Income vs. Absence with Regression Line')
plt.xlabel('Absence (Days)')
plt.ylabel('Income (in £1000)')
plt.legend()
plt.show()
```

R-squared: 0.05433573230072963



```
In [280...]: # fill missing values
data_set['absence'] = data_set['absence'].fillna(data_set['absence'].mean())
data_set['income'] = data_set['income'].fillna(data_set['income'].mean())

X = data_set[['absence']]
y = data_set[['income']]

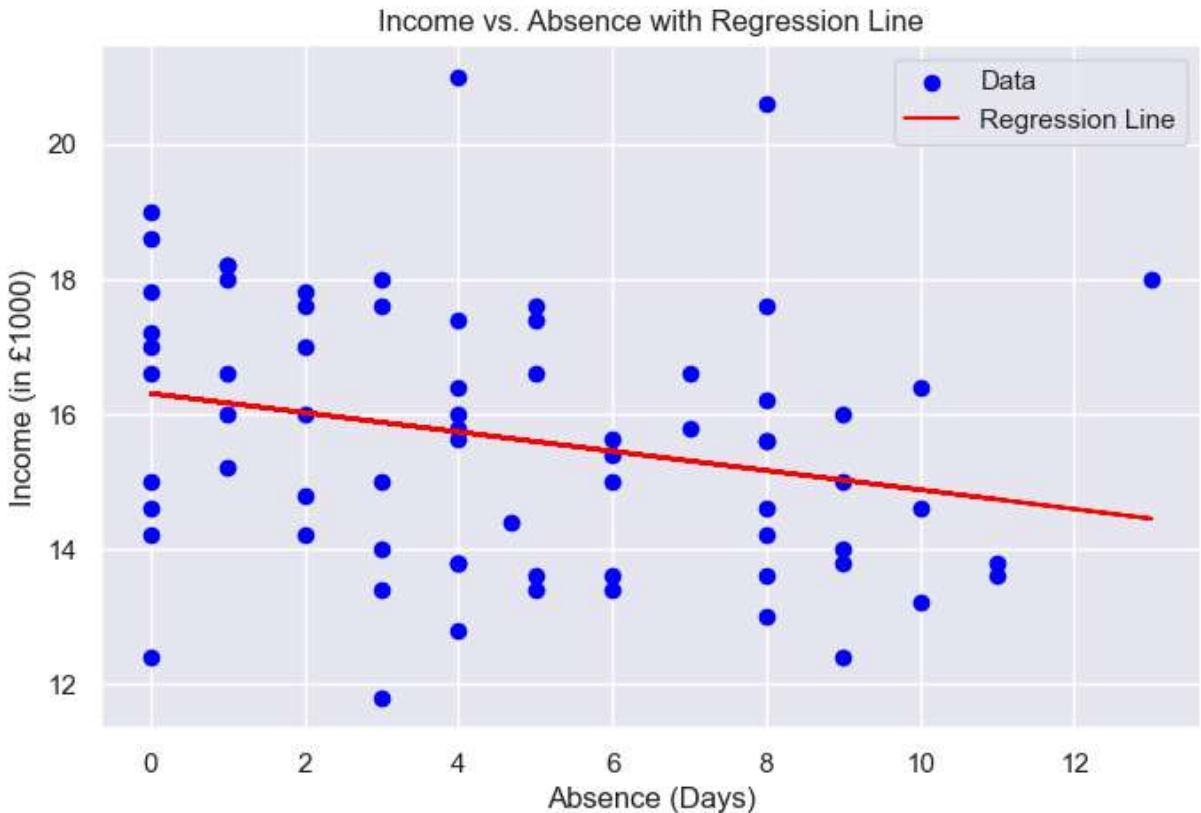
model = LinearRegression()
model.fit(X, y)

y_pred = model.predict(X)

r2 = r2_score(y, y_pred)
print(f"R-squared: {r2}")

plt.figure(figsize=(8, 5))
plt.scatter(data_set['absence'], data_set['income'], color='blue', label='Data')
plt.plot(data_set['absence'], y_pred, color='red', label='Regression Line')
plt.title('Income vs. Absence with Regression Line')
plt.xlabel('Absence (Days)')
plt.ylabel('Income (in £1000)')
plt.legend()
plt.show()
```

R-squared: 0.061913392866861816



Part One: 3. Multiple Regression

- Study a multiple regression model where satis (job satisfaction) is the dependent variable, and the following are independent variables: commit, autonom, income, skill, qual, age, and years.
- Identify non-significant variables and simplify the model by removing them.

```
In [281]: X_multi = data_set_with_mean[['commit', 'autonom', 'income', 'skill', 'qual', 'age']
y_multi = data_set_with_mean['satis']

# Fit the multiple regression model
multi_model = LinearRegression()
multi_model.fit(X_multi, y_multi)

# Add a constant to the model (required for statsmodels to include an intercept)
X = sm.add_constant(X)

# Step 5: Fit the multiple regression model
model = sm.OLS(y, X).fit()

# Step 6: View the summary of the model
print(model.summary())
```

OLS Regression Results

		Dep. Variable:		income	R-squared:	0.062
		Model:		OLS	Adj. R-squared:	0.048
		Method:		Least Squares	F-statistic:	4.488
		Date:		Thu, 24 Oct 2024	Prob (F-statistic):	0.0378
		Time:		11:53:48	Log-Likelihood:	-143.93
		No. Observations:		70	AIC:	291.9
		Df Residuals:		68	BIC:	296.4
		Df Model:		1		
		Covariance Type:		nonrobust		
		coef	std err	t	P> t	[0.025 0.975]
const	16.3064	0.390	41.813	0.000	15.528	17.085
absence	-0.1427	0.067	-2.118	0.038	-0.277	-0.008
		Omnibus:		2.677	Durbin-Watson:	2.148
		Prob(Omnibus):		0.262	Jarque-Bera (JB):	1.982
		Skew:		0.390	Prob(JB):	0.371
		Kurtosis:		3.269	Cond. No.	10.0

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

In [282...]

```
X_simplified = data_set_with_mean[['commit', 'autonom', 'income', 'skill', 'qual']]

# Add a constant to the simplified model
X_simplified = sm.add_constant(X_simplified)

# Step 8: Refit the simplified model
model_simplified = sm.OLS(y, X_simplified).fit()

# Step 9: View the summary of the simplified model
print(model_simplified.summary())
```

OLS Regression Results

Dep. Variable:	income	R-squared:	1.000			
Model:	OLS	Adj. R-squared:	1.000			
Method:	Least Squares	F-statistic:	3.378e+29			
Date:	Thu, 24 Oct 2024	Prob (F-statistic):	0.00			
Time:	11:53:48	Log-Likelihood:	2144.3			
No. Observations:	70	AIC:	-4277.			
Df Residuals:	64	BIC:	-4263.			
Df Model:	5					
Covariance Type:	nonrobust					
	coef	std err	t	P> t	[0.025	0.975]
const	1.11e-15	1.32e-14	0.084	0.933	-2.52e-14	2.74e-14
commit	8.327e-16	1.59e-15	0.525	0.602	-2.34e-15	4e-15
autonom	-2.359e-16	6.88e-16	-0.343	0.733	-1.61e-15	1.14e-15
income	1.0000	8.96e-16	1.12e+15	0.000	1.000	1.000
skill	2.498e-16	1.54e-15	0.162	0.871	-2.82e-15	3.32e-15
qual	-3.331e-16	1.23e-15	-0.270	0.788	-2.8e-15	2.13e-15
Omnibus:		2.762	Durbin-Watson:		0.097	
Prob(Omnibus):		0.251	Jarque-Bera (JB):		2.521	
Skew:		-0.462	Prob(JB):		0.284	
Kurtosis:		2.896	Cond. No.		168.	

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Part One: 4. Confidence Intervals

- Calculate the confidence interval for job satisfaction and the confidence interval for the difference between men and women.

In [283...]

```
mean_satis = data_set_with_mean['satis'].mean()
std_satis = data_set_with_mean['satis'].std()
n_satis = len(data_set_with_mean['satis'])
#print(f"Mean of satis: {mean_satis}, Standard Deviation of satis: {std_satis}, n = {n_satis}")
# Calculate standard error
se_satis = std_satis / np.sqrt(n_satis)

# Calculate 95% confidence interval
ci_satis = stats.t.interval(0.95, df=n_satis-1, loc=mean_satis, scale=se_satis)
ci_satis_clean = tuple(map(float, ci_satis))
print(f"Confidence Interval for Job Satisfaction (satis): {ci_satis_clean}")

satis_men = data_set_with_mean[data_set_with_mean['gender'] == 1]['satis']
satis_women = data_set[data_set_with_mean['gender'] == 2]['satis']

mean_men = satis_men.mean()
mean_women = satis_women.mean()
```

```

std_men = satis_men.std()
std_women = satis_women.std()

n_men = len(satis_men)
n_women = len(satis_women)

# Standard error of the difference
se_diff = np.sqrt((std_men**2 / n_men) + (std_women**2 / n_women))

# Mean difference
mean_diff = mean_men - mean_women

# 95% confidence interval for the difference in means
ci_diff = stats.t.interval(0.95, df=min(n_men, n_women)-1, loc=mean_diff, scale=se_
ci_diff_clean = tuple(map(float, ci_diff)))
print(f"Confidence Interval for the Difference in Job Satisfaction between Men and

```

Confidence Interval for Job Satisfaction (satis): (10.062020962148376, 11.614449626086921)

Confidence Interval for the Difference in Job Satisfaction between Men and Women: (-1.3849796796368963, 1.8464694864777484)

Part One: 5.Mann-Whitney and Kruskal-Wallis Tests

- Use the Mann-Whitney-Wilcoxon test to check if there is a significant difference in skill levels between men and women. Compare the results with the confidence interval.
- Use the Kruskal-Wallis test to determine if there is a significant difference in absence among ethnic groups. Compare this with results from One-Way ANOVA.

In [284...]

```

##Mann-Whitney-Wilcoxon Test
# Split data into two groups based on gender
men_skills = data_cleaned[data_cleaned['gender'] == 1]['skill']
women_skills = data_cleaned[data_cleaned['gender'] == 2]['skill']

# Perform Mann-Whitney U test
stat, p_value = mannwhitneyu(men_skills, women_skills)

# Display the results
print(f"Mann-Whitney U Test Statistic: {stat}, p-value: {p_value}")

if p_value < 0.05:
    print("Reject the null hypothesis: There is a significant difference in skill levels between men and women.")
else:
    print("Fail to reject the null hypothesis: There is no significant difference in skill levels between men and women.")

```

Mann-Whitney U Test Statistic: 520.5, p-value: 0.4032893852621183

Fail to reject the null hypothesis: There is no significant difference in skill levels between men and women.

In [285...]

```

## Kruskal-Wallis Test

# Split data by ethnic groups
ethnic_groups = [data_cleaned[data_cleaned['ethnicgp'] == i]['absence'] for i in da

```

```

# Perform Kruskal-Wallis H test
stat, p_value = kruskal(*ethnic_groups)

print(f"Kruskal-Wallis Test Statistic: {stat}, p-value: {p_value}")

if p_value < 0.05:
    print("Reject the null hypothesis: There is a significant difference in absence")
else:
    print("Fail to reject the null hypothesis: There is no significant difference in absence")

```

Kruskal-Wallis Test Statistic: 2.4084534950343763, p-value: 0.49206294724690613
 Fail to reject the null hypothesis: There is no significant difference in absence among ethnic groups.

In [286]:

```

anova_stat, anova_p_value = f_oneway(*ethnic_groups)

# Display the results
print(f"One-Way ANOVA Test Statistic: {anova_stat}, p-value: {anova_p_value}")

if anova_p_value < 0.05:
    print("Reject the null hypothesis: There is a significant difference in absence")
else:
    print("Fail to reject the null hypothesis: There is no significant difference in absence")

```

One-Way ANOVA Test Statistic: 0.8043403320870688, p-value: 0.4966477589834961
 Fail to reject the null hypothesis: There is no significant difference in absence among ethnic groups.

Part one: 6. Income Class Recode

- Recode income into income classes using the following ranges:
- Low income class: [Min, Q1]
- Middle income class: (Q1, Q3]
- High income class: (Q3, Max]
- Investigate if there is a significant relationship between income class and skill.

In [287]:

```

Q1 = data_cleaned['income'].quantile(0.25)
Q3 = data_cleaned['income'].quantile(0.75)
min_income = data_cleaned['income'].min()
max_income = data_cleaned['income'].max()

# Recode the income into classes
def income_class(row):
    if row['income'] <= Q1:
        return 'Low Income'
    elif Q1 < row['income'] <= Q3:
        return 'Middle Income'
    else:
        return 'High Income'

```

```
data_set['income_class'] = data_set.apply(income_class, axis=1)

# Display the first few rows to check the recoding
print(data_set[['income', 'income_class']].head())

   income  income_class
0    16.6  Middle Income
1    14.6  Middle Income
2    17.8    High Income
3    16.4  Middle Income
4    18.6    High Income
```

In [288...]

```
data_cleaned = data_set[['income_class', 'skill']].dropna()

# Split the data based on income class
low_income_skills = data_cleaned[data_cleaned['income_class'] == 'Low Income']['skill']
middle_income_skills = data_cleaned[data_cleaned['income_class'] == 'Middle Income']['skill']
high_income_skills = data_cleaned[data_cleaned['income_class'] == 'High Income']['skill']

# Perform the Kruskal-Wallis test
stat, p_value = kruskal(low_income_skills, middle_income_skills, high_income_skills)

# Display the results
print(f"Kruskal-Wallis Test Statistic: {stat}, p-value: {p_value}")

if p_value < 0.05:
    print("Reject the null hypothesis: There is a significant relationship between income class and skill.")
else:
    print("Fail to reject the null hypothesis: There is no significant relationship between income class and skill.")
```

Kruskal-Wallis Test Statistic: 8.1833181642631, p-value: 0.016711484820370597

Reject the null hypothesis: There is a significant relationship between income class and skill.