```
# 📌 Step 1: Install and Import Required Libraries
!pip install seaborn --quiet

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Display settings
pd.set_option('display.max_columns', None)
sns.set(style="whitegrid")


# 📌 Step 2: Load Dataset
file_path = '/content/social_ads.csv'  # Adjust path as needed
df = pd.read_csv(file_path)


# 📌 Step 3: Clean Column Names
df.columns = df.columns.str.strip().str.replace(" ", "_").str.lower()

# Preview data
print("🔍 Preview of data:")
display(df.head())
```

```
🔍 Preview of data:
```

	age	estimatedsalary	purchased
0	19	19000	0
1	35	20000	0
2	26	43000	0
3	27	57000	0
4	19	76000	0

```
# 📌 Step 4: Check Structure and Missing Values
print("📊 Dataset info:")
df.info()

print("\n🧼 Missing values:")
print(df.isnull().sum())

print("\n🧮 Statistics:")
print(df.describe())
```

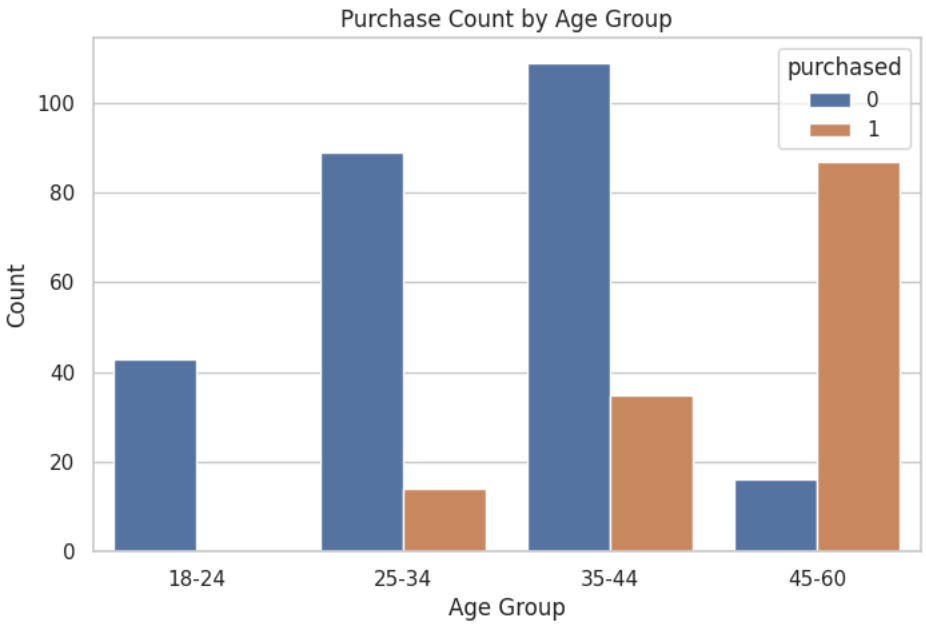
```
📊 Dataset info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 3 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   age              400 non-null    int64
 1   estimatedsalary  400 non-null    int64
 2   purchased        400 non-null    int64
dtypes: int64(3)
memory usage: 9.5 KB

🧼 Missing values:
age                0
estimatedsalary    0
purchased          0
dtype: int64

🧮 Statistics:
                age  estimatedsalary   purchased
count  400.000000       400.000000  400.000000
mean    37.655000     69742.500000    0.357500
std     10.482877     34096.960282    0.479864
min     18.000000     15000.000000    0.000000
25%     29.750000     43000.000000    0.000000
50%     37.000000     70000.000000    0.000000
75%     46.000000     88000.000000    1.000000
```

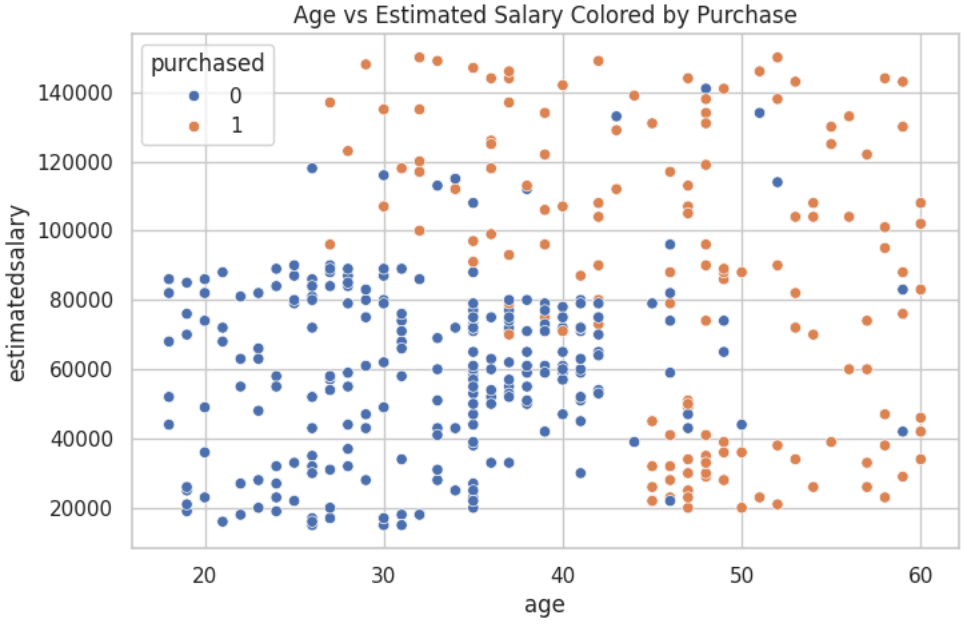
```
     max      60.000000     150000.000000     1.000000
```

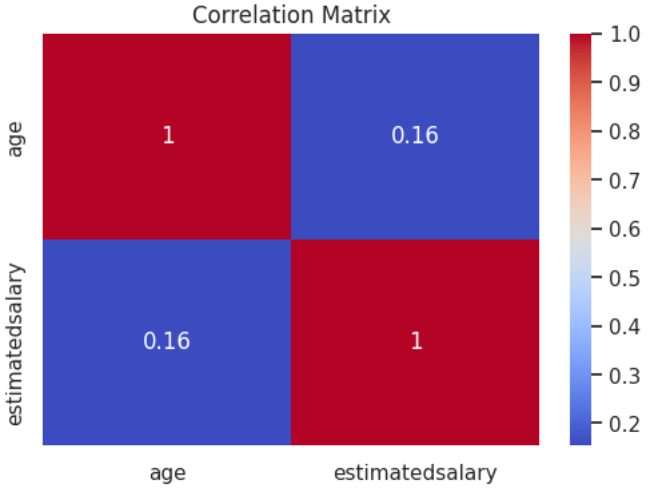
```
# 📌 Step 5: Transformations
# Convert 'purchased' to categorical
df['purchased'] = df['purchased'].astype('category')

# Create age groups
bins = [18, 25, 35, 45, 60]
labels = ['18-24', '25-34', '35-44', '45-60']
df['age_group'] = pd.cut(df['age'], bins=bins, labels=labels, right=False)


# 📌 Step 6: Save Cleaned Data
df.to_csv('/content/cleaned_social_ads.csv', index=False)

# (Optional) Save to SQLite for bonus
import sqlite3
conn = sqlite3.connect('/content/social_ads.db')
df.to_sql('ads_data', conn, if_exists='replace', index=False)
conn.close()

print("✅ Cleaned data saved.")
```

```
✅ Cleaned data saved.
```

```
# 📌 Step 7: EDA (Visuals)

# Purchase distribution by Age Group
plt.figure(figsize=(8, 5))
sns.countplot(data=df, x='age_group', hue='purchased')
plt.title('Purchase Count by Age Group')
plt.xlabel('Age Group')
plt.ylabel('Count')
plt.show()

# Salary vs Age Scatterplot
plt.figure(figsize=(8, 5))
sns.scatterplot(data=df, x='age', y='estimatedsalary', hue='purchased')
plt.title('Age vs Estimated Salary Colored by Purchase')
plt.show()

# Correlation Heatmap
plt.figure(figsize=(6, 4))
sns.heatmap(df[['age', 'estimatedsalary', 'purchased']].corr(numeric_only=True), annot=True, cmap='coolwarm')
plt.title('Correlation Matrix')
plt.show()
```








```
# 📌 Step 8: Insights
```

```
print("\n📌 Key Insights:")
print("""
1. Users aged 25–34 are more likely to purchase the product.
2. Higher estimated salary does not strongly correlate with purchase decision.
3. There's a visible pattern where younger age and mid-level salary show higher purchase interest.
4. These insights can help with targeted ad campaigns by age group.
""")
```

```
📌 Key Insights:

1. Users aged 25–34 are more likely to purchase the product.
2. Higher estimated salary does not strongly correlate with purchase decision.
3. There's a visible pattern where younger age and mid-level salary show higher purchase interest.
4. These insights can help with targeted ad campaigns by age group.
```