In [2]:
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import warnings

warnings.filterwarnings('ignore')


# Display all the columns of the Dataframe
pd.pandas.set_option('display.max_columns',None)
```

In [2]:
```python
data=pd.read_csv(r'D:\New folder\Datasets\Census Income dataset\adult.data',names=["Age"
```

In [3]:
```python
data.head()
```

Out[3]:

	Age	Workclass	fnlwgt	Education	Education_num	Marital_Status	Occupation	Relationship	Race	Sex
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female

In [4]:
```python
data.shape
```

Out[4]:
```
(32561, 15)
```

In [5]:
```python
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32561 entries, 0 to 32560
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   Age             32561 non-null  int64 
 1   Workclass       32561 non-null  object
 2   fnlwgt          32561 non-null  int64 
 3   Education       32561 non-null  object
 4   Education_num   32561 non-null  int64 
 5   Marital_Status  32561 non-null  object
 6   Occupation      32561 non-null  object
 7   Relationship    32561 non-null  object
 8   Race            32561 non-null  object
 9   Sex             32561 non-null  object
 10  Capital_gain    32561 non-null  int64 
 11  Capital_loss    32561 non-null  int64 
 12  Hours_per_week  32561 non-null  int64 
 13  Native_Country  32561 non-null  object
 14  Class           32561 non-null  object
dtypes: int64(6), object(9)
memory usage: 3.7+ MB
```

In [6]:
```
data_test=pd.read_csv(r'D:\New folder\Datasets\Census Income dataset\adult.test',names=[
```

In [7]:
```
data_test.head()
```

Out[7]:

	Age	Workclass	fnlwgt	Education	Education_num	Marital_Status	Occupation	Relationship	Race	Se
0	|1x3 Cross validator	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
1	25	Private	226802.0	11th	7.0	Never-married	Machine-op-inspct	Own-child	Black	Ma
2	38	Private	89814.0	HS-grad	9.0	Married-civ-spouse	Farming-fishing	Husband	White	Ma
3	28	Local-gov	336951.0	Assoc-acdm	12.0	Married-civ-spouse	Protective-serv	Husband	White	Ma
4	44	Private	160323.0	Some-college	10.0	Married-civ-spouse	Machine-op-inspct	Husband	Black	Ma

In [8]:
```
# Deleting first row
data_test.drop(index=0,inplace=True)
```

In [9]:
```
data_test.head()
```

Out[9]:

	Age	Workclass	fnlwgt	Education	Education_num	Marital_Status	Occupation	Relationship	Race	Sex
1	25	Private	226802.0	11th	7.0	Never-married	Machine-op-inspct	Own-child	Black	Male
2	38	Private	89814.0	HS-grad	9.0	Married-civ-spouse	Farming-fishing	Husband	White	Male
3	28	Local-gov	336951.0	Assoc-acdm	12.0	Married-civ-spouse	Protective-serv	Husband	White	Male
4	44	Private	160323.0	Some-college	10.0	Married-civ-spouse	Machine-op-inspct	Husband	Black	Male
5	18	?	103497.0	Some-college	10.0	Never-married	?	Own-child	White	Female

In [10]:
```
data_test.shape
```

Out[10]:
```
(16281, 15)
```

In [11]:
```
data_test.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16281 entries, 1 to 16281
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Age             16281 non-null  object
 1   Workclass       16281 non-null  object
 2   fnlwgt          16281 non-null  float64
 3   Education       16281 non-null  object
 4   Education_num   16281 non-null  float64
 5   Marital_Status  16281 non-null  object
 6   Occupation      16281 non-null  object
 7   Relationship    16281 non-null  object
 8   Race            16281 non-null  object
```

```
 9   Sex             16281 non-null  object
 10  Capital_gain    16281 non-null  float64
 11  Capital_loss    16281 non-null  float64
 12  Hours_per_week  16281 non-null  float64
 13  Native_Country  16281 non-null  object
 14  Class           16281 non-null  object
dtypes: float64(5), object(10)
memory usage: 1.9+ MB
```

In [12]:
```
# Here combining the both dataframe data and data_test using concat function
data=pd.concat([data,data_test], axis=0)
data.head()
```

Out[12]:

	Age	Workclass	fnlwgt	Education	Education_num	Marital_Status	Occupation	Relationship	Race	Sex
0	39	State-gov	77516.0	Bachelors	13.0	Never-married	Adm-clerical	Not-in-family	White	Male
1	50	Self-emp-not-inc	83311.0	Bachelors	13.0	Married-civ-spouse	Exec-managerial	Husband	White	Male
2	38	Private	215646.0	HS-grad	9.0	Divorced	Handlers-cleaners	Not-in-family	White	Male
3	53	Private	234721.0	11th	7.0	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male
4	28	Private	338409.0	Bachelors	13.0	Married-civ-spouse	Prof-specialty	Wife	Black	Female

In [13]:
```
data.shape
```

Out[13]:
```
(48842, 15)
```

In [14]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 48842 entries, 0 to 16281
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Age             48842 non-null  object
 1   Workclass       48842 non-null  object
 2   fnlwgt          48842 non-null  float64
 3   Education       48842 non-null  object
 4   Education_num   48842 non-null  float64
 5   Marital_Status  48842 non-null  object
 6   Occupation      48842 non-null  object
 7   Relationship    48842 non-null  object
 8   Race            48842 non-null  object
 9   Sex             48842 non-null  object
 10  Capital_gain    48842 non-null  float64
 11  Capital_loss    48842 non-null  float64
 12  Hours_per_week  48842 non-null  float64
 13  Native_Country  48842 non-null  object
 14  Class           48842 non-null  object
dtypes: float64(5), object(10)
memory usage: 6.0+ MB
```

In [15]:
```
# Age column is in object dtype convertimg it into integer dtype
data['Age']=data['Age'].astype(np.int64)
```

In [16]:
```
data.dtypes
```

Out[16]:
```
Age                   int64
```

```
Workclass          object
fnlwgt            float64
Education          object
Education_num     float64
Marital_Status     object
Occupation         object
Relationship       object
Race               object
Sex                object
Capital_gain      float64
Capital_loss      float64
Hours_per_week    float64
Native_Country     object
Class              object
dtype: object
```

In [17]:

```
data.describe()
```

Out[17]:

	Age	fnlwgt	Education_num	Capital_gain	Capital_loss	Hours_per_week
count	48842.000000	4.884200e+04	48842.000000	48842.000000	48842.000000	48842.000000
mean	38.643585	1.896641e+05	10.078089	1079.067626	87.502314	40.422382
std	13.710510	1.056040e+05	2.570973	7452.019058	403.004552	12.391444
min	17.000000	1.228500e+04	1.000000	0.000000	0.000000	1.000000
25%	28.000000	1.175505e+05	9.000000	0.000000	0.000000	40.000000
50%	37.000000	1.781445e+05	10.000000	0.000000	0.000000	40.000000
75%	48.000000	2.376420e+05	12.000000	0.000000	0.000000	45.000000
max	90.000000	1.490400e+06	16.000000	99999.000000	4356.000000	99.000000

In [18]:

```
# Checking if there are duplicates
data.duplicated().sum()
```

Out[18]:

```
29
```

In [19]:

```
data[data.duplicated()]
```

Out[19]:

	Age	Workclass	fnlwgt	Education	Education_num	Marital_Status	Occupation	Relationship	Race
4881	25	Private	308144.0	Bachelors	13.0	Never-married	Craft-repair	Not-in-family	White
5104	90	Private	52386.0	Some-college	10.0	Never-married	Other-service	Not-in-family	Asian-Pac-Islander
9171	21	Private	250051.0	Some-college	10.0	Never-married	Prof-specialty	Own-child	White
11631	20	Private	107658.0	Some-college	10.0	Never-married	Tech-support	Not-in-family	White
13084	25	Private	195994.0	1st-4th	2.0	Never-married	Priv-house-serv	Not-in-family	White
15059	21	Private	243368.0	Preschool	1.0	Never-married	Farming-fishing	Not-in-family	White
17040	46	Private	173243.0	HS-grad	9.0	Married-civ-spouse	Craft-repair	Husband	White

18555	30	Private	144593.0	HS-grad	9.0	Never-married	Other-service	Not-in-family	Black	
18698	19	Private	97261.0	HS-grad	9.0	Never-married	Farming-fishing	Not-in-family	White	
21318	19	Private	138153.0	Some-college	10.0	Never-married	Adm-clerical	Own-child	White	F
21490	19	Private	146679.0	Some-college	10.0	Never-married	Exec-managerial	Own-child	Black	
21875	49	Private	31267.0	7th-8th	4.0	Married-civ-spouse	Craft-repair	Husband	White	
22300	25	Private	195994.0	1st-4th	2.0	Never-married	Priv-house-serv	Not-in-family	White	F
22367	44	Private	367749.0	Bachelors	13.0	Never-married	Prof-specialty	Not-in-family	White	F
22494	49	Self-emp-not-inc	43479.0	Some-college	10.0	Married-civ-spouse	Craft-repair	Husband	White	
25872	23	Private	240137.0	5th-6th	3.0	Never-married	Handlers-cleaners	Not-in-family	White	
26313	28	Private	274679.0	Masters	14.0	Never-married	Prof-specialty	Not-in-family	White	
28230	27	Private	255582.0	HS-grad	9.0	Never-married	Machine-op-inspct	Not-in-family	White	F
28522	42	Private	204235.0	Some-college	10.0	Married-civ-spouse	Prof-specialty	Husband	White	
28846	39	Private	30916.0	HS-grad	9.0	Married-civ-spouse	Craft-repair	Husband	White	
29157	38	Private	207202.0	HS-grad	9.0	Married-civ-spouse	Machine-op-inspct	Husband	White	
30845	46	Private	133616.0	Some-college	10.0	Divorced	Adm-clerical	Unmarried	White	F
31993	19	Private	251579.0	Some-college	10.0	Never-married	Other-service	Own-child	White	
32404	35	Private	379959.0	HS-grad	9.0	Divorced	Other-service	Not-in-family	White	F
865	24	Private	194630.0	Bachelors	13.0	Never-married	Prof-specialty	Not-in-family	White	
11190	37	Private	52870.0	Bachelors	13.0	Married-civ-spouse	Exec-managerial	Husband	White	
11213	29	Private	36440.0	Bachelors	13.0	Never-married	Adm-clerical	Not-in-family	White	F
13849	30	Private	180317.0	Assoc-voc	11.0	Divorced	Machine-op-inspct	Not-in-family	White	
15961	18	Self-emp-inc	378036.0	12th	8.0	Never-married	Farming-fishing	Own-child	White	

In [20]:
```
data=data.drop_duplicates()
```

In [21]:
```
data.shape
```

Out[21]: (48813, 15)

In [22]:
```
data.head()
```

Out[22]:

	Age	Workclass	fnlwgt	Education	Education_num	Marital_Status	Occupation	Relationship	Race	Sex
0	39	State-gov	77516.0	Bachelors	13.0	Never-married	Adm-clerical	Not-in-family	White	Male
1	50	Self-emp-not-inc	83311.0	Bachelors	13.0	Married-civ-spouse	Exec-managerial	Husband	White	Male
2	38	Private	215646.0	HS-grad	9.0	Divorced	Handlers-cleaners	Not-in-family	White	Male
3	53	Private	234721.0	11th	7.0	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male
4	28	Private	338409.0	Bachelors	13.0	Married-civ-spouse	Prof-specialty	Wife	Black	Female

In [23]:
```
data['Workclass'].unique()
```

Out[23]:
```
array([' State-gov', ' Self-emp-not-inc', ' Private', ' Federal-gov',
       ' Local-gov', ' ?', ' Self-emp-inc', ' Without-pay',
       ' Never-worked'], dtype=object)
```

In [24]:
```
data['Occupation'].unique()
```

Out[24]:
```
array([' Adm-clerical', ' Exec-managerial', ' Handlers-cleaners',
       ' Prof-specialty', ' Other-service', ' Sales', ' Craft-repair',
       ' Transport-moving', ' Farming-fishing', ' Machine-op-inspct',
       ' Tech-support', ' ?', ' Protective-serv', ' Armed-Forces',
       ' Priv-house-serv'], dtype=object)
```

In [25]:
```
#Replacing ' ?' with NA and droping all the NA values
data_cleaned = data.replace(' ?',pd.NA).dropna()
data_cleaned.head()
```

Out[25]:

	Age	Workclass	fnlwgt	Education	Education_num	Marital_Status	Occupation	Relationship	Race	Sex
0	39	State-gov	77516.0	Bachelors	13.0	Never-married	Adm-clerical	Not-in-family	White	Male
1	50	Self-emp-not-inc	83311.0	Bachelors	13.0	Married-civ-spouse	Exec-managerial	Husband	White	Male
2	38	Private	215646.0	HS-grad	9.0	Divorced	Handlers-cleaners	Not-in-family	White	Male
3	53	Private	234721.0	11th	7.0	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male
4	28	Private	338409.0	Bachelors	13.0	Married-civ-spouse	Prof-specialty	Wife	Black	Female

In [26]:
```
data_cleaned.shape
```

Out[26]: (45194, 15)

In [27]:
```
data_cleaned[data_cleaned['Occupation']==' ?']
```

Out[27]:

Age	Workclass	fnlwgt	Education	Education_num	Marital_Status	Occupation	Relationship	Race	Sex	Capita

In [28]:
```python
# Checking null values
for col in data_cleaned:
    pct_missing=data_cleaned[col].isnull().mean()
    print(f'{col} - {pct_missing :0.1%}')
```

```
Age - 0.0%
Workclass - 0.0%
fnlwgt - 0.0%
Education - 0.0%
Education_num - 0.0%
Marital_Status - 0.0%
Occupation - 0.0%
Relationship - 0.0%
Race - 0.0%
Sex - 0.0%
Capital_gain - 0.0%
Capital_loss - 0.0%
Hours_per_week - 0.0%
Native_Country - 0.0%
Class - 0.0%
```

There are no null values

In [29]:
```python
# Storing the cleaned dataset in csv format
data_cleaned.to_csv('Cleaned_Census_Income_dataset.csv',index=False)
```

In [4]:
```python
df=pd.read_csv('Cleaned_Census_Income_dataset.csv')
```

In [31]:
```python
df.head()
```

Out[31]:

	Age	Workclass	fnlwgt	Education	Education_num	Marital_Status	Occupation	Relationship	Race	Sex
0	39	State-gov	77516.0	Bachelors	13.0	Never-married	Adm-clerical	Not-in-family	White	Male
1	50	Self-emp-not-inc	83311.0	Bachelors	13.0	Married-civ-spouse	Exec-managerial	Husband	White	Male
2	38	Private	215646.0	HS-grad	9.0	Divorced	Handlers-cleaners	Not-in-family	White	Male
3	53	Private	234721.0	11th	7.0	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male
4	28	Private	338409.0	Bachelors	13.0	Married-civ-spouse	Prof-specialty	Wife	Black	Female

In [32]:
```python
df.shape
```

Out[32]:
```
(45194, 15)
```

In [33]:
```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45194 entries, 0 to 45193
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Age             45194 non-null  int64
 1   Workclass       45194 non-null  object
 2   fnlwgt          45194 non-null  float64
 3   Education       45194 non-null  object
```

```
 4   Education_num   45194 non-null  float64
 5   Marital_Status  45194 non-null  object
 6   Occupation      45194 non-null  object
 7   Relationship    45194 non-null  object
 8   Race            45194 non-null  object
 9   Sex             45194 non-null  object
 10  Capital_gain    45194 non-null  float64
 11  Capital_loss    45194 non-null  float64
 12  Hours_per_week  45194 non-null  float64
 13  Native_Country  45194 non-null  object
 14  Class           45194 non-null  object
dtypes: float64(5), int64(1), object(9)
memory usage: 5.2+ MB
```

In [17]:
```
#list of categorical variables
categorical_features = [feature for feature in df.columns if df[feature].dtype == 'O']
```

In [18]:
```
categorical_features
```

Out[18]:
```
['Workclass',
 'Education',
 'Marital_Status',
 'Occupation',
 'Relationship',
 'Race',
 'Sex',
 'Native_Country',
 'Class']
```

In [36]:
```
df[categorical_features]
```

Out[36]:

	Workclass	Education	Marital_Status	Occupation	Relationship	Race	Sex	Native_Country	Class
0	State-gov	Bachelors	Never-married	Adm-clerical	Not-in-family	White	Male	United-States	<=50K
1	Self-emp-not-inc	Bachelors	Married-civ-spouse	Exec-managerial	Husband	White	Male	United-States	<=50K
2	Private	HS-grad	Divorced	Handlers-cleaners	Not-in-family	White	Male	United-States	<=50K
3	Private	11th	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	United-States	<=50K
4	Private	Bachelors	Married-civ-spouse	Prof-specialty	Wife	Black	Female	Cuba	<=50K
...	...	...	...	...	...	...	...	...	...
45189	Private	Bachelors	Never-married	Prof-specialty	Own-child	White	Male	United-States	<=50K.
45190	Private	Bachelors	Divorced	Prof-specialty	Not-in-family	White	Female	United-States	<=50K.
45191	Private	Bachelors	Married-civ-spouse	Prof-specialty	Husband	White	Male	United-States	<=50K.
45192	Private	Bachelors	Divorced	Adm-clerical	Own-child	Asian-Pac-Islander	Male	United-States	<=50K.
45193	Self-emp-inc	Bachelors	Married-civ-spouse	Exec-managerial	Husband	White	Male	United-States	>50K.

45194 rows × 9 columns

In [19]:
```
# list of numeric variables
numeric_features = [feature for feature in df.columns if df[feature].dtype!='O']
```

In [20]:
```
numeric_features
```

Out[20]:
```
['Age',
 'fnlwgt',
 'Education_num',
 'Capital_gain',
 'Capital_loss',
 'Hours_per_week']
```

In [39]:
```
df[numeric_features]
```

Out[39]:

	Age	fnlwgt	Education_num	Capital_gain	Capital_loss	Hours_per_week
0	39	77516.0	13.0	2174.0	0.0	40.0
1	50	83311.0	13.0	0.0	0.0	13.0
2	38	215646.0	9.0	0.0	0.0	40.0
3	53	234721.0	7.0	0.0	0.0	40.0
4	28	338409.0	13.0	0.0	0.0	40.0
...	...	...	...	...	...	...
45189	33	245211.0	13.0	0.0	0.0	40.0
45190	39	215419.0	13.0	0.0	0.0	36.0
45191	38	374983.0	13.0	0.0	0.0	50.0
45192	44	83891.0	13.0	5455.0	0.0	40.0
45193	35	182148.0	13.0	0.0	0.0	60.0

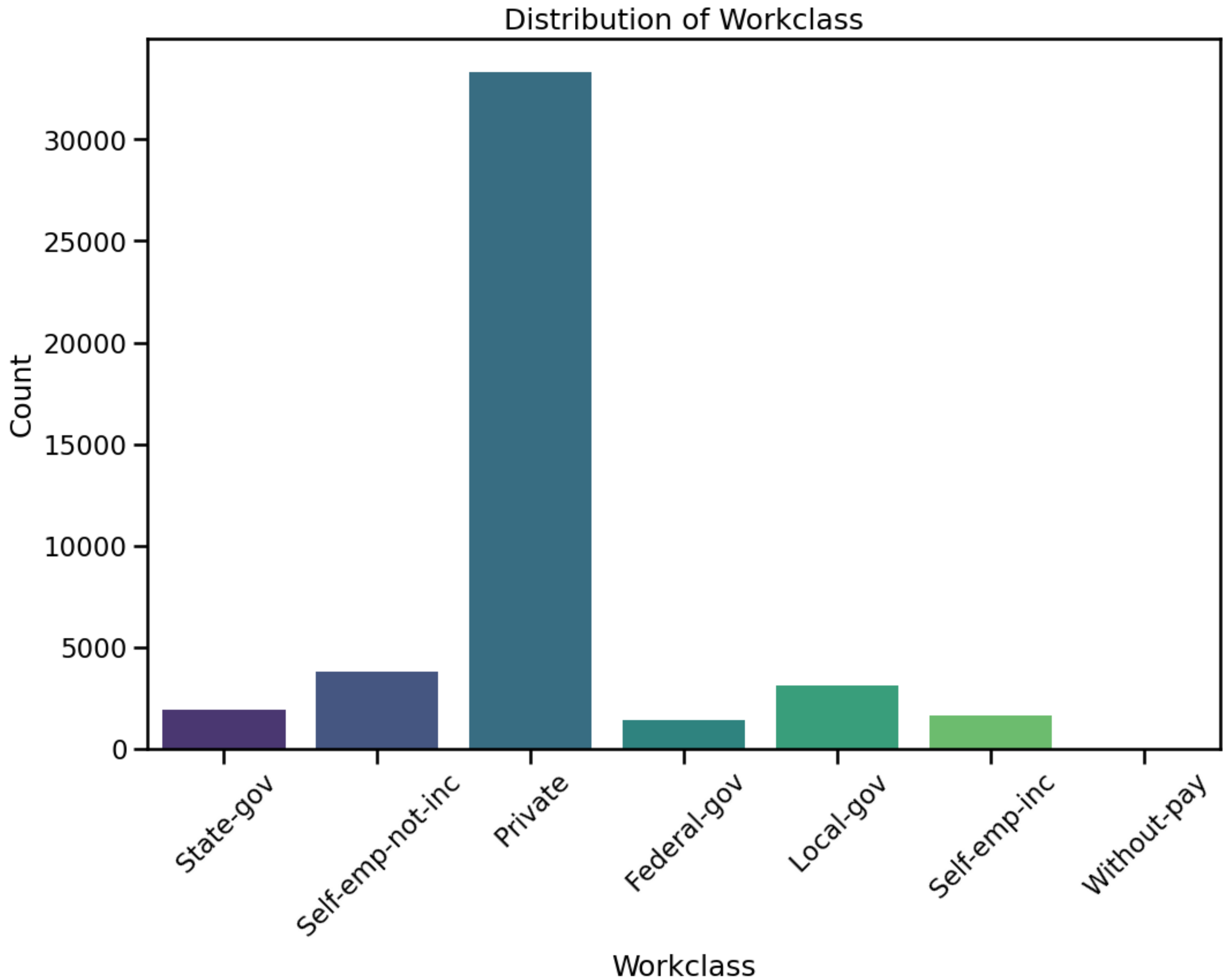
45194 rows × 6 columns

In [57]:
```
df['Class'].value_counts(normalize=True)*100
```

Out[57]:
```
Class
 <=50K    75.204673
 >50K     24.795327
Name: proportion, dtype: float64
```

Univeriate Analysis

In [58]:
```
# Distribution of Workclass
plt.figure(figsize=(12, 8))
sns.countplot(x='Workclass', data=df, palette='viridis')
plt.title('Distribution of Workclass')
plt.xlabel('Workclass')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```

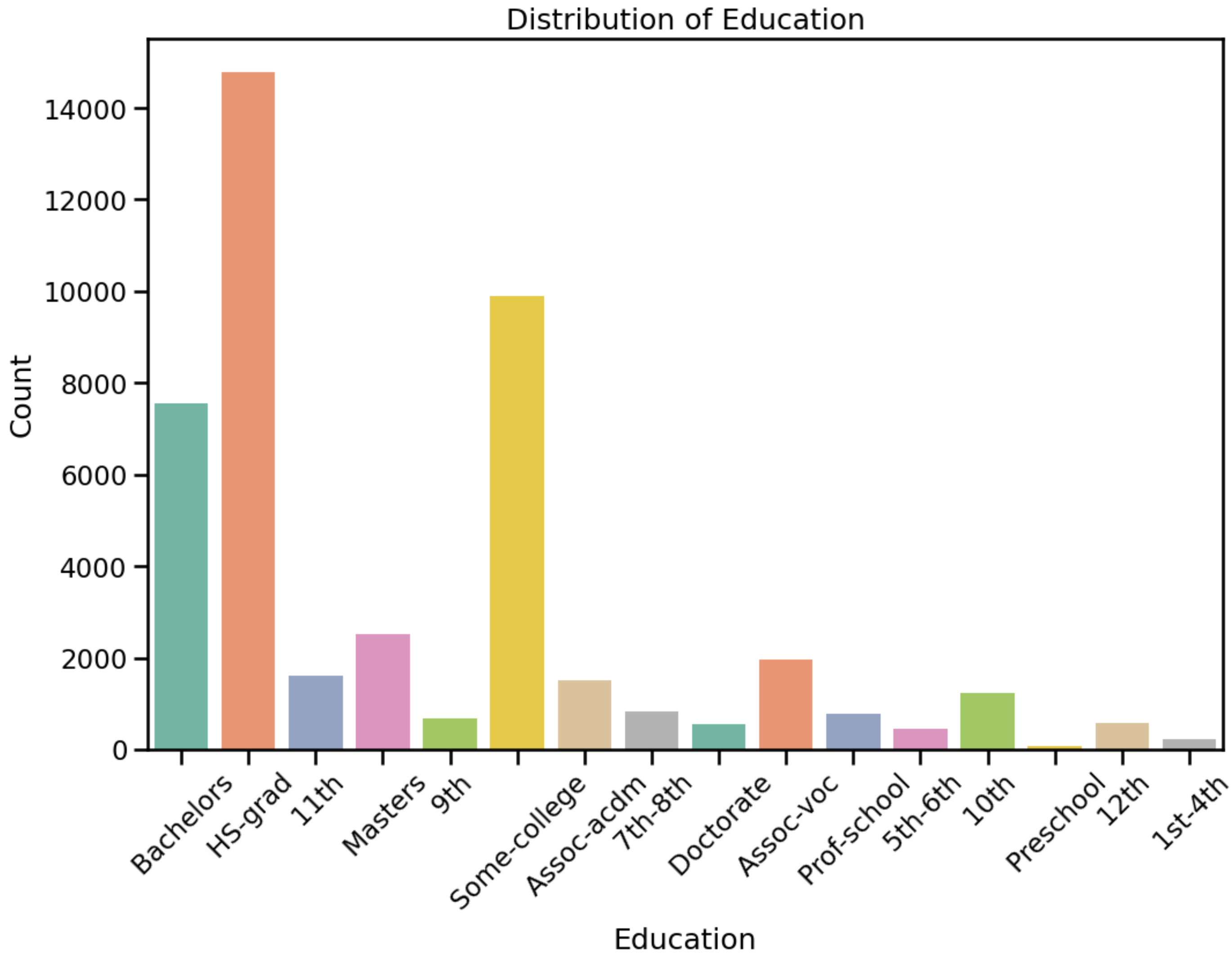


Observations

The majority of individuals are employed in the private sector, with private employment significantly outnumbering other sectors. Based on the chart, it is evident that approximately 80% of the population is engaged in private-sector work.

In [59]:

```
# Distribution of Education

plt.figure(figsize=(12, 8))
sns.countplot(x='Education', data=df, palette='Set2')
plt.title('Distribution of Education')
plt.xlabel('Education')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```

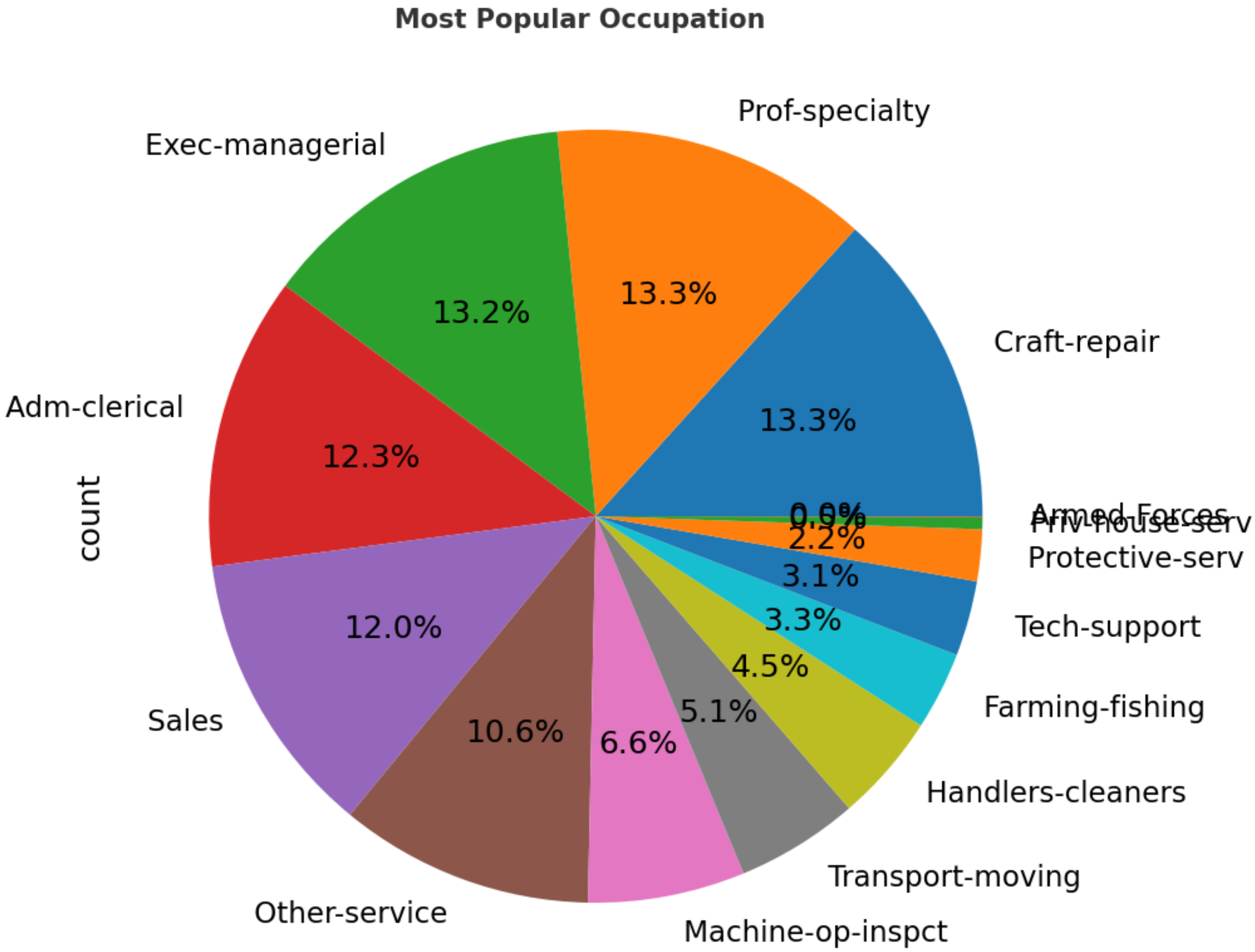


Observations

The educational graph reveals that the highest number of individuals possess an HS-grad degree, followed by Some-college degree, and Bachelors degree, respectively.

In [62]:
```
# Most popular Occupation
plt.suptitle('Most Popular Occupation', fontsize=15, fontweight='bold', alpha=0.8, y=0.9
df['Occupation'].value_counts().plot.pie(y=df['Occupation'],figsize=(10,10), autopct='%1
```

Out[62]:
```
<Axes: ylabel='count'>
```

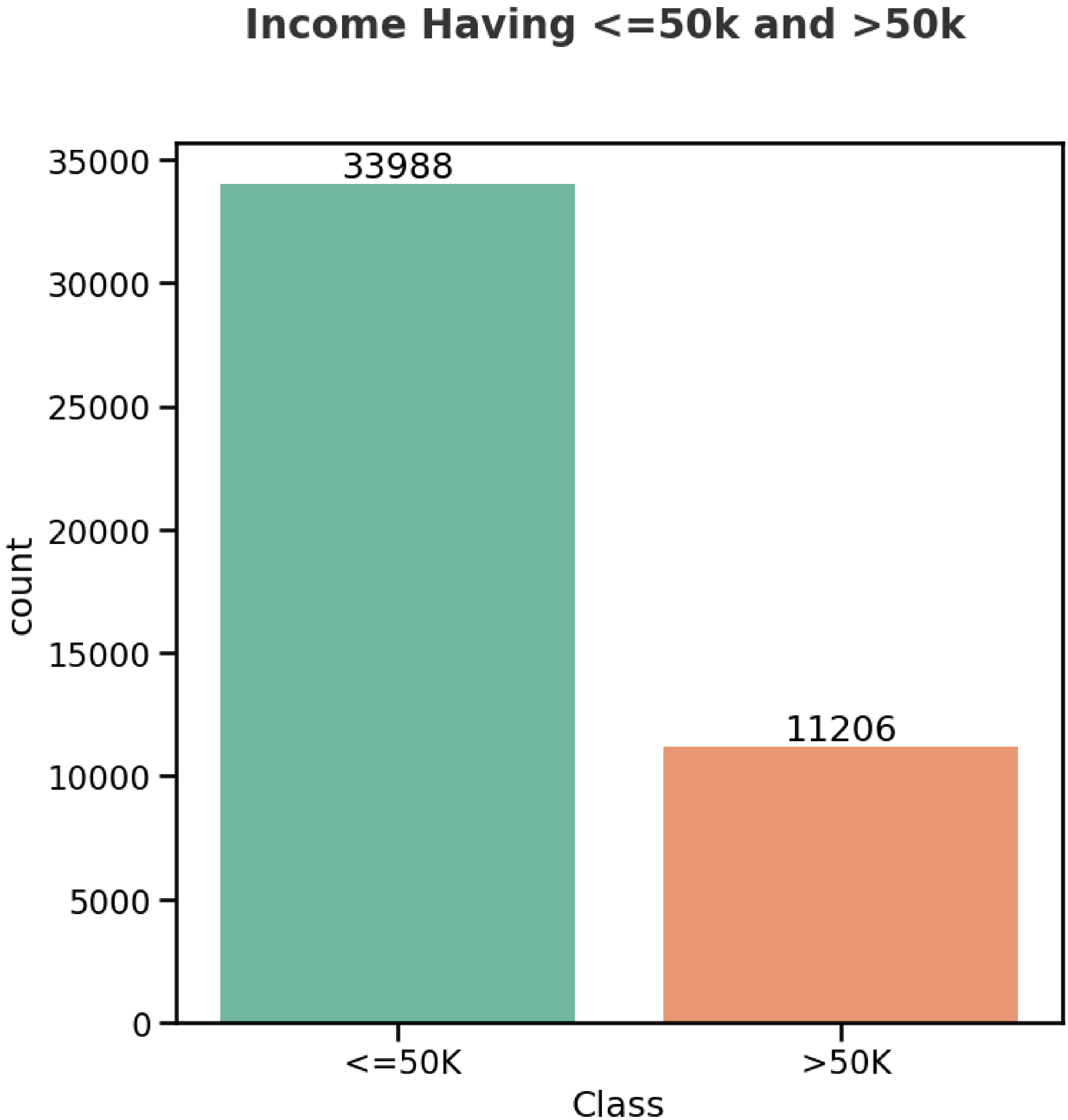


In [63]:

```
#Checking people income ratio

plt.figure(figsize=(8,8))
plt.suptitle('Income Having <=50k and >50k',fontsize=20, fontweight='bold',alpha=0.8,y=1
plt.tight_layout()

graph=sns.countplot(x=df['Class'],palette='Set2')
values = df['Class'].value_counts(ascending=False).values

graph.bar_label(container=graph.containers[0], labels=values)
```

Out[63]:

```
[Text(0, 0, '33988'), Text(0, 0, '11206')]
```

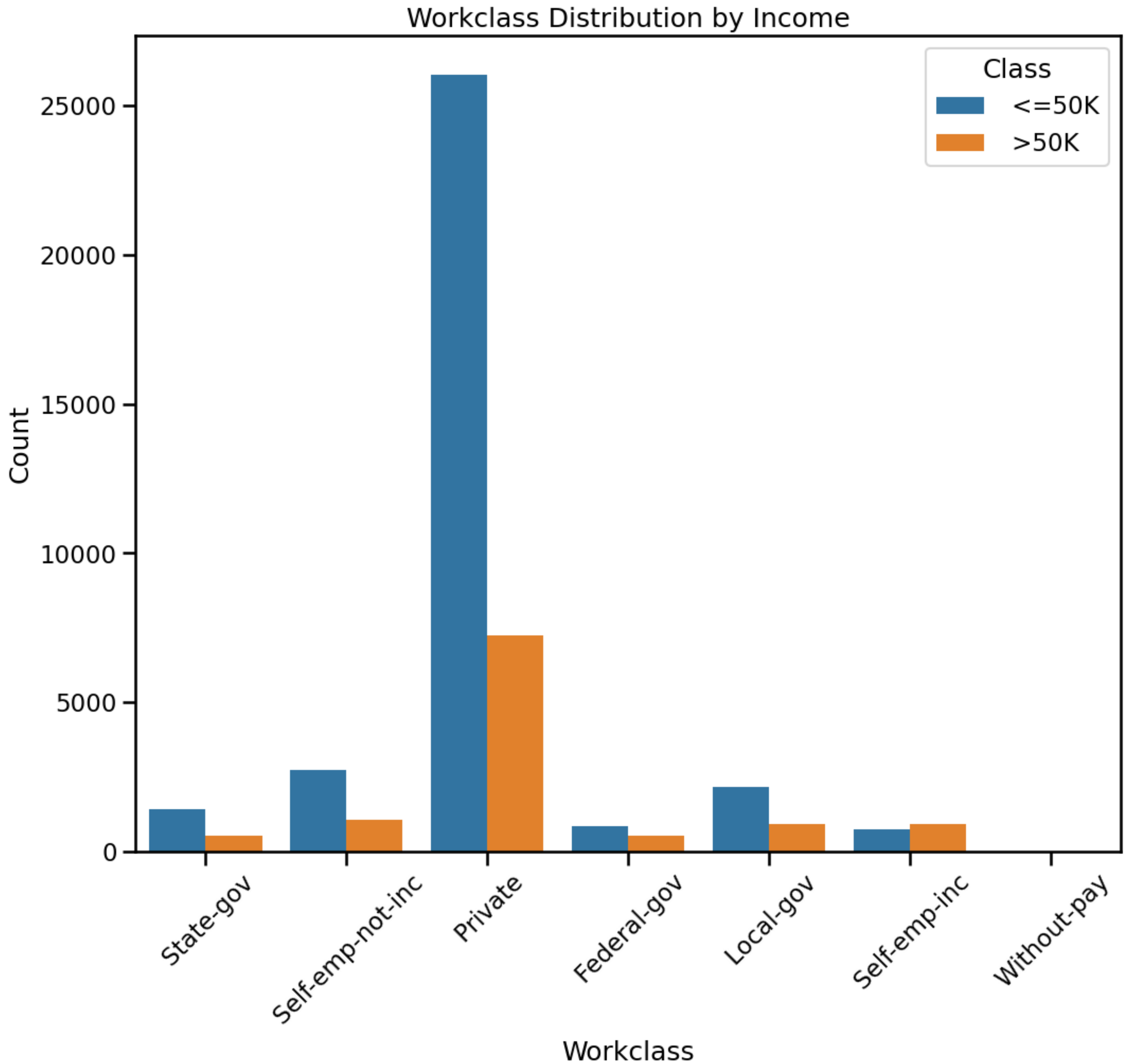


Observations

Here we can see that there is a huge difference between people having income <=50k and >50k. People having income <=50k is much greater than those are having >50k income.

In [65]:

```
plt.figure(figsize=(12, 10))
sns.countplot(x='Workclass', hue='Class', data=df)
plt.title('Workclass Distribution by Income')
plt.xlabel('Workclass')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```



Observations:

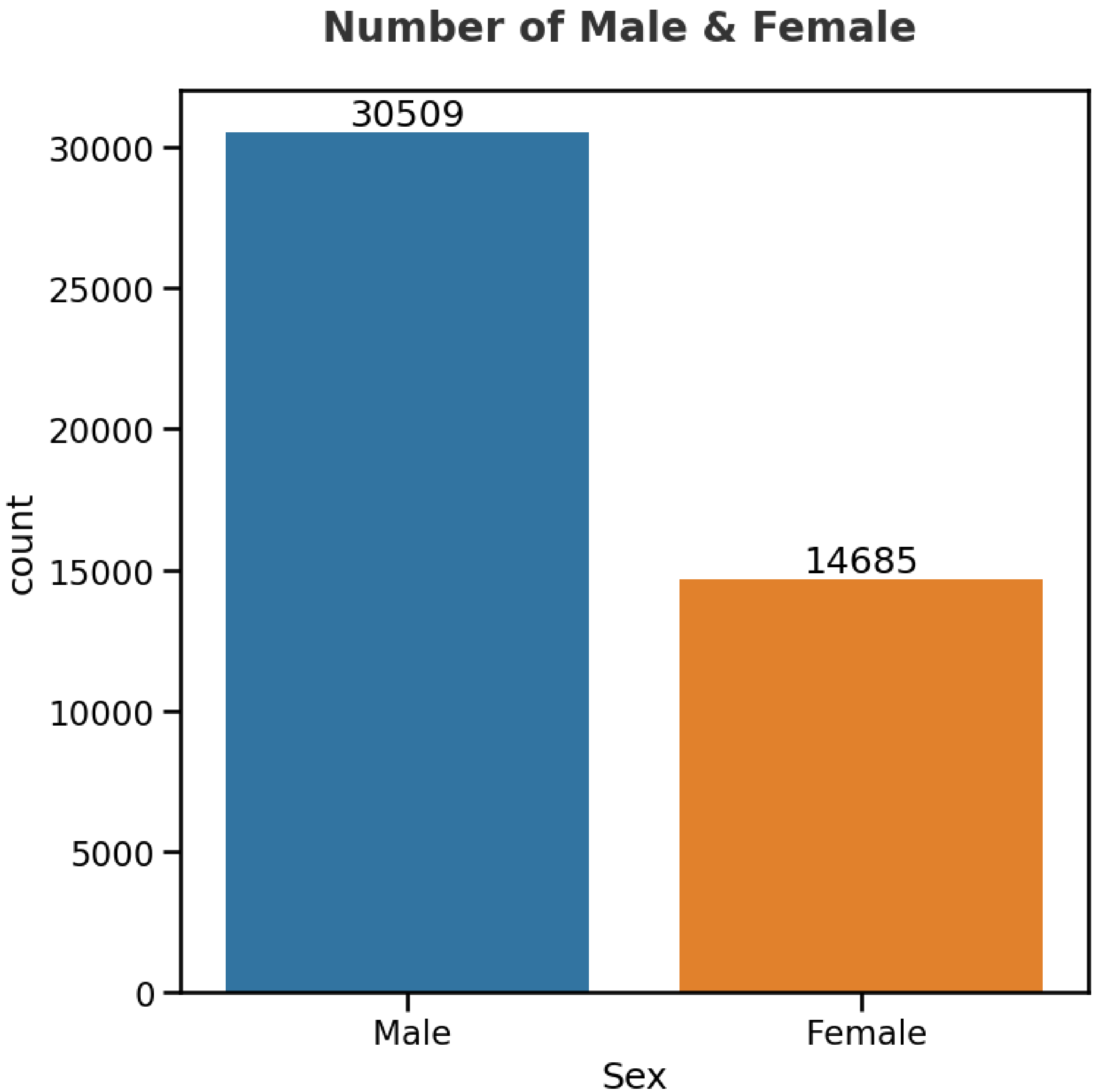
1. A significant income gap exists among individuals working in the private sector, with a notably higher count having incomes exceeding 50k compared to those earning 50k or less.

2. Within the Self-emp-inc category, there are more individuals earning ove50k than those earning 50k or less.

In [68]:

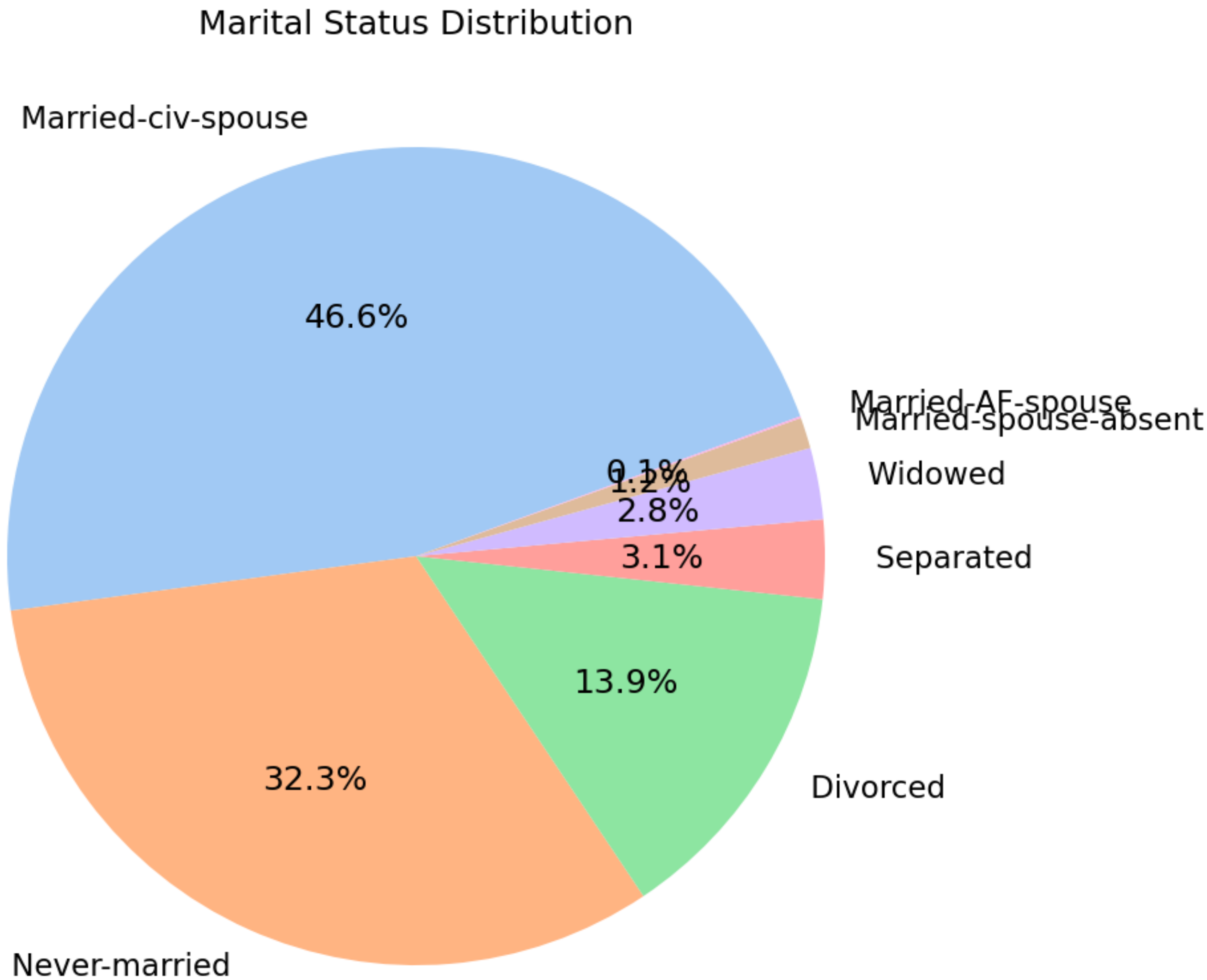
```
# number of people with respect to gender

plt.figure(figsize=(8,8))
plt.suptitle('Number of Male & Female',fontsize=20, fontweight='bold',alpha=0.8,y=0.95)
plt.tight_layout()

graph=sns.countplot(x=df['Sex'])
values = df['Sex'].value_counts(ascending=False).values

graph.bar_label(container=graph.containers[0], labels=values)
```

Out[68]: [Text(0, 0, '30509'), Text(0, 0, '14685')]



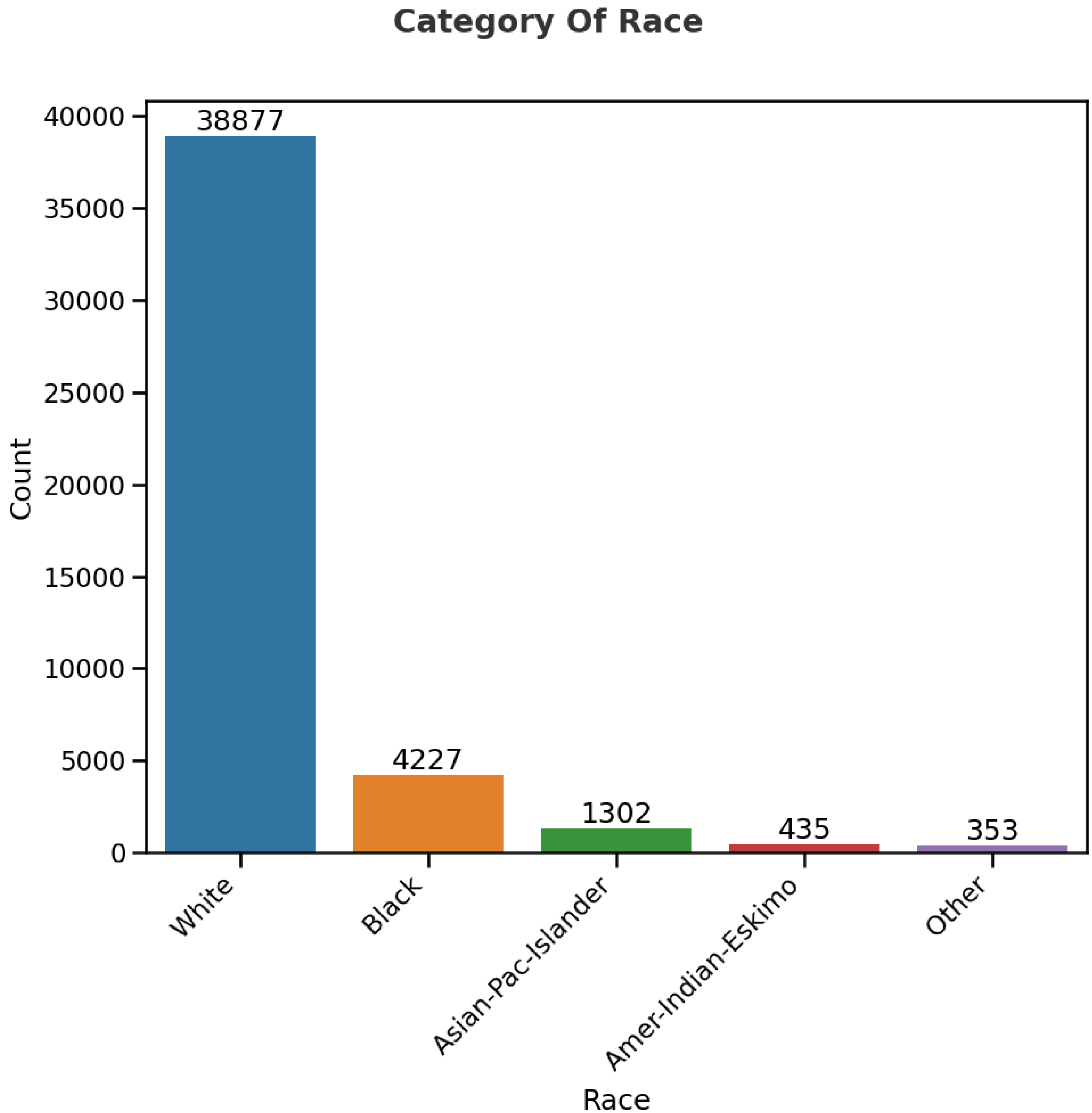
The count of males significantly exceeds the count of females.

In [80]:
```
marital_status_counts = df['Marital_Status'].value_counts()
plt.figure(figsize=(10,10))
plt.pie(marital_status_counts, labels=marital_status_counts.index, autopct='%1.1f%%',sta
plt.title('Marital Status Distribution')
plt.show()
```

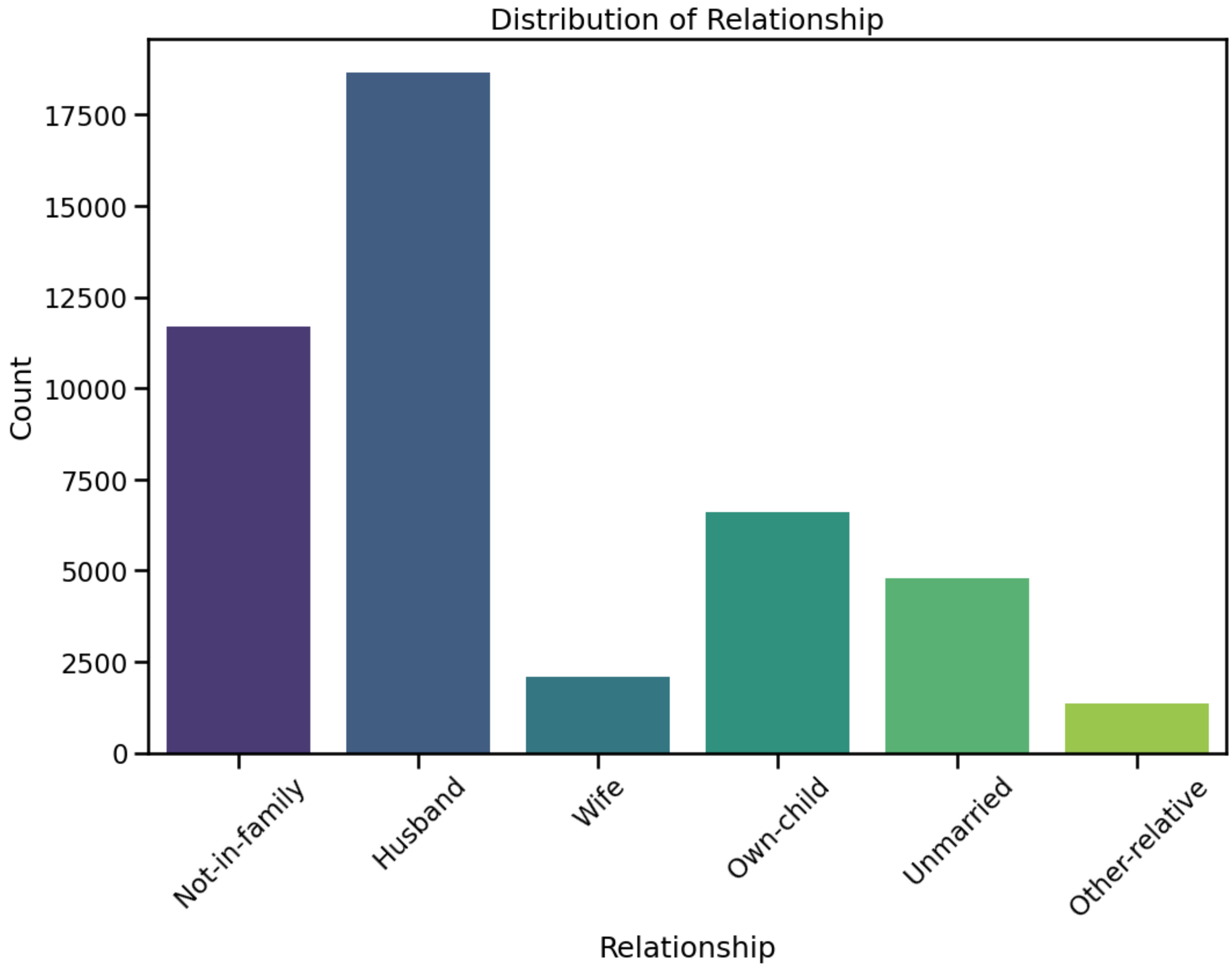


In [83]:

```
# people's belonging to community with respect to race

plt.figure(figsize=(10,10))
plt.suptitle('Category Of Race',fontsize=20, fontweight='bold',alpha=0.8,y=1.0)
plt.tight_layout()

graph=sns.countplot(x=df['Race'])
values = df['Race'].value_counts(ascending=False).values

graph.bar_label(container=graph.containers[0], labels=values)

plt.xlabel('Race')
plt.ylabel('Count')
plt.xticks(rotation=45, ha='right')  # Adjust rotation for better visibility
plt.tight_layout()

plt.show()
```

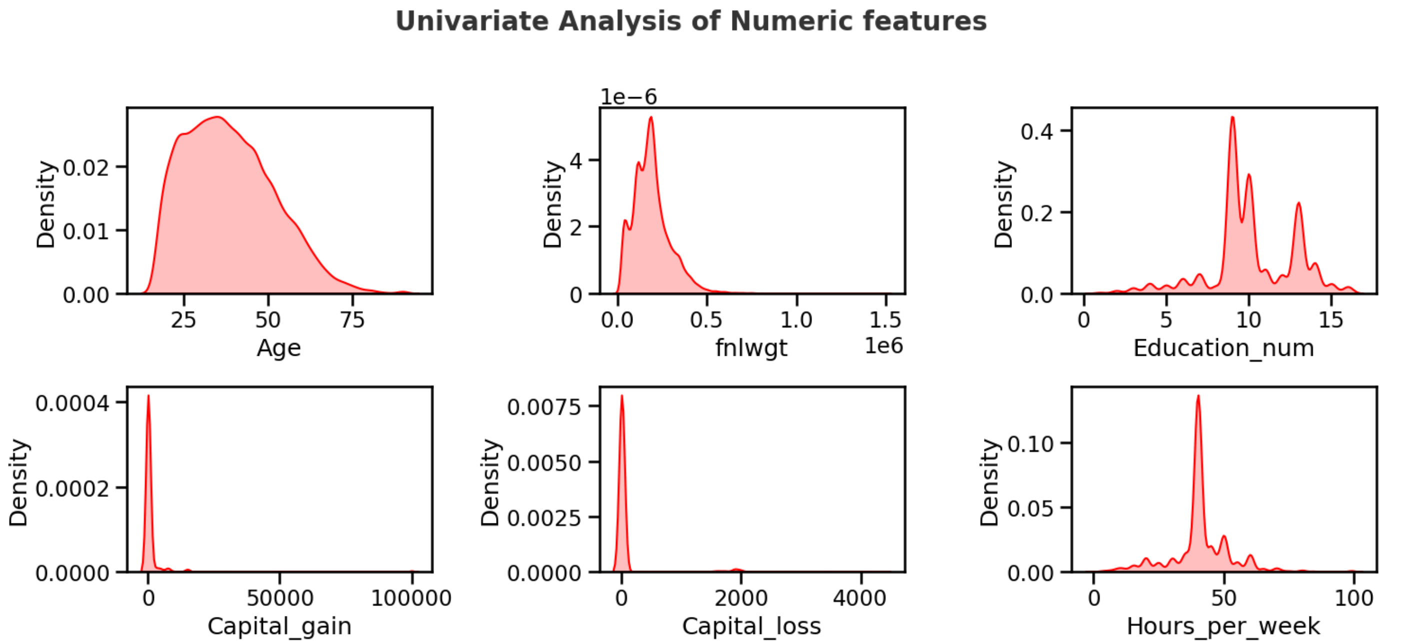


In [84]:
```
plt.figure(figsize=(12, 8))
sns.countplot(x='Relationship', data=df, palette='viridis')
plt.title('Distribution of Relationship')
plt.xlabel('Relationship')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```



Univariate analysis for numeric features

In [85]:
```
plt.figure(figsize=(15,15))
plt.suptitle('Univariate Analysis of Numeric features',fontsize=20, fontweight='bold',al

for i in range(0,len(numeric_features)):
    plt.subplot(5,3,i+1)
    sns.kdeplot(data=df[numeric_features[i]],shade=True,color='r')
    plt.xlabel(numeric_features[i])
    plt.tight_layout()
```



In [86]:
```
# Checking for skewness
df['Age'].skew()
```

Out[86]:
```
0.531904913753565
```

The skewness of data is 0.5319 means data are nearly symmetrical

In [52]:
```
"""plt.figure(figsize=(10, 6))
sns.boxplot(x='Class', y='Age', data=df)
plt.title('Age Distribution by Income')
plt.xlabel('Income')
plt.ylabel('Age')
plt.show()"""
```

Out[52]:
```
"plt.figure(figsize=(10, 6))\nsns.boxplot(x='Class', y='Age', data=df)\nplt.title('Age D
istribution by Income')\nplt.xlabel('Income')\nplt.ylabel('Age')\nplt.show()"
```

Bivariate Analysis

Which Eduction category is having Highest Capital_gain?

In [87]:
```
df_edu = df.groupby('Education')['Capital_gain'].sum().sort_values(ascending=False).rese
```

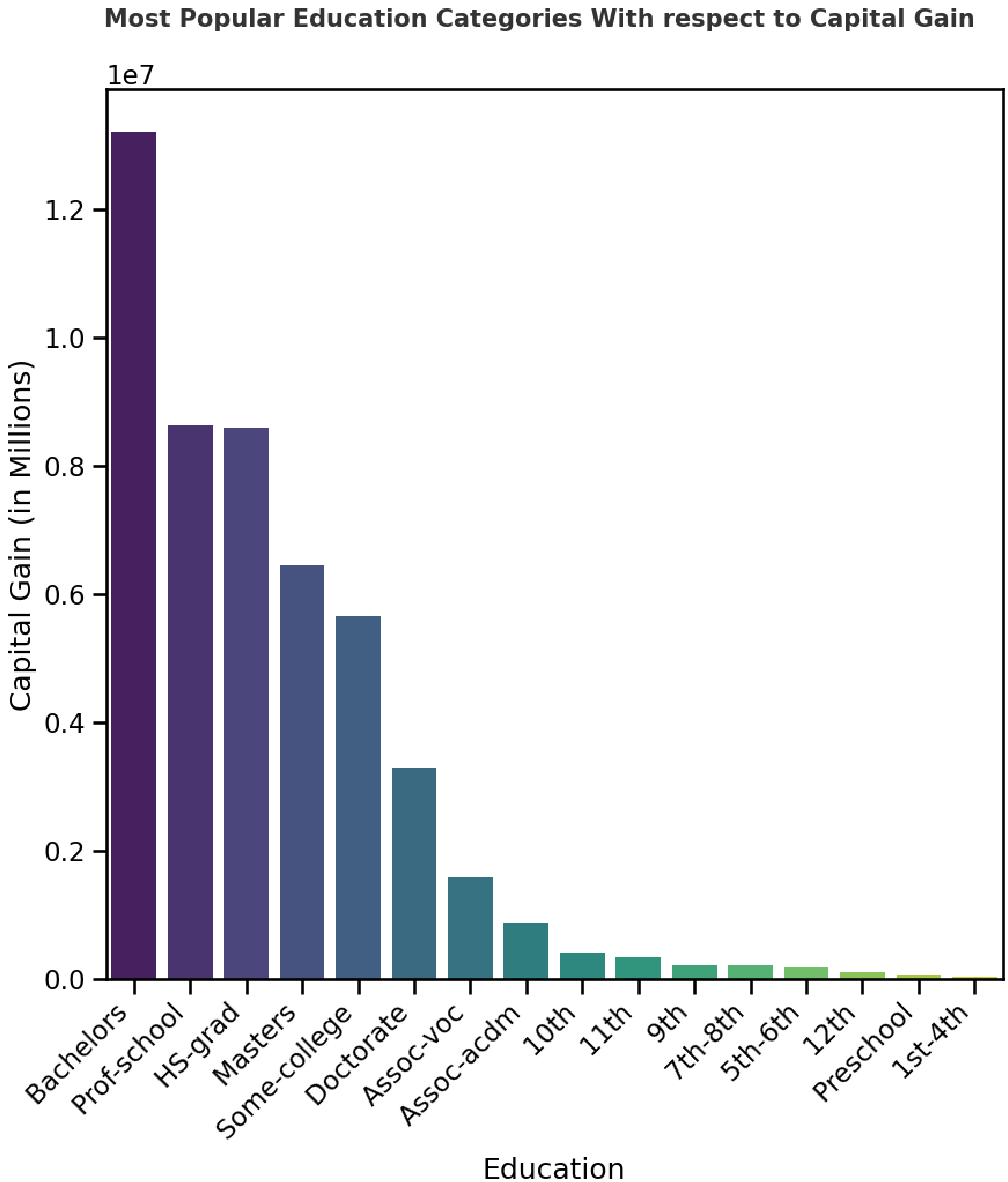
In [88]:
```
df_edu
```

Out[88]:

	Education	Capital_gain
0	Bachelors	13206072
1	Prof-school	8620249
2	HS-grad	8596846
3	Masters	6453814

4	Some-college	5646811
5	Doctorate	3301004
6	Assoc-voc	1577271
7	Assoc-acdm	862675
8	10th	395898
9	11th	344052
10	9th	221246
11	7th-8th	207701
12	5th-6th	173782
13	12th	114121
14	Preschool	60756
15	1st-4th	26585

In [89]:
```
plt.figure(figsize=(10, 10))
sns.set_context('talk')

sns.barplot(x='Education', y='Capital_gain', data=df_edu, ci=None, palette='viridis')

plt.suptitle('Most Popular Education Categories With respect to Capital Gain', fontsize=
plt.ylabel('Capital Gain (in Millions)')
plt.xlabel('Education')
plt.xticks(rotation=45, ha='right')  # Adjust rotation for better visibility

plt.show()
```



Observation

1. Bachelors degree ranks as the most prevalent education level in terms of capital gain.
2. Following closely, Prof-school degree secures the second-highest position in capital gain.
3. HS-grad claims the third spot in popularity concerning capital gain.
4. Masters degree takes the fourth position in terms of capital gain.
5. Some-college follows as the fifth most popular education level with respect to capital gain.

6. Doctorate degree holds the sixth position among the most popular education levels in relation to capital gain.

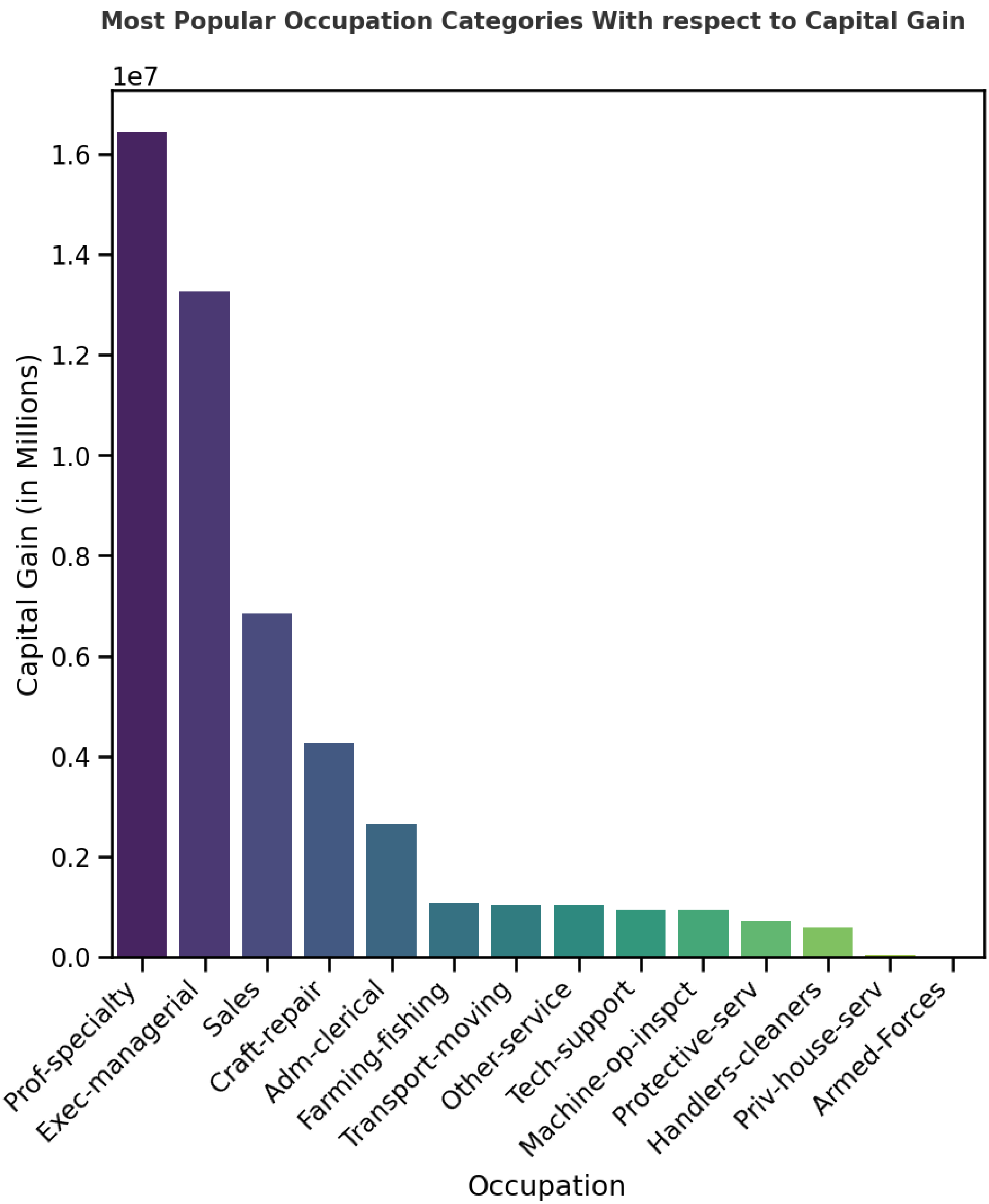
In [90]:
```
# Relationship between Occupation and Capital gain
df_Occptn = df.groupby('Occupation')['Capital_gain'].sum().sort_values(ascending=False).
```

In [91]:
```
df_Occptn
```

Out[91]:

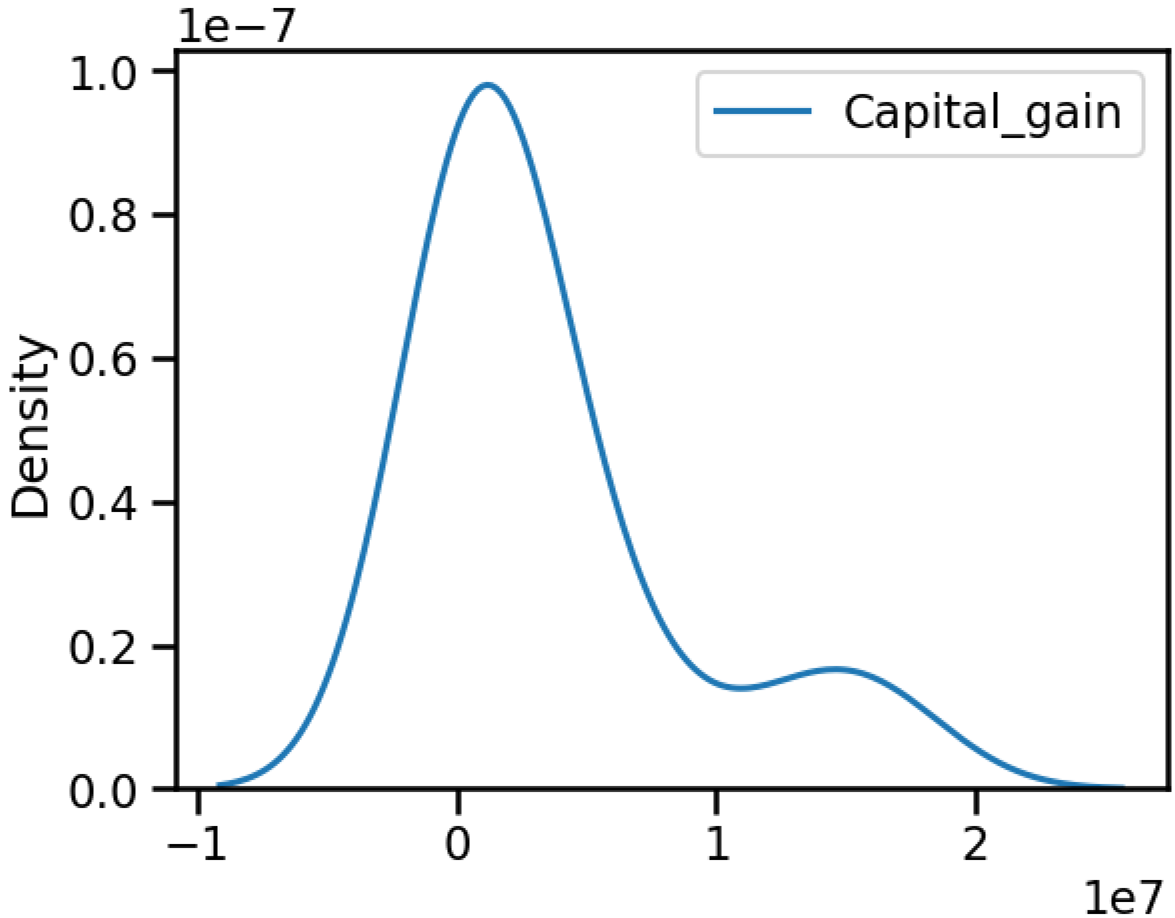
	Occupation	Capital_gain
0	Prof-specialty	16462086
1	Exec-managerial	13264248
2	Sales	6845789
3	Craft-repair	4261508
4	Adm-clerical	2636552
5	Farming-fishing	1066747
6	Transport-moving	1039836
7	Other-service	1022218
8	Tech-support	938848
9	Machine-op-inspct	931222
10	Protective-serv	704876
11	Handlers-cleaners	582845
12	Priv-house-serv	44810
13	Armed-Forces	7298

In [94]:
```
plt.figure(figsize=(10, 10))
sns.set_context('talk')

sns.barplot(x='Occupation', y='Capital_gain', data=df_Occptn, ci=None, palette='viridis'

plt.suptitle('Most Popular Occupation Categories With respect to Capital Gain', fontsize
plt.ylabel('Capital Gain (in Millions)')
plt.xlabel('Occupation')
plt.xticks(rotation=45, ha='right')   # Adjust rotation for better visibility

plt.show()
```



In [95]:
```
sns.kdeplot(df_Occptn)
```

Out[95]:
```
<Axes: ylabel='Density'>
```

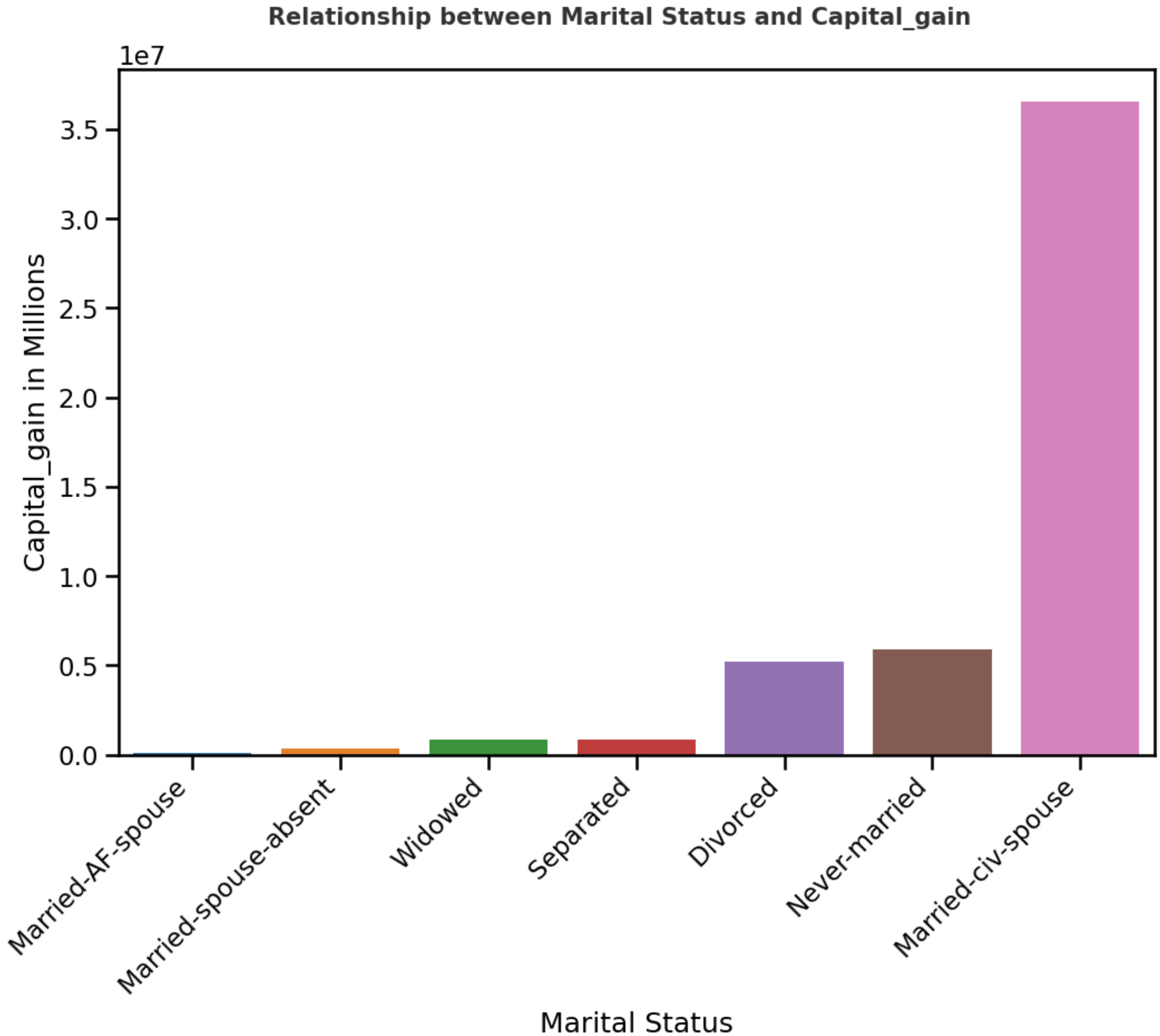


In [98]:
```
df_marital = df.groupby('Marital_Status')['Capital_gain'].sum().sort_values().reset_inde
df_marital
```

Out[98]:

	Marital_Status	Capital_gain
0	Married-AF-spouse	107297
1	Married-spouse-absent	364841
2	Widowed	824365
3	Separated	879148
4	Divorced	5195657
5	Never-married	5875449
6	Married-civ-spouse	36562126

In [101…
```
plt.figure(figsize=(12,8))
sns.set_context('talk')
sns.barplot(x='Marital_Status',y='Capital_gain',data=df_marital,ci=None)
plt.suptitle('Relationship between Marital Status and Capital_gain', fontsize=15, fontwe
plt.ylabel('Capital_gain in Millions')
plt.xlabel('Marital Status')
plt.xticks(rotation=45, ha='right')
plt.show()
```



Observations:

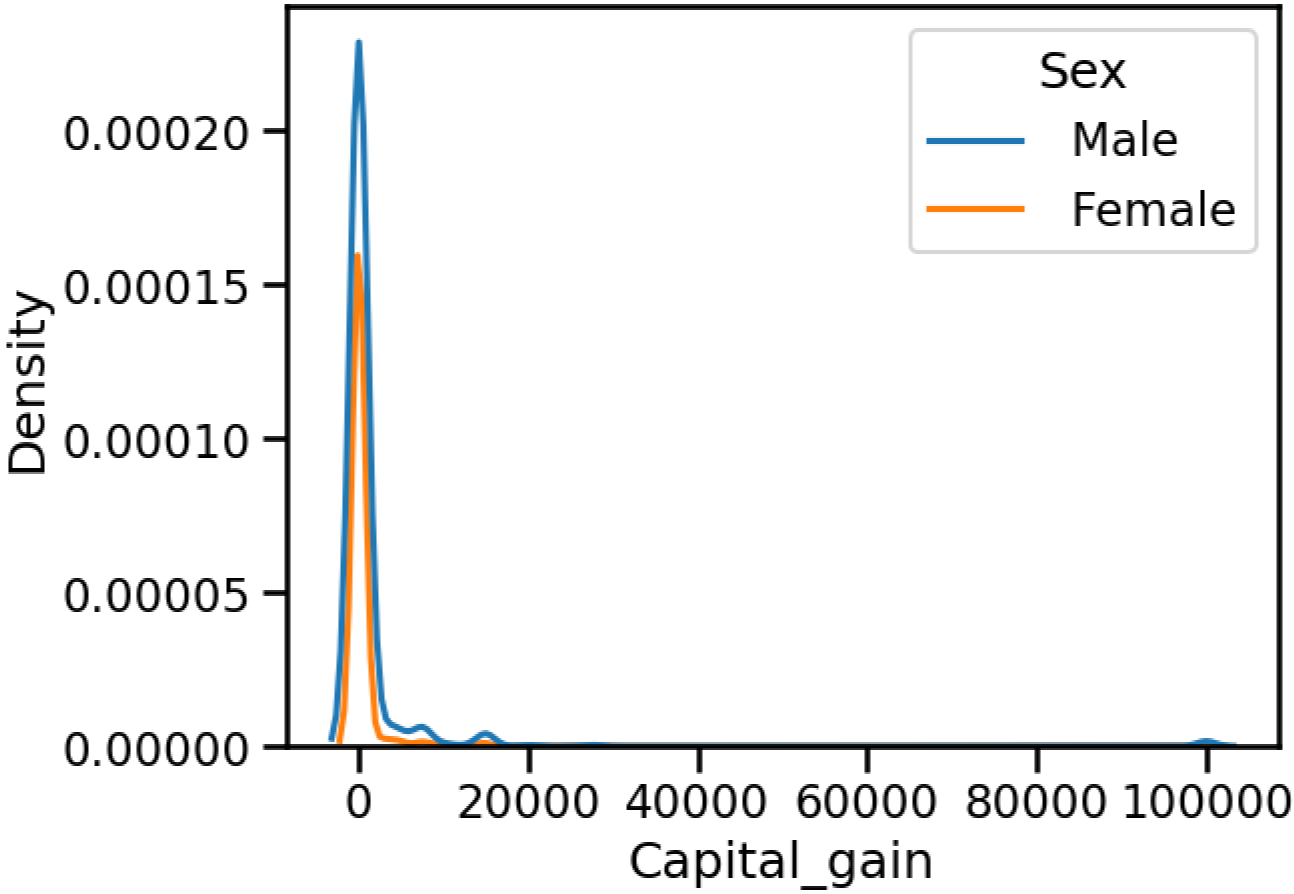
1. The category with the highest capital gain is "Married-civ-spouse."
2. Capital gains for individuals in the "Widowed" and "Separated" categories are notably lower compared to those in the "Married-civ-spouse" category.
3. Individuals in the "Divorced" and "Never-married" categories fall between those who are "Married-civ-spouse" and those who are "Widowed" or "Separated" in terms of their capital gain. This suggests an intermediate standing for capital gain within the spectrum of marital statuses, with "Divorced" and "Never-married" serving as intermediary points between the extremes represented by "Married-civ-spouse" and "Widowed" or "Separated."
4. Individuals identified as "Married-AF-spouse" exhibit the lowest capital gain among the specified marital statuses.

In [102…

```
# comparing capital gain between male & female
sns.kdeplot(x='Capital_gain', data=df,hue='Sex')
```

Out[102]:

```
<Axes: xlabel='Capital_gain', ylabel='Density'>
```

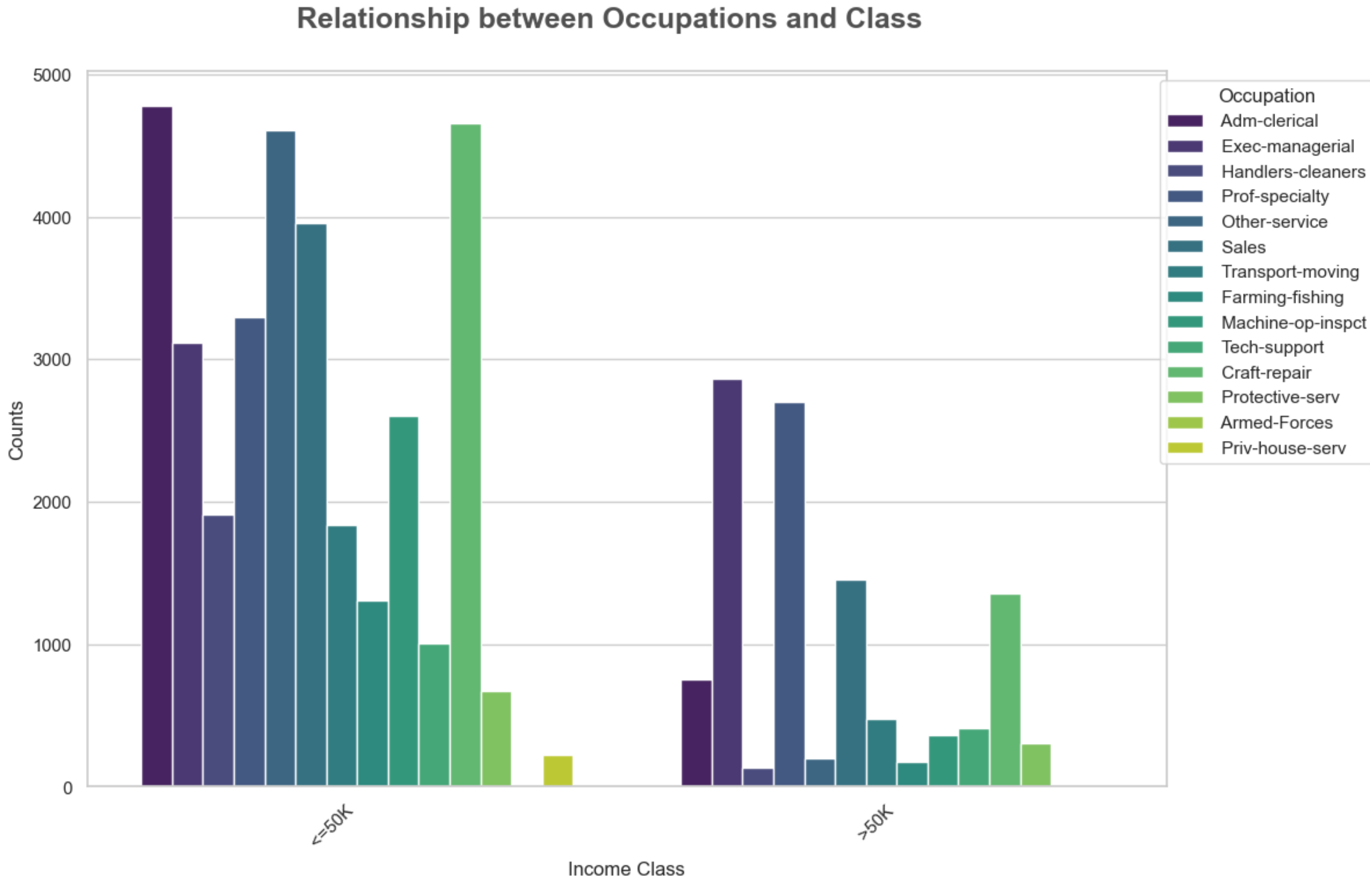


In [107…

```
plt.figure(figsize=(10, 6))
sns.set(style="whitegrid")

sns.countplot(x='Class', hue='Sex', data=df, palette='Set2')

plt.title('Gender-wise Income Distribution', fontsize=16, fontweight='bold')
plt.xlabel('Income Class')
plt.ylabel('Count')
plt.legend(title='Gender', loc='upper right', labels=['Male', 'Female'])
plt.xticks(rotation=0)  # Adjust rotation for better visibility

plt.show()
```

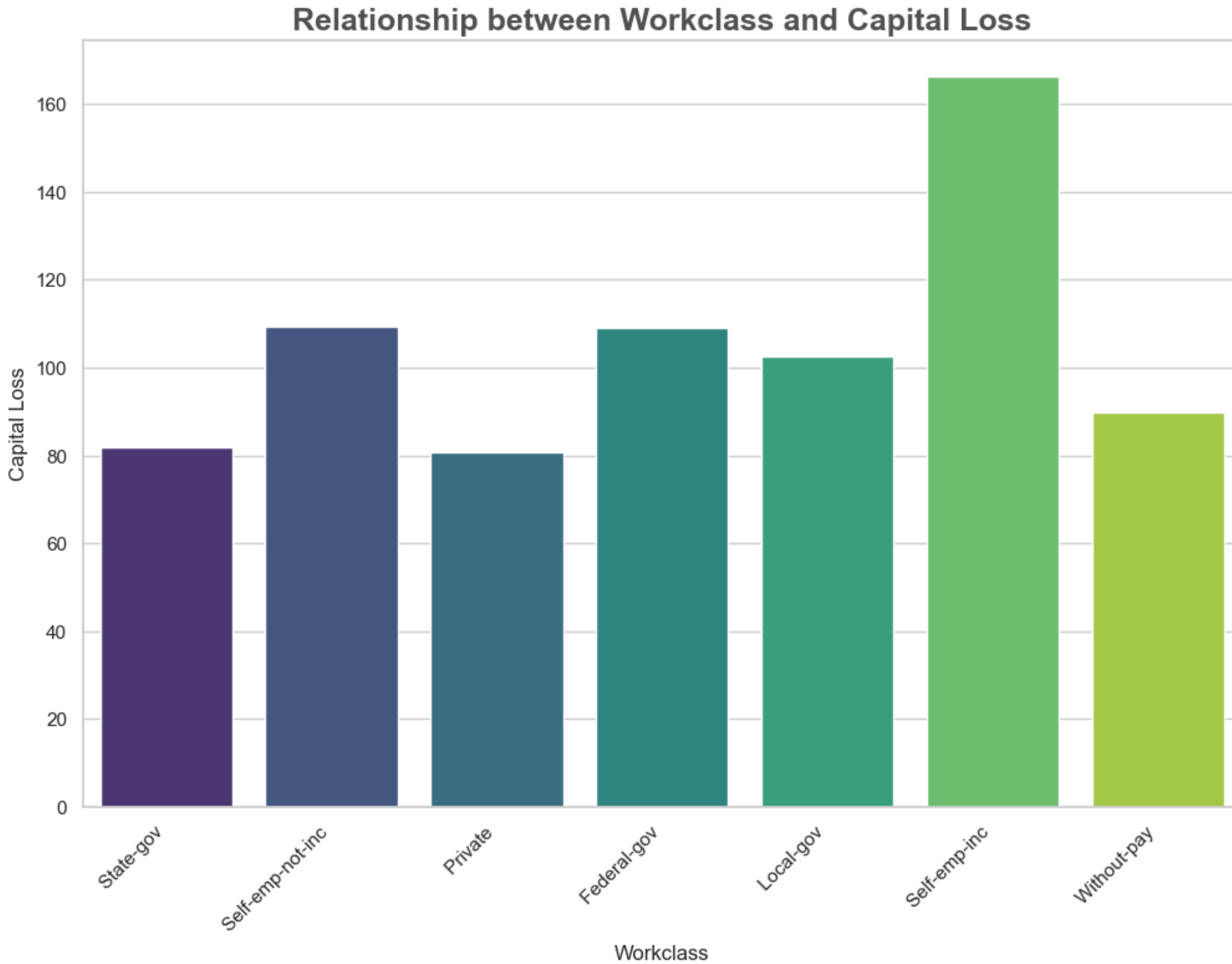
In [110…

```
plt.figure(figsize=(12, 8))
sns.set_context('talk')
sns.set(style="whitegrid")

ax = sns.countplot(x='Class', hue='Occupation', data=df, palette='viridis', dodge=True)

plt.suptitle('Relationship between Occupations and Class', fontsize=18, fontweight='bold
plt.ylabel('Counts')
plt.xlabel('Income Class')
plt.xticks(rotation=45, ha='right')  # Adjust rotation for better visibility
plt.legend(title='Occupation', loc='upper right', bbox_to_anchor=(1.2, 1))

plt.show()
```



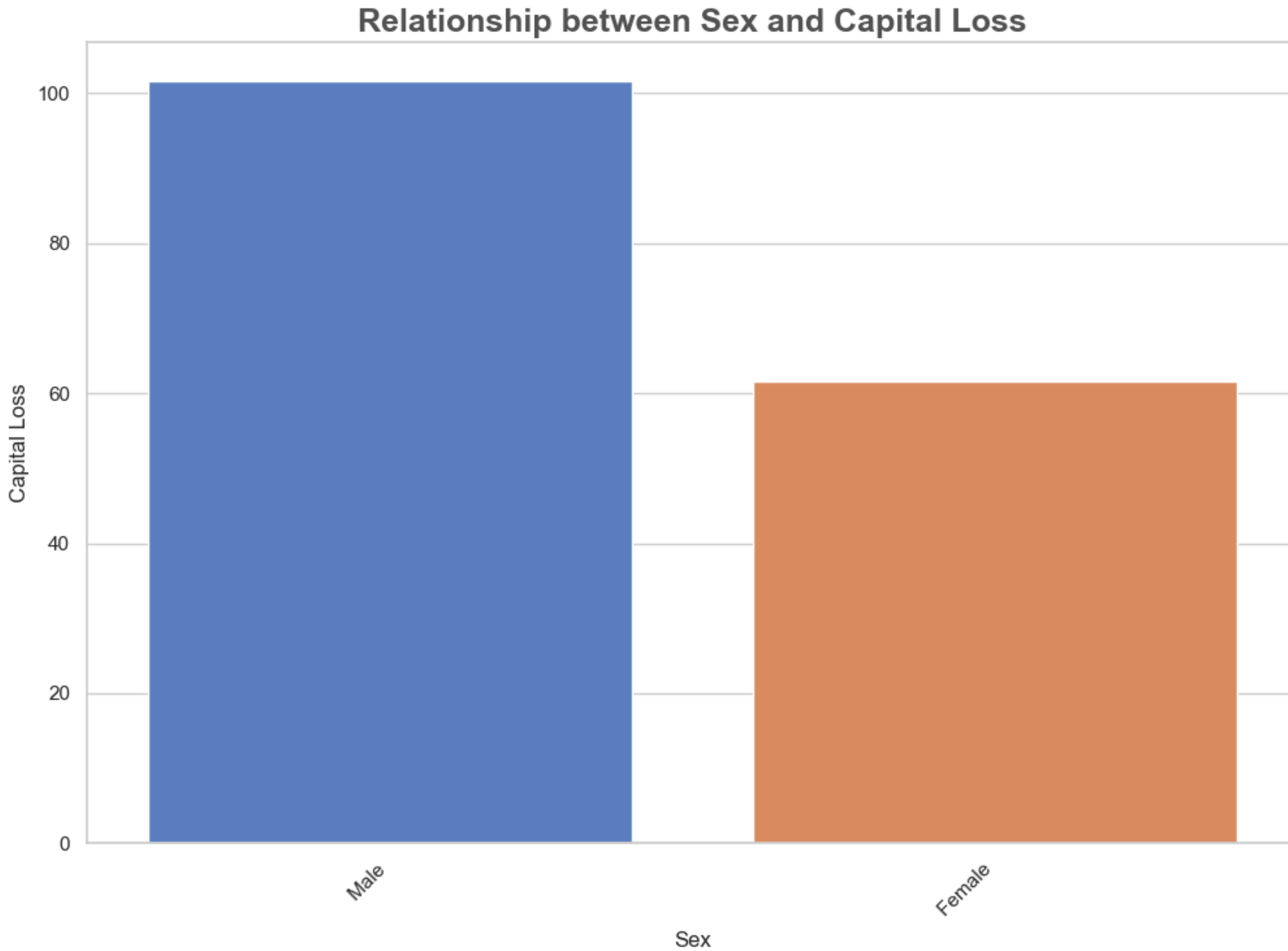
In [8]:

```
# Relation between workclass and Capital-loss

plt.figure(figsize=(12,8))
sns.set_context('talk')
sns.set(style='whitegrid')

sns.barplot(x='Workclass',y='Capital_loss',data=df,ci=None,palette='viridis')
plt.title('Relationship between Workclass and Capital Loss', fontsize=18, fontweight='bo
plt.xlabel('Workclass')
plt.ylabel('Capital Loss')
plt.xticks(rotation=45, ha='right')
plt.show()
```



In [ ]:

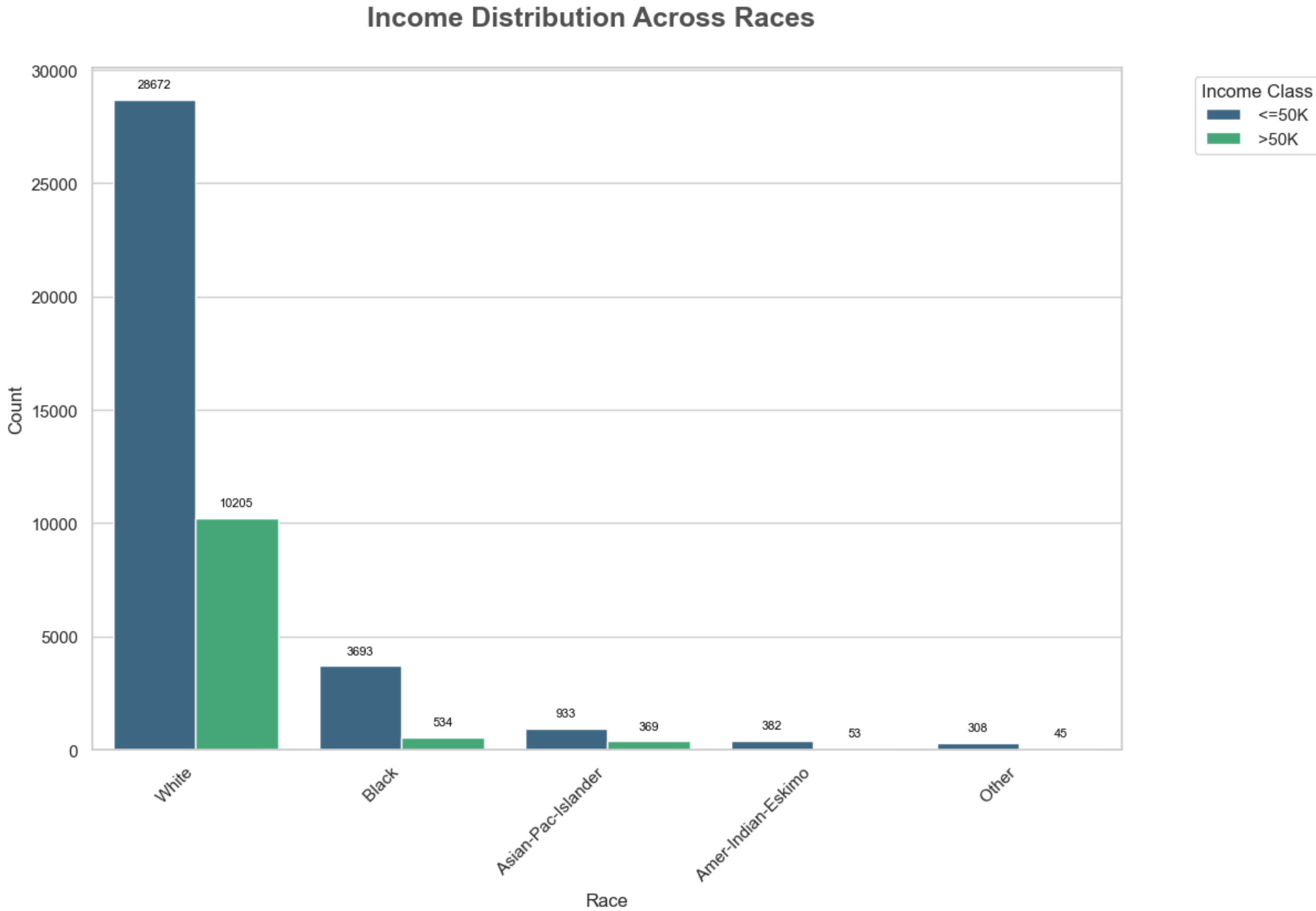
In [12]:
```
# Relation between sex and capital loss

plt.figure(figsize=(12,8))
sns.set_context('talk')
sns.set(style='whitegrid')

sns.barplot(x='Sex',y='Capital_loss',data=df,ci=None,palette='muted')
plt.title('Relationship between Sex and Capital Loss', fontsize=18, fontweight='bold', a
plt.xlabel('Sex')
plt.ylabel('Capital Loss')
plt.xticks(rotation=45, ha='right')
plt.show()
```



Observations:

The capital loss for males is higher than that for females, indicating a notable disparity in financial impact between the two genders.

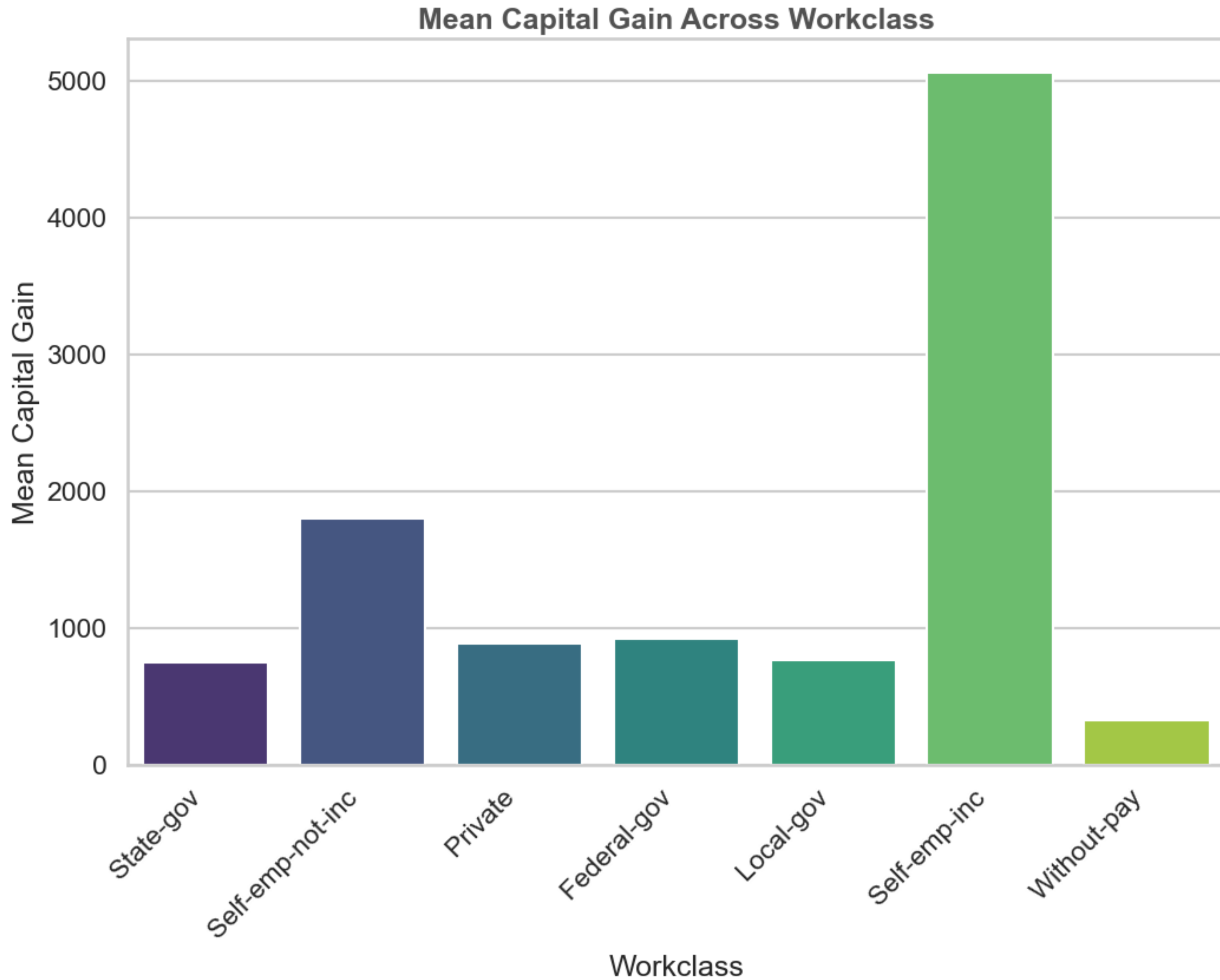
In [24]:
```
plt.figure(figsize=(12, 8))
sns.set_context('talk')
sns.set(style="whitegrid")

ax = sns.countplot(x='Race', hue='Class', data=df, palette='viridis', dodge=True)

plt.suptitle('Income Distribution Across Races', fontsize=18, fontweight='bold', alpha=0
plt.ylabel('Count')
plt.xlabel('Race')
plt.xticks(rotation=45, ha='right')  # Adjust rotation for better visibility
plt.legend(title='Income Class', loc='upper right', bbox_to_anchor=(1.2, 1))

# Adding data labels on top of the bars
for p in ax.patches:
    ax.annotate(f'{p.get_height():.0f}', (p.get_x() + p.get_width() / 2., p.get_height()
                ha='center', va='center', xytext=(0, 10), textcoords='offset points', fo

plt.show()
```

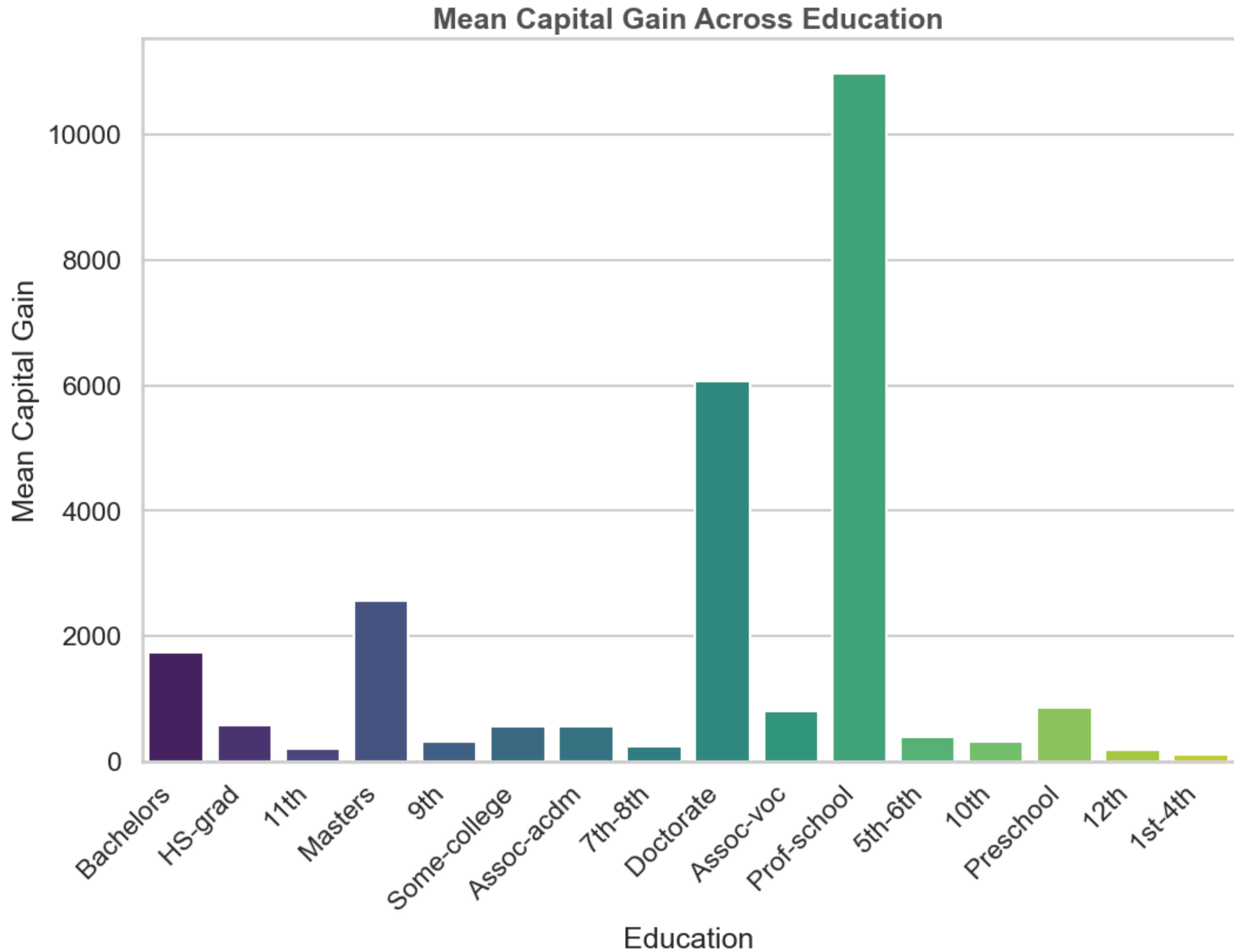


Observations:

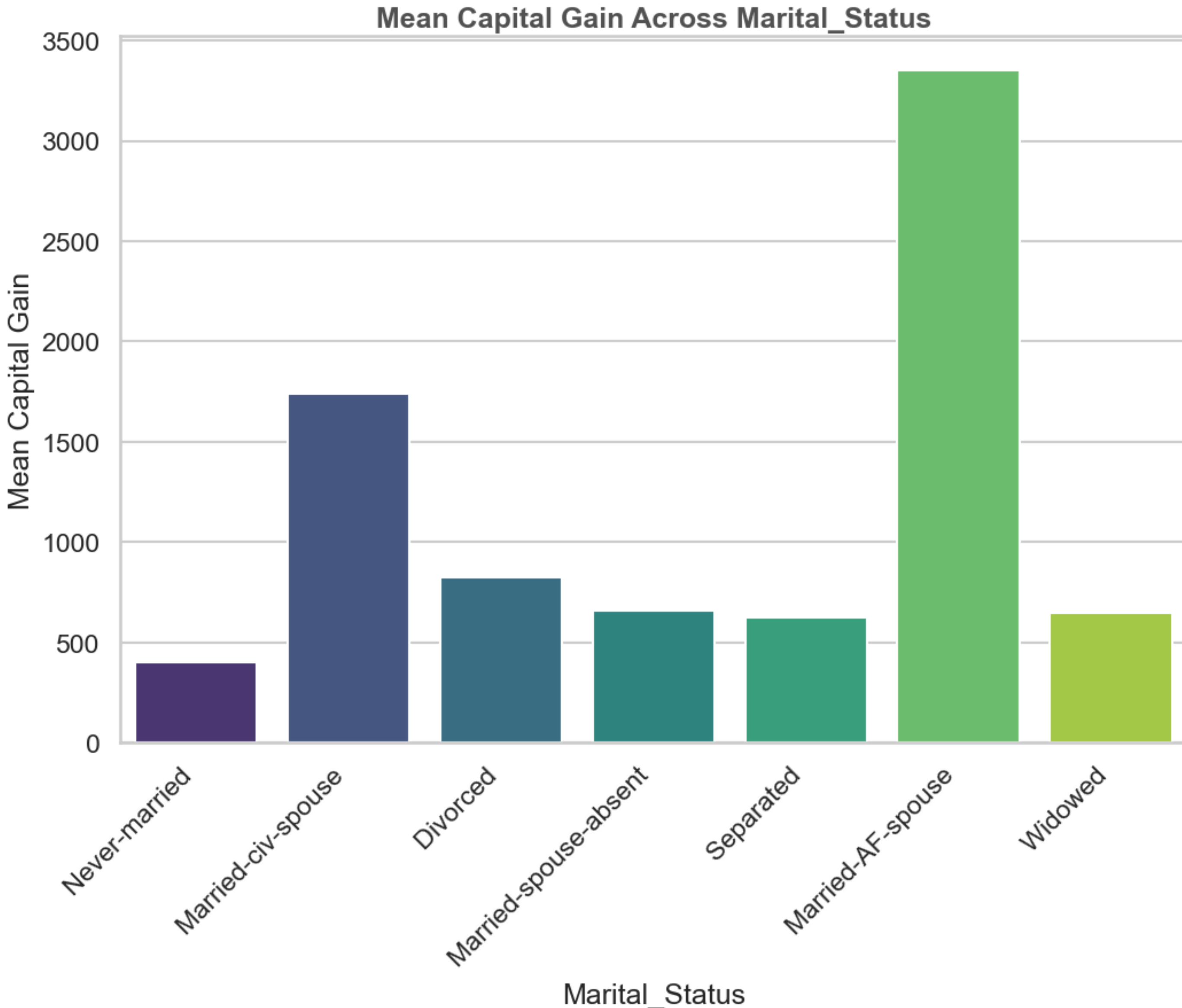
1. The category with the highest number of individuals of White ethnicity is observed to have income both less than or equal to 50k and greater than 50k.
2. Following White ethnicity, individuals of Black ethnicity show the highest counts in both income categorie —those earning less than or equa to $50k an those earning more than 50k.
3. Individuals classified as Asian-Pac-Islander, Amer-Indian-Eskimo, and those falling into the "Other" category display lower incomes compared to both White and Black ethnicities.

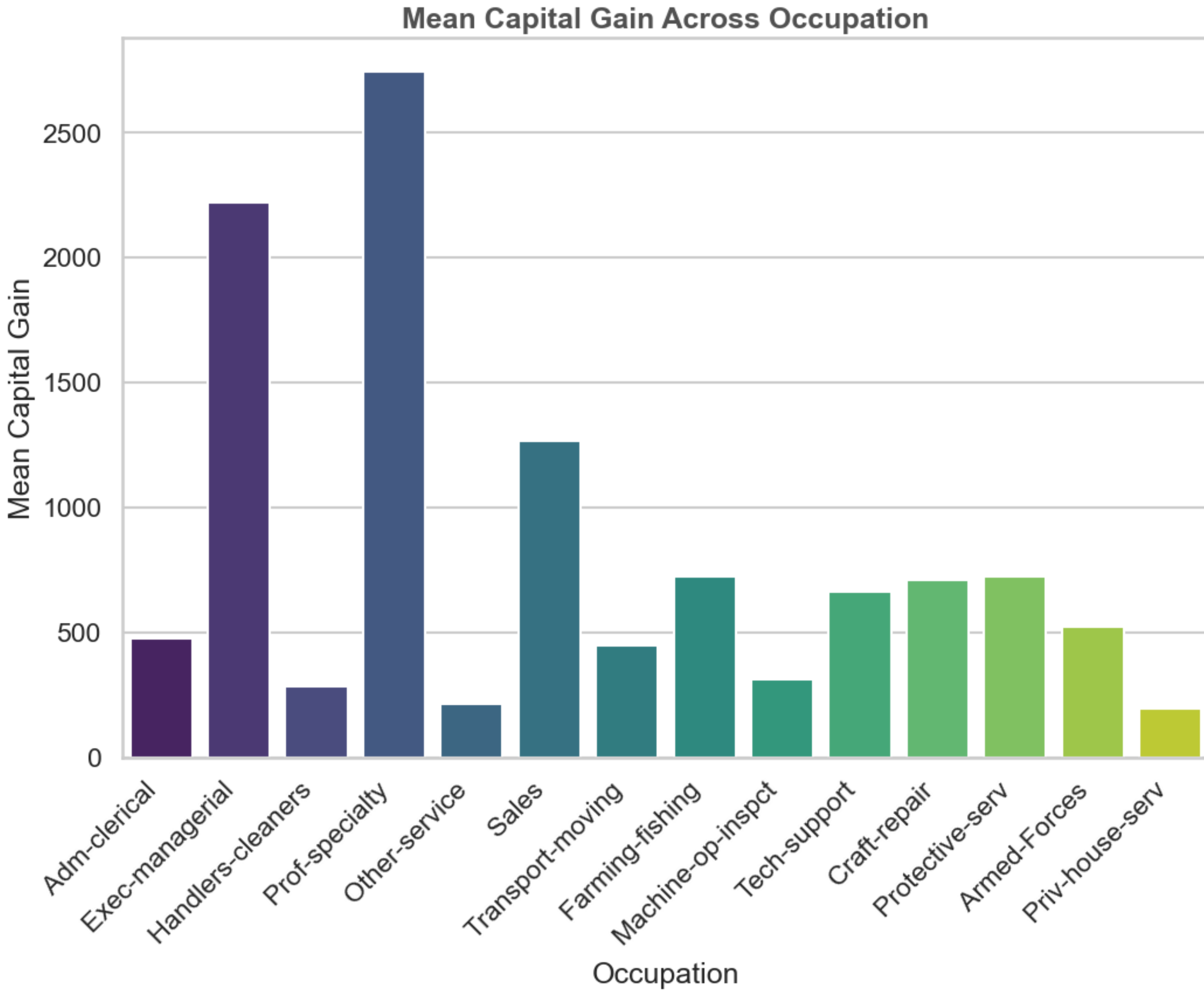
In [22]:
```
## find the relationship between categorical feature and Capital gain
for feature in categorical_features:
    plt.figure(figsize=(12, 8))
    sns.set_context('talk')

    sns.barplot(x=feature, y='Capital_gain', data=df, ci=None, palette='viridis')

    plt.xlabel(feature)
    plt.ylabel('Mean Capital Gain')
    plt.title(f'Mean Capital Gain Across {feature}', fontsize=18, fontweight='bold', alp
    plt.xticks(rotation=45, ha='right')  # Adjust rotation for better visibility

    plt.show()
```


Mean Capital Gain Across Workclass
Mean Capital Gain
0
1000
2000
3000
4000
5000
State-gov
Self-emp-not-inc
Private
Federal-gov
Local-gov
Self-emp-inc
Without-pay
Workclass

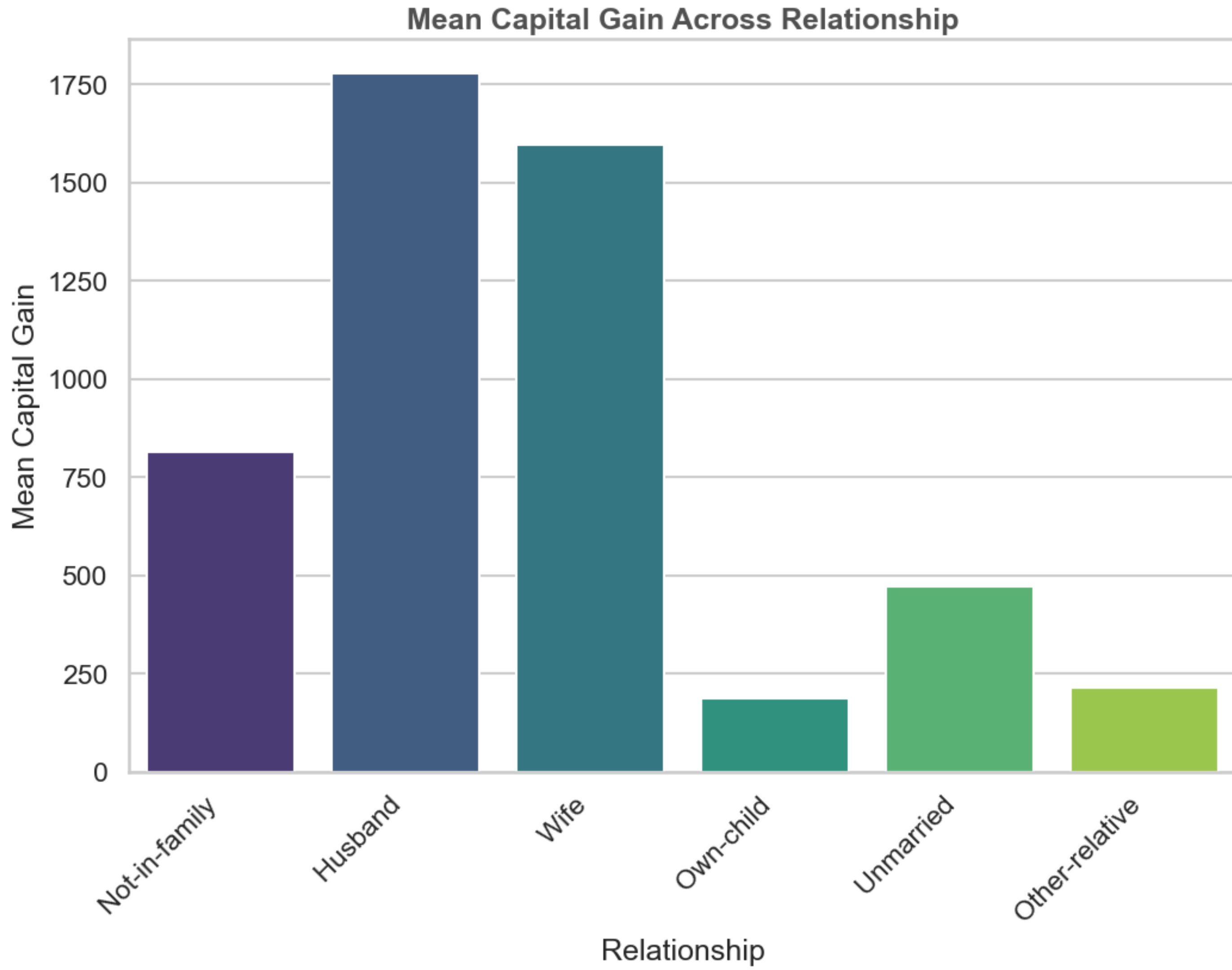


Mean Capital Gain Across Education
Mean Capital Gain
10000
8000
6000
4000
2000
0
Bachelors
HS-grad
11th
Masters
9th
Some-college
Assoc-acdm
7th-8th
Doctorate
Assoc-voc
Prof-school
5th-6th
10th
Preschool
12th
1st-4th
Education

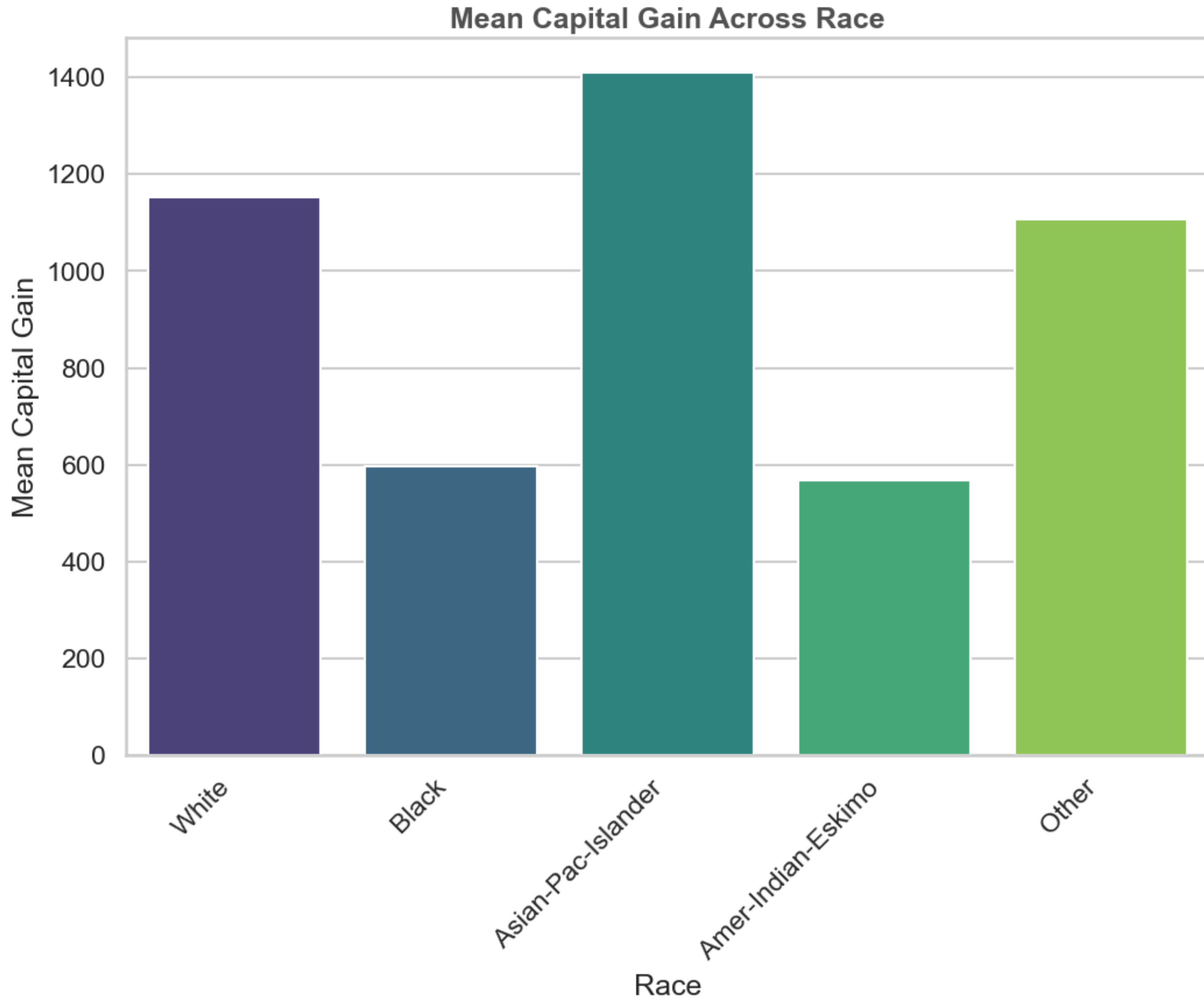

Mean Capital Gain Across Marital_Status
Mean Capital Gain
3500
3000
2500
2000
1500
1000
500
0
Never-married
Married-civ-spouse
Divorced
Married-spouse-absent
Separated
Married-AF-spouse
Widowed
Marital_Status



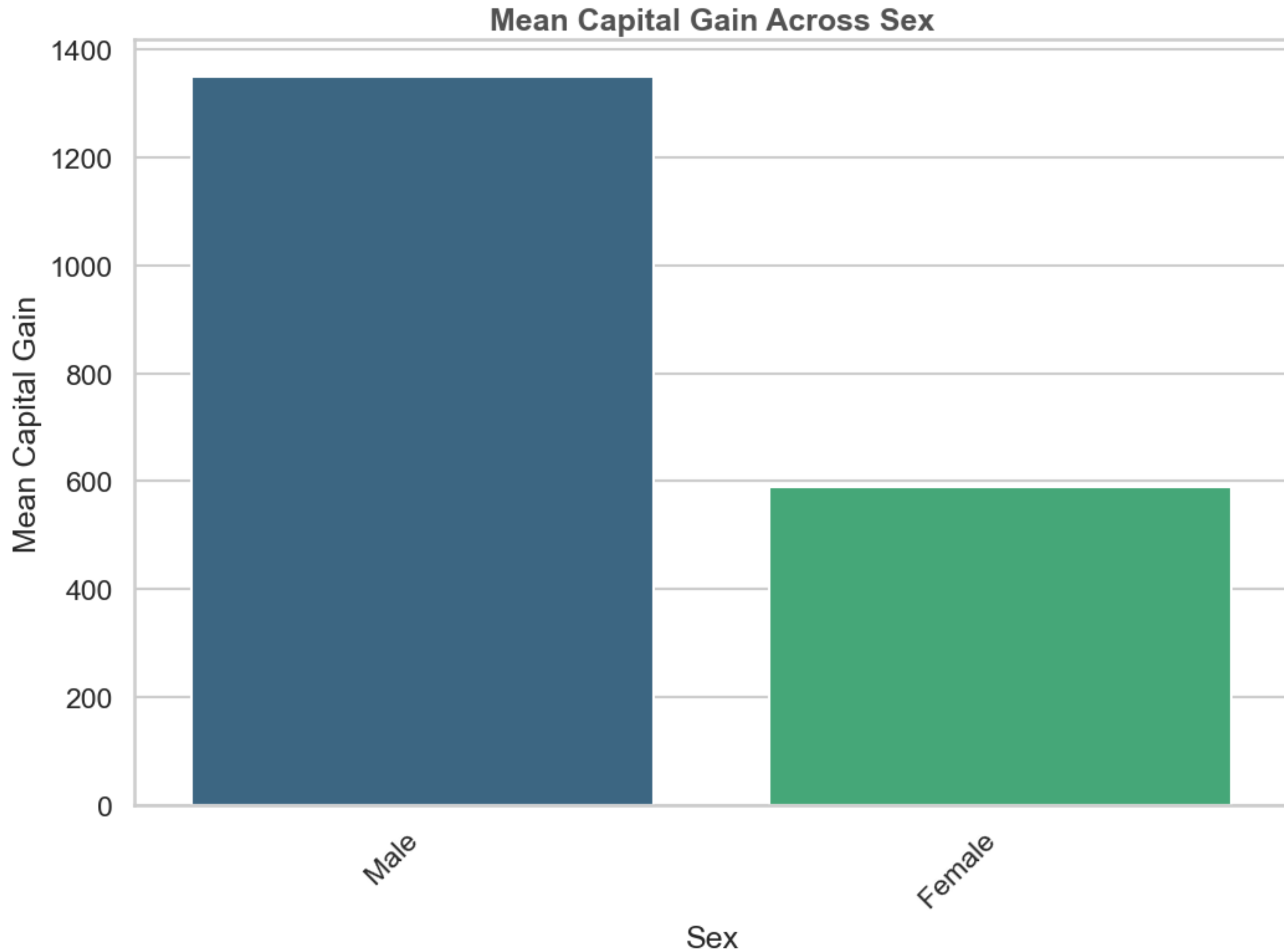
Mean Capital Gain Across Occupation
Mean Capital Gain
0
500
1000
1500
2000
2500
Adm-clerical
Exec-managerial
Handlers-cleaners
Prof-specialty
Other-service
Sales
Transport-moving
Farming-fishing
Machine-op-inspct
Tech-support
Craft-repair
Protective-serv
Armed-Forces
Priv-house-serv
Occupation

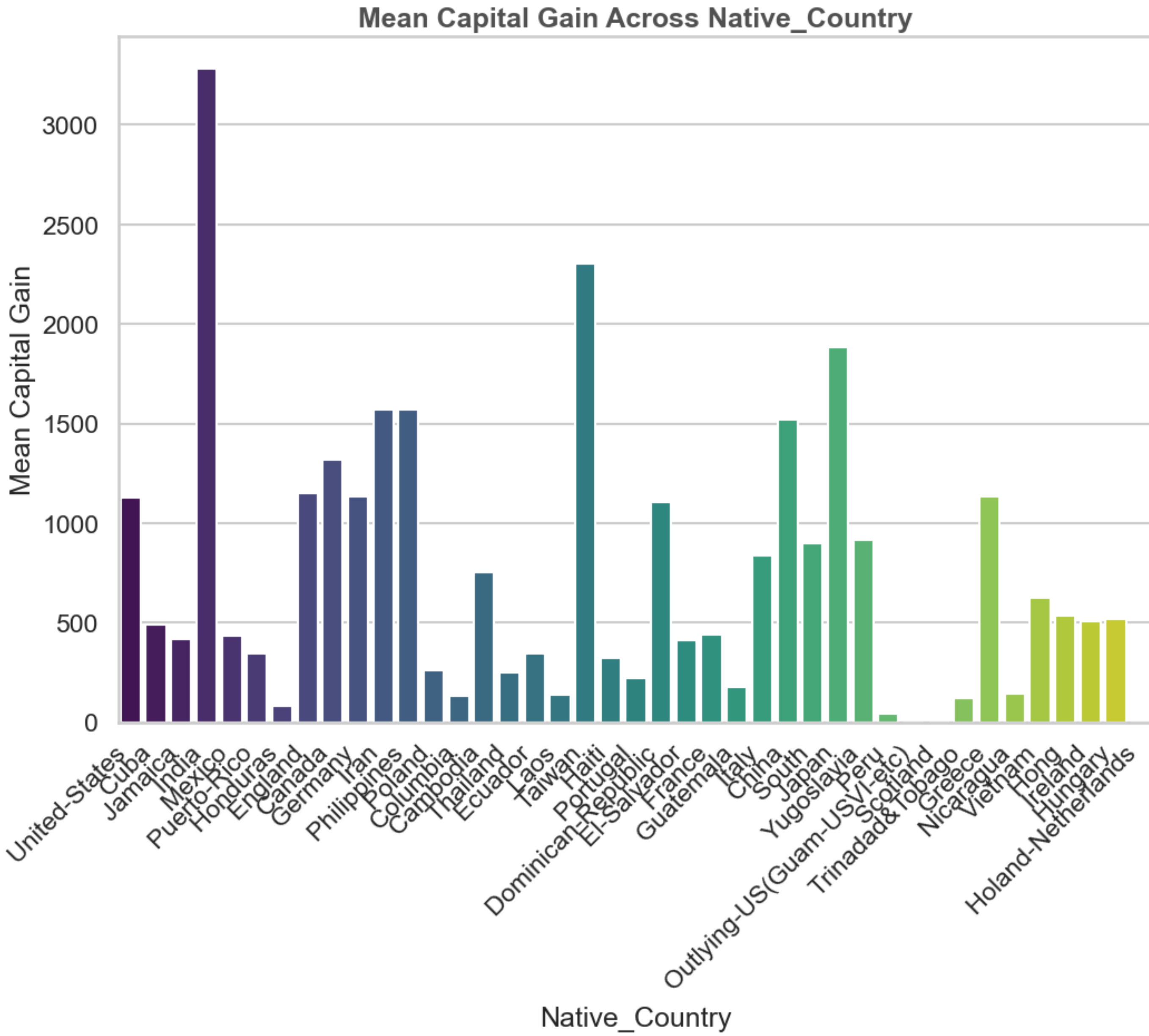


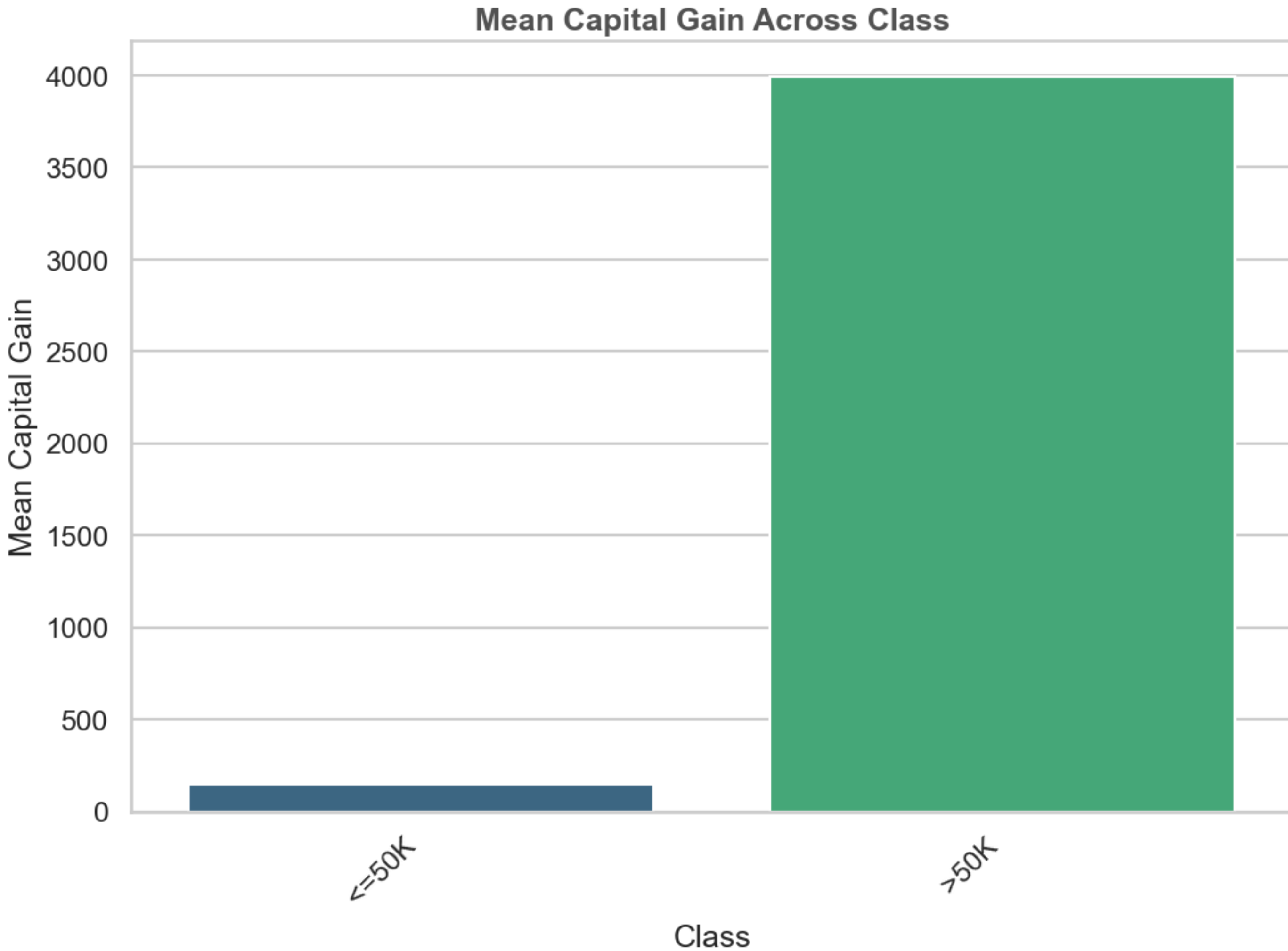
Mean Capital Gain Across Relationship
Mean Capital Gain
1750
1500
1250
1000
750
500
250
0
Not-in-family
Husband
Wife
Own-child
Unmarried
Other-relative
Relationship



Mean Capital Gain Across Race
Mean Capital Gain
0
200
400
600
800
1000
1200
1400
White
Black
Asian-Pac-Islander
Amer-Indian-Eskimo
Other
Race

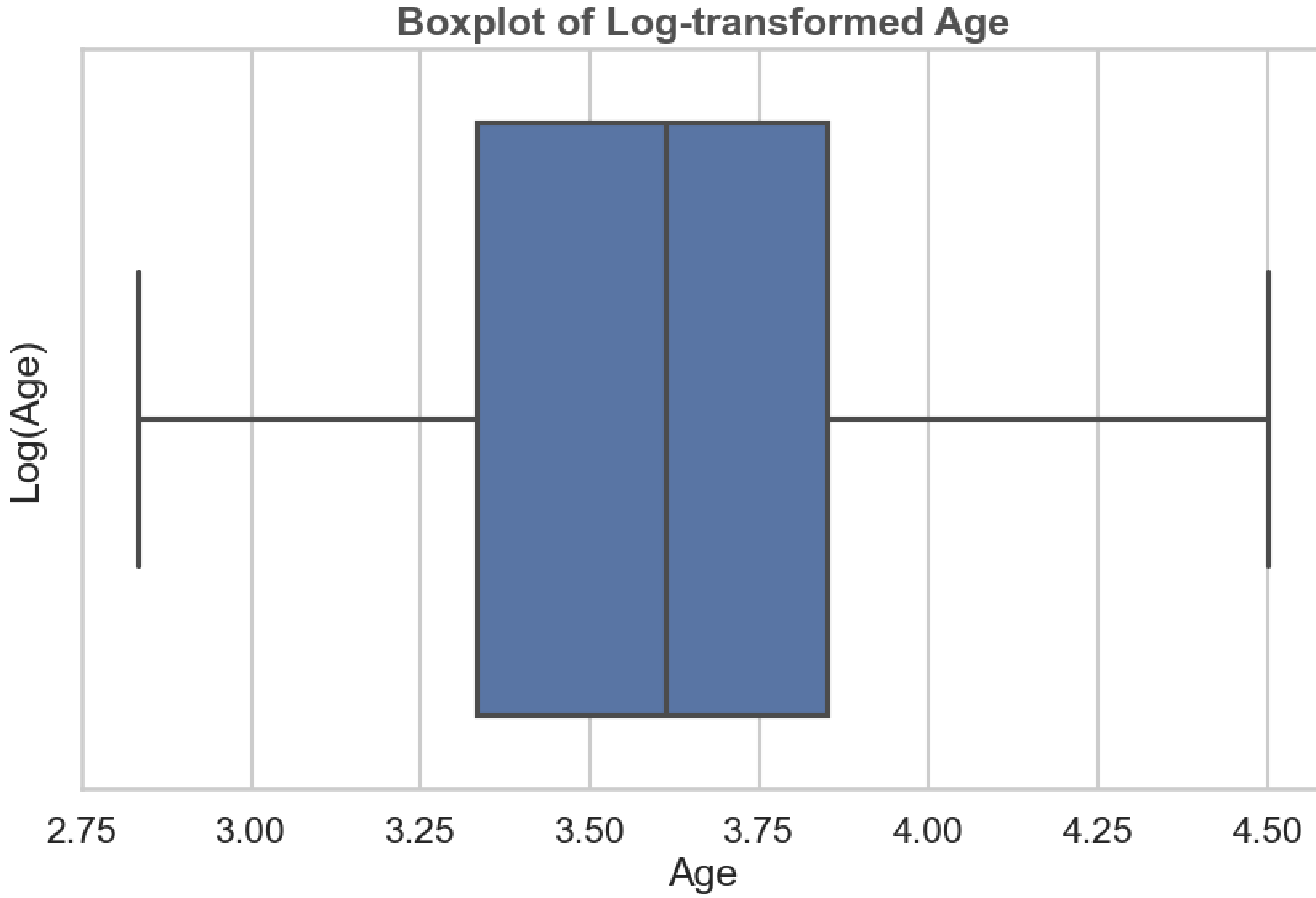

Mean Capital Gain Across Sex
Mean Capital Gain
1400
1200
1000
800
600
400
200
0
Male
Female
Sex

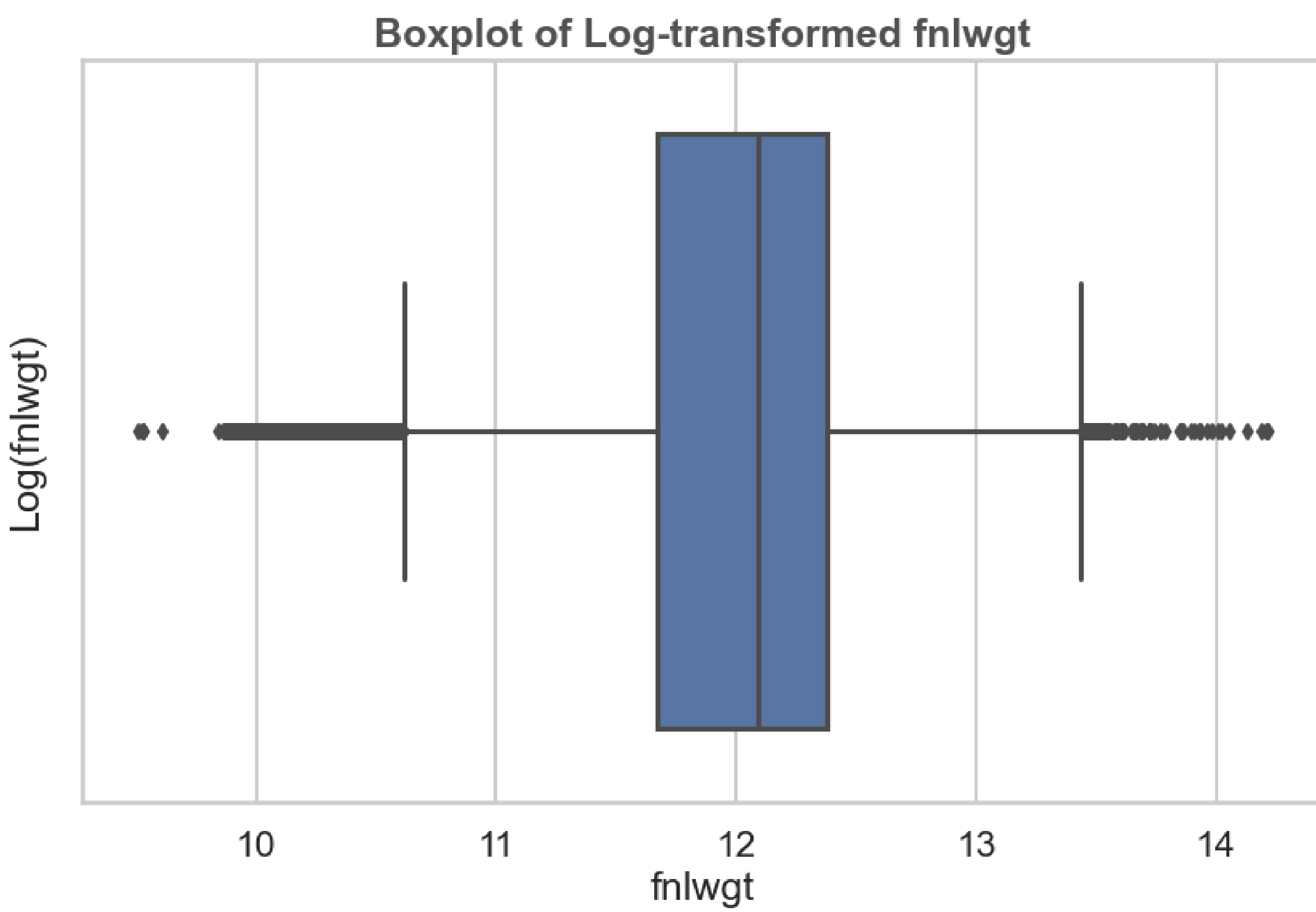

Mean Capital Gain Across Native_Country
Mean Capital Gain
0
500
1000
1500
2000
2500
3000
United-States
Cuba
Jamaica
India
Mexico
Puerto-Rico
Honduras
England
Canada
Germany
Iran
Philippines
Poland
Columbia
Cambodia
Thailand
Ecuador
Laos
Taiwan
Haiti
Portugal
Dominican-Republic
El-Salvador
France
Guatemala
Italy
China
South
Japan
Yugoslavia
Peru
Outlying-US(Guam-USVI-etc)
Scotland
Trinadad&Tobago
Greece
Nicaragua
Vietnam
Hong
Ireland
Hungary
Holand-Netherlands
Native_Country

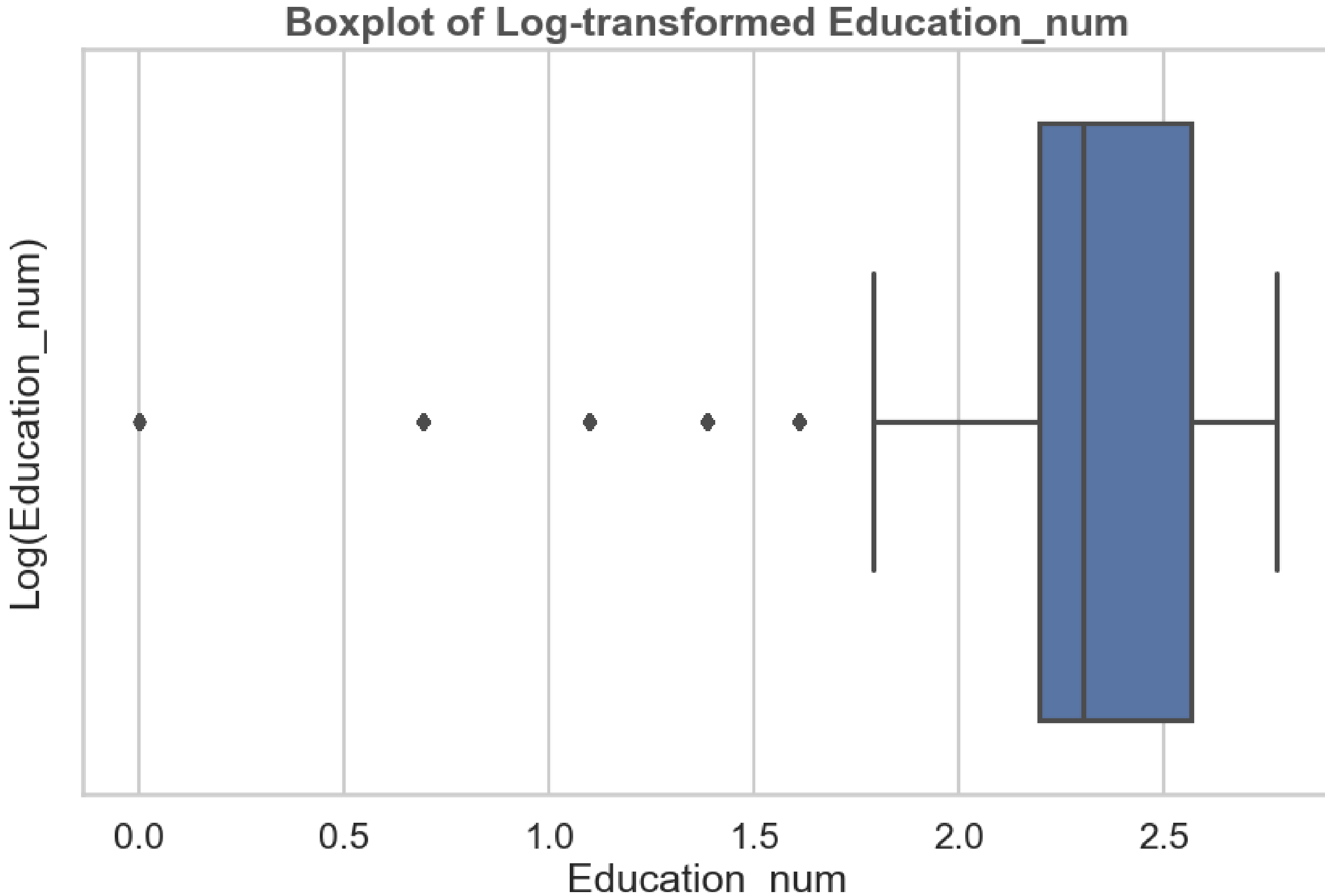


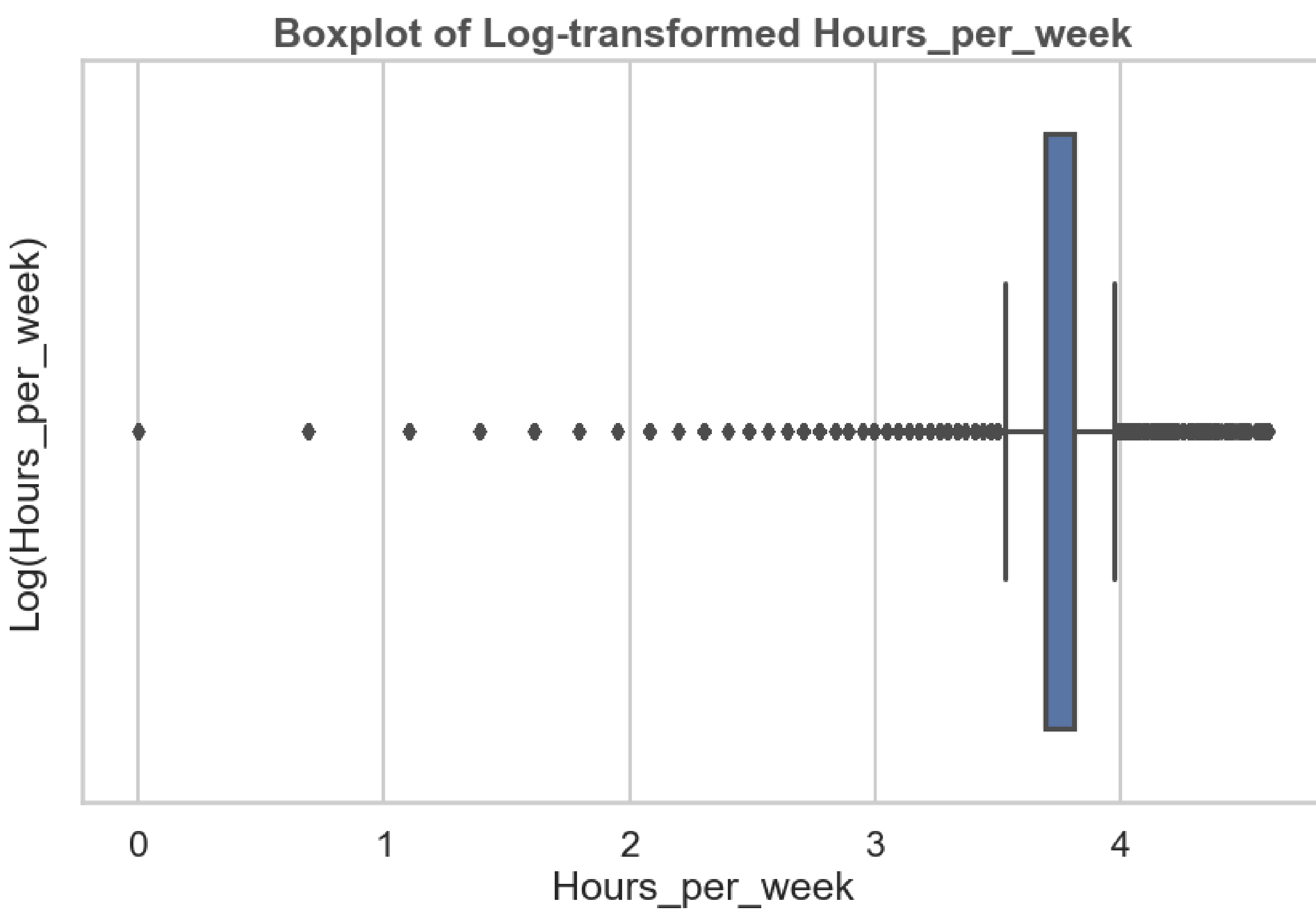
Outliers

In [23]:
```
# Outlier

for feature in numeric_features:
    if 0 in df[feature].unique():
        pass
    else:
        plt.figure(figsize=(10, 6))
        sns.set_context('talk')

        # Apply logarithmic transformation to the feature
        df[feature] = np.log(df[feature])

        sns.boxplot(x=df[feature])

        plt.ylabel(f'Log({feature})')
        plt.title(f'Boxplot of Log-transformed {feature}', fontsize=18, fontweight='bold

        plt.show()
```


Boxplot of Log-transformed Age
Log(Age)
2.75
3.00
3.25
3.50
3.75
4.00
4.25
4.50
Age


Boxplot of Log-transformed fnlwgt
Log(fnlwgt)
10
11
12
13
14
fnlwgt





Observations:

The 'hours-per-week' and 'final weight' categories exhibit the highest prevalence of outliers in the dataset.

Key Insights from data:

1. The majority of individuals are employed in the private sector, with private employment significantly outnumbering other sectors. It is evident that approximately 80% of the population is engaged in private-sector work. A significant income gap exists among individuals working in the private sector, with a notably higher count having incomes exceeding 50k compared to those earning 50k or less.

2. Within the Self-emp-inc category, there are more individuals earning over 50k than those earning 50k or less.

3. The highest number of individuals possess an HS-grad degree, followed by Some-college degree, and Bachelors degree, respectively.

4. A substantial disparity is evident between individuals earning less than or equal to 50k and those earning more than 50k. The population with an income of $50k or less significantly exceeds the count of individuals with incomes exceeding 50k.

5. The count of males significantly exceeds the count of females. The number of males with incomes less than or equal to 50k surpasses the corresponding count of females, and similarly, for incomes greater than 50k, the count of males exceeds that of females.

6. The capital loss for males is higher than that for females, indicating a notable disparity in financial impact between the two genders

7. Bachelors degree ranks as the most prevalent education level in terms of capital gain.

8. The Marital_status category with the highest capital gain is "Married-civ-spouse. "9

9. The category with the highest number of individuals of White ethnicity is observed to have income both less than or equal to 50k and greater than 50k. Following White ethnicity, individuals of Black ethnicity show the highest counts in both income categorie —those earning less than or equa t $50k an those earning more than 50

10. The 'hours-per-week' and 'final weight' categories exhibit the highest prevalence of outliers in the dataset. . .

s in the dataset.

```
In [ ]:
```