```
import pandas as pd
import numpy as np


data = pd.read_csv('adult.csv')
df = pd.DataFrame(data)
df
```

| | age | workclass | fnlwgt | education | education.num | marital.status | occupation | re |
|---|---|---|---|---|---|---|---|---|
| **0** | 90 | ? | 77053 | HS-grad | 9 | Widowed | ? | N |
| **1** | 82 | Private | 132870 | HS-grad | 9 | Widowed | Exec-managerial | N |
| **2** | 66 | ? | 186061 | Some-college | 10 | Widowed | ? | |
| **3** | 54 | Private | 140359 | 7th-8th | 4 | Divorced | Machine-op-inspct | |
| **4** | 41 | Private | 264663 | Some-college | 10 | Separated | Prof-specialty | |
| **...** | ... | ... | ... | ... | ... | ... | ... | |
| **32556** | 22 | Private | 310152 | Some-college | 10 | Never-married | Protective-serv | N |
| **32557** | 27 | Private | 257302 | Assoc-acdm | 12 | Married-civ-spouse | Tech-support | |
| **32558** | 40 | Private | 154374 | HS-grad | 9 | Married-civ-spouse | Machine-op-inspct | |
| **32559** | 58 | Private | 151910 | HS-grad | 9 | Widowed | Adm-clerical | |
| **32560** | 22 | Private | 201490 | HS-grad | 9 | Never-married | Adm-clerical | |

32561 rows × 15 columns

Next steps: Generate code with `df` View recommended plots

```
df1 = df[['age','education.num','fnlwgt','capital.gain','capital.loss','hours.per.week']].copy()
df1
```

| | age | education.num | fnlwgt | capital.gain | capital.loss | hours.per.week |
|---|---|---|---|---|---|---|
| **0** | 90 | 9 | 77053 | 0 | 4356 | 40 |
| **1** | 82 | 9 | 132870 | 0 | 4356 | 18 |
| **2** | 66 | 10 | 186061 | 0 | 4356 | 40 |
| **3** | 54 | 4 | 140359 | 0 | 3900 | 40 |
| **4** | 41 | 10 | 264663 | 0 | 3900 | 40 |
| **...** | ... | ... | ... | ... | ... | ... |
| **32556** | 22 | 10 | 310152 | 0 | 0 | 40 |
| **32557** | 27 | 12 | 257302 | 0 | 0 | 38 |
| **32558** | 40 | 9 | 154374 | 0 | 0 | 40 |
| **32559** | 58 | 9 | 151910 | 0 | 0 | 40 |
| **32560** | 22 | 9 | 201490 | 0 | 0 | 20 |

32561 rows × 6 columns

Next steps: Generate code with `df1` View recommended plots

```
d = []
for column in df1.columns:
  d.append(df1[column].tolist())
```

```
def cov_coffe(x_mean,y_mean,x,y):
  l = list()
  for i in x:
    l.append(i-x_mean)
  k = list()
  for j in y:
    k.append(j-y_mean)
  sum =0
  for t in range(len(l)):
    sum+=(l[t]*k[t])
  cov = sum/(len(l)-1)
  return cov;


n = len(df1.columns)


cov_matrix = [[0 for i in range(n)]
                 for j in range(n)];
for i in range(n):
  x = d[i]
  x_mean = np.mean(x)
  for j in range(n):
    y = d[j]
    y_mean = np.mean(y)
    cov_matrix[i][j] = cov_coffe(x_mean,y_mean,x,y)
cov_matrix
```

```
[[186.0614002487955,
  1.2818493235187831,
  -110350.68530013379,
  7824.818536517176,
  317.5607422803612,
  11.580129717973232],
 [1.2818493235187831,
  6.618889907032666,
  -11729.527298134162,
  2330.007877380341,
  82.85644469547239,
  4.705337944611554],
 [-110350.68530013379,
  -11729.527298134162,
  11140797791.841866,
  336662.49599813507,
  -436030.33316658007,
  -24460.426185446686],
 [7824.818536517176,
  2330.007877380341,
  336662.49599813507,
  54542539.17834288,
  -94085.76068824342,
  7150.032029176822],
 [317.5607422803612,
  82.85644469547239,
  -436030.33316658007,
  -94085.76068824342,
  162376.93781406686,
  269.9537545839114],
 [11.580129717973232,
  4.705337944611554,
  -24460.426185446686,
  7150.032029176822,
  269.9537545839114,
  152.4589950504292]]
```

+ Code + Text

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.