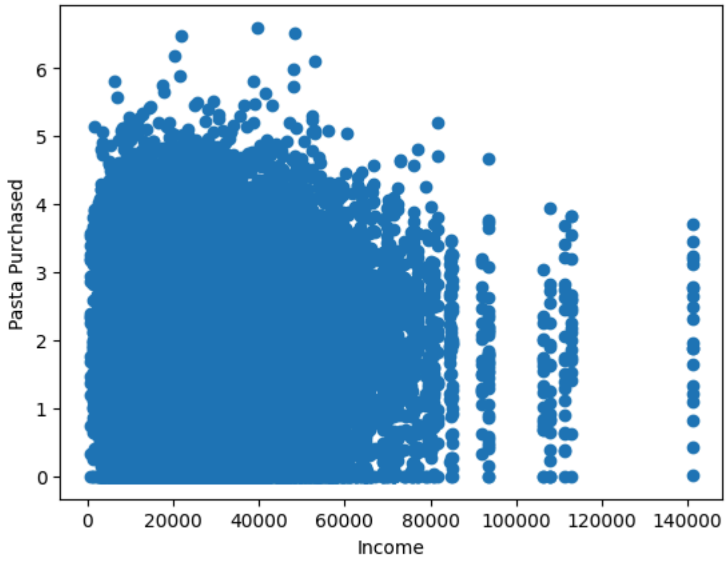
```
from google.colab import files
uploaded = files.upload()
```

Choose files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
import pandas as pd

df = pd.read_csv('_a979e0f25f27452188ad26384c784407_PASTAPURCHASE_EDITED.csv', sep=';')
df.head()
```

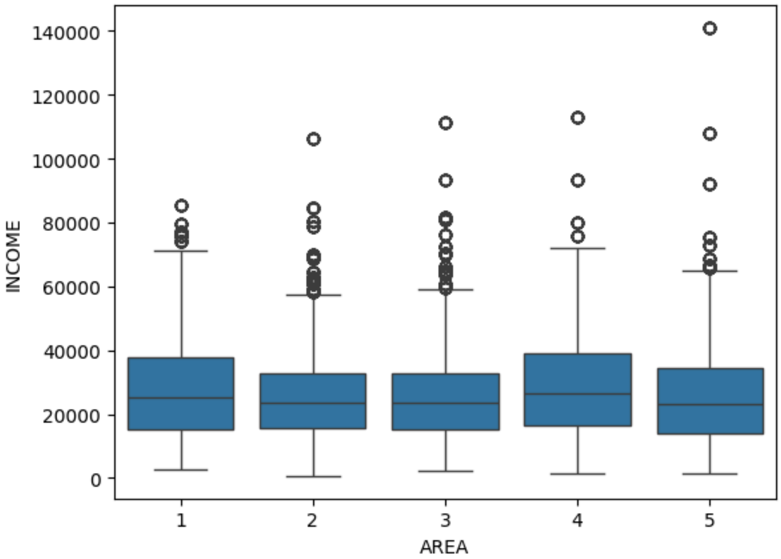
	HHID	TIME	PASTA	EXPOS	AGE	INCOME	AREA
0	1	1	0.939444	1	61.710758	25186.798772	3
1	1	2	2.560969	2	61.710758	25186.798772	3
2	1	3	0.901123	0	61.710758	25186.798772	3
3	1	4	1.916530	1	61.710758	25186.798772	3
4	1	5	1.548751	0	61.710758	25186.798772	3

```
df.describe()
df.isna().sum()
df['PASTA'].mean(), df['PASTA'].std()
import matplotlib.pyplot as plt

plt.scatter(df['INCOME'], df['PASTA'])
plt.xlabel('Income')
plt.ylabel('Pasta Purchased')
plt.show()
import statsmodels.formula.api as smf
model = smf.ols('PASTA ~ EXPOS + AGE + INCOME', data=df).fit()
model.summary()
df.to_csv('cleaned_pasta.csv', index=False)
files.download('cleaned_pasta.csv')
```

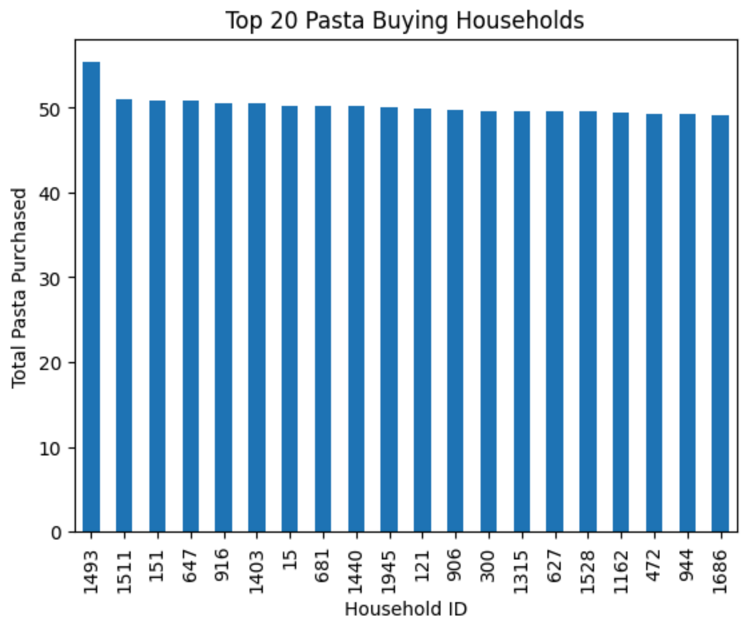


```
df.head()
df.info()
df.describe()
df.isna().sum()
df.dtypes
df.duplicated().sum()
poorest = df.loc[df['INCOME'].idxmin()]
wealthiest = df.loc[df['INCOME'].idxmax()]
poorest_area = poorest['AREA']
wealthiest_area = wealthiest['AREA']
df[df['INCOME'] == df['INCOME'].min()]
df[df['INCOME'] == df['INCOME'].max()]
import seaborn as sns
sns.boxplot(x='AREA', y='INCOME', data=df)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 40000 entries, 0 to 39999
Data columns (total 7 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   HHID    40000 non-null  int64
 1   TIME    40000 non-null  int64
 2   PASTA   40000 non-null  float64
 3   EXPOS   40000 non-null  int64
 4   AGE     40000 non-null  float64
 5   INCOME  40000 non-null  float64
 6   AREA    40000 non-null  int64
dtypes: float64(3), int64(4)
memory usage: 2.1 MB
<Axes: xlabel='AREA', ylabel='INCOME'>
```



```
# total_pasta_per_household = sum(PASTA grouped by HHID)
# max_pasta = maximum of total_pasta_per_household
# Step 1: Group by household and sum pasta quantities
household_totals = df.groupby('HHID')['PASTA'].sum()

# Step 2: Find the maximum total pasta bought by any household
max_pasta = household_totals.max()

max_pasta
household_totals.idxmax()
household_totals.describe()
household_totals.sort_values(ascending=False).head(10)
import matplotlib.pyplot as plt

household_totals.sort_values(ascending=False).head(20).plot(kind='bar')
plt.xlabel('Household ID')
plt.ylabel('Total Pasta Purchased')
plt.title('Top 20 Pasta Buying Households')
plt.show()
```



```
# Filter households in area 4
area4_df = df[df['AREA'] == 4]

# Calculate average income
average_income_area4 = area4_df['INCOME'].mean()

average_income_area4
```

```
np.float64(29260.13313734934)
```

```
pasta_per_household = df.groupby("HHID")["PASTA"].sum().reset_index()
pasta_per_household.rename(columns={"PASTA": "TOTAL_PASTA"}, inplace=True)

# STEP 2 — Merge totals back into main dataset
df_merged = pd.merge(df, pasta_per_household, on="HHID", how="left")

# STEP 3 — Apply all three conditions:
# 1. Area = 2
# 2. Income > 20000
# 3. Total pasta > 30
filtered = df_merged[
    (df_merged["AREA"] == 2) &
    (df_merged["INCOME"] > 20000) &
    (df_merged["TOTAL_PASTA"] > 30)
]

# STEP 4 — Count unique households
answer = filtered["HHID"].nunique()

print("📌 Number of households meeting all conditions:", answer)
```

```
📌 Number of households meeting all conditions: 218
```

```
poorest_area = poorest['AREA']
wealthiest_area = wealthiest['AREA']

print(f"The poorest household is present in Area: {poorest_area}")
print(f"The wealthiest household is present in Area: {wealthiest_area}")
```

```
The poorest household is present in Area: 2.0
The wealthiest household is present in Area: 5.0
```

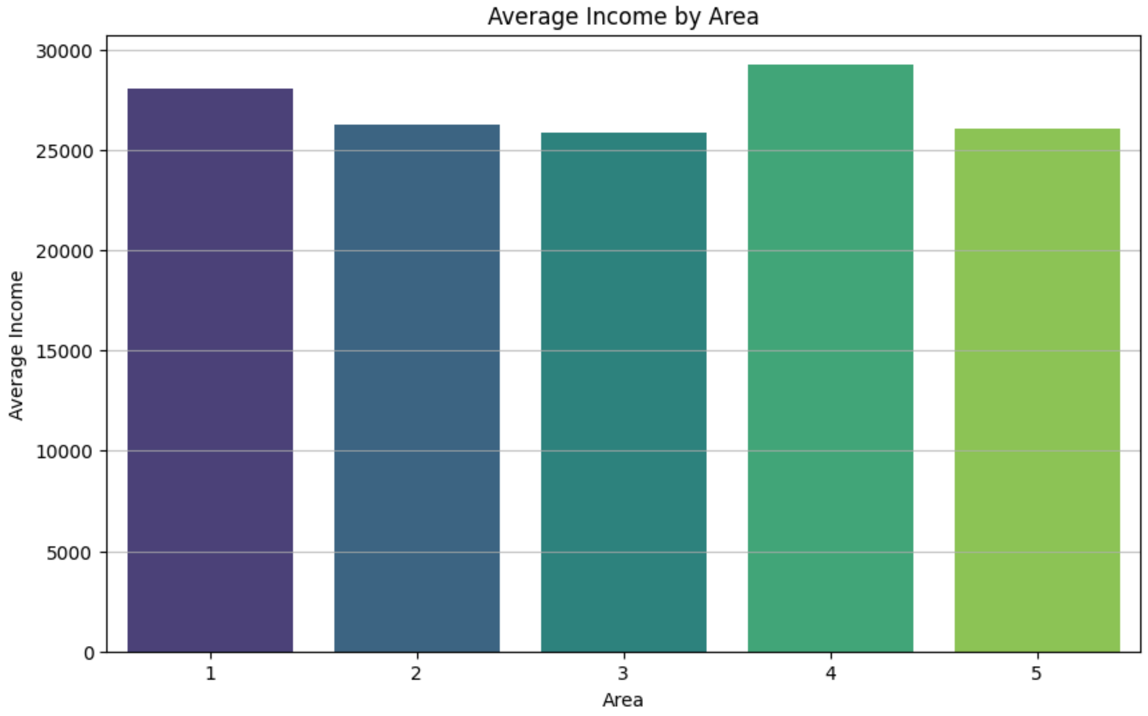
```
pasta_stats_by_hh_time = df.groupby(['HHID', 'TIME'])['PASTA'].agg(['mean', 'std'])
print("Mean and Standard Deviation of Pasta Purchased by Household and Time Unit:")
display(pasta_stats_by_hh_time.head())
```

```
Mean and Standard Deviation of Pasta Purchased by Household and Time Unit:
```

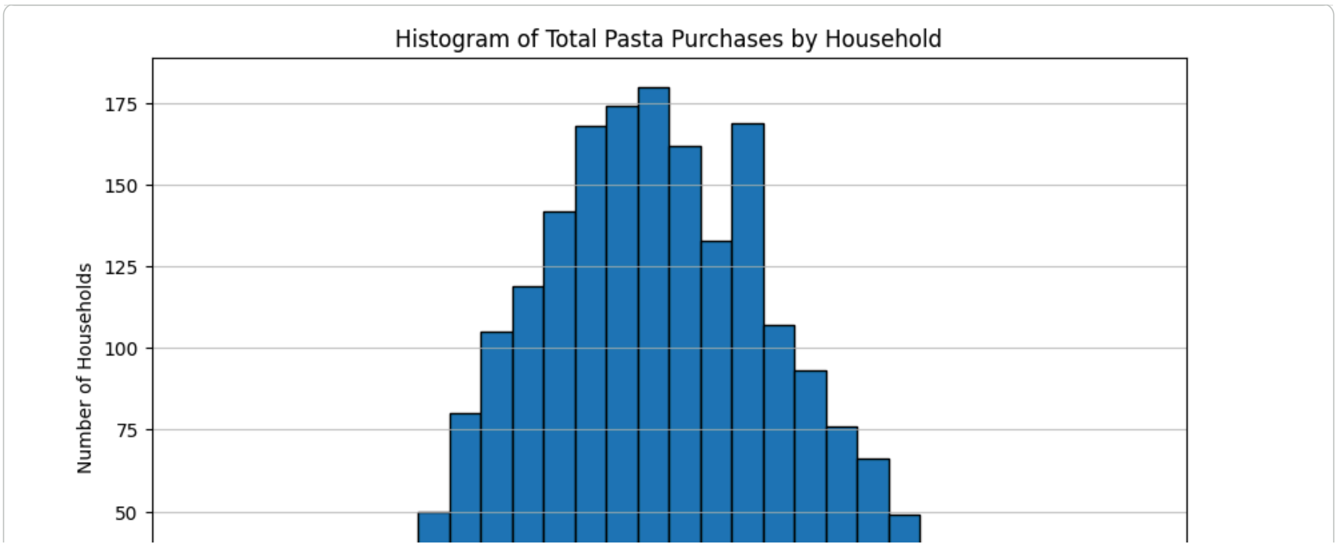
		mean	std
HHID	**TIME**		
1	**1**	0.939444	NaN
	2	2.560969	NaN
	3	0.901123	NaN
	4	1.916530	NaN
	5	1.548751	NaN

```
import matplotlib.pyplot as plt
import seaborn as sns

# Calculate average income by area
average_income_by_area = df.groupby('AREA')['INCOME'].mean().reset_index()

# Create the bar plot
plt.figure(figsize=(10, 6))
sns.barplot(x='AREA', y='INCOME', data=average_income_by_area, palette='viridis')
plt.xlabel('Area')
plt.ylabel('Average Income')
plt.title('Average Income by Area')
plt.grid(axis='y', alpha=0.75)
plt.show()
```

```
/tmp/ipython-input-434604406.py:9: FutureWarning:

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

  sns.barplot(x='AREA', y='INCOME', data=average_income_by_area, palette='viridis')
```



```
import matplotlib.pyplot as plt

# Assuming 'household_totals' is already calculated and available from previous cells.
# If not, it would need to be recalculated:
# household_totals = df.groupby('HHID')['PASTA'].sum()

plt.figure(figsize=(10, 6))
plt.hist(household_totals, bins=30, edgecolor='black')
plt.xlabel('Total Pasta Purchased (Units)')
plt.ylabel('Number of Households')
plt.title('Histogram of Total Pasta Purchases by Household')
plt.grid(axis='y', alpha=0.75)
plt.show()
```


Histogram of Total Pasta Purchases by Household
Number of Households
175
150
125
100
75
50