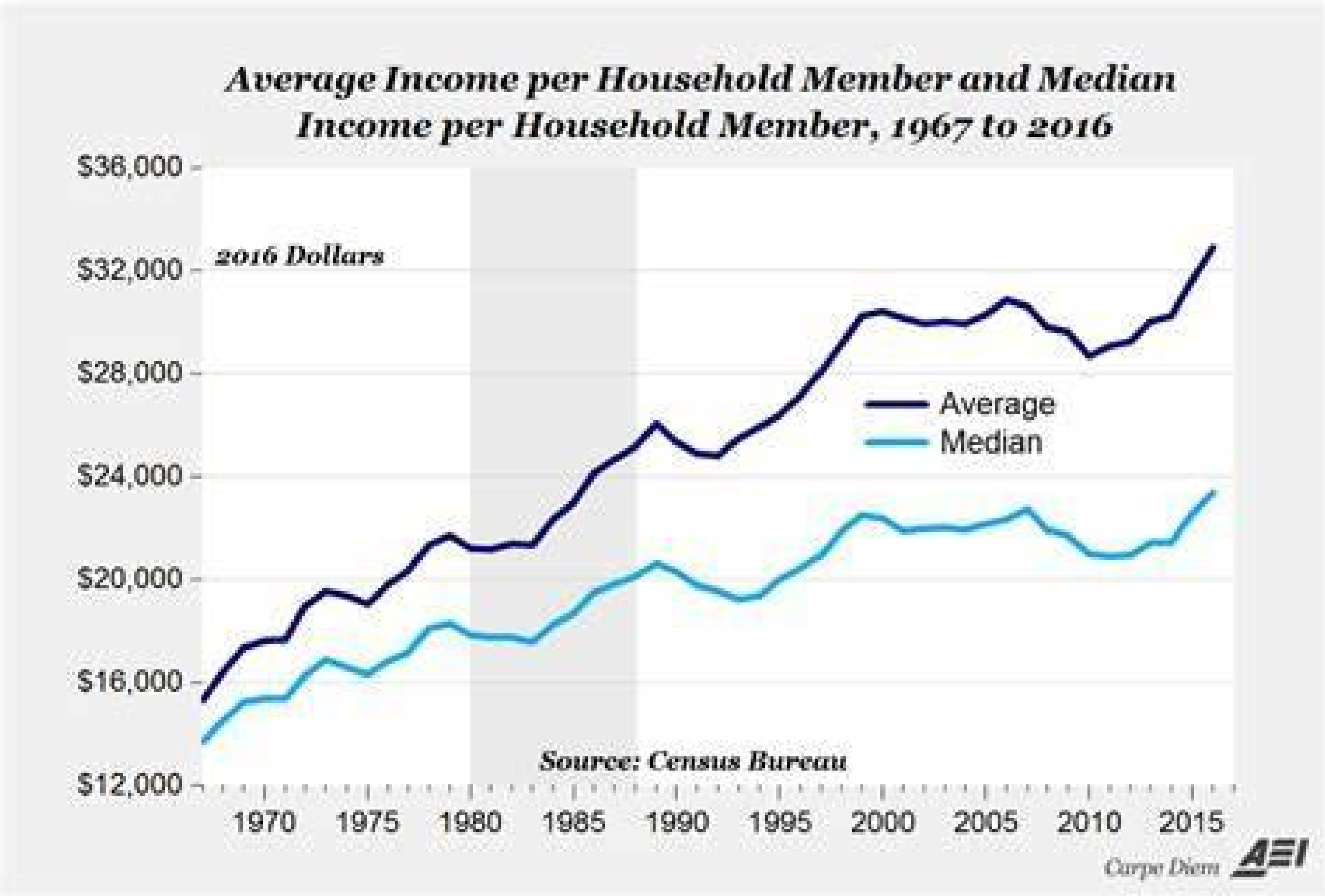
0 Census Income EDA



0.0.1 Prepared By:

Bankole Ayoade

GitHub: [https://github.com/acaaattunde2012 (https://github.com/acaaattunde2012)](https://github.com/acaaattunde2012)

LinkedIn: [https://www.linkedin.com/in/bankole-ayoade-fca-acti-cipfa-161406a7/ (https://www.linkedin.com/in/bankole-ayoade-fca-acti-cipfa-161406a7/)](https://www.linkedin.com/in/bankole-ayoade-fca-acti-cipfa-161406a7/)

0.1 About Census Income Dataset

The dataset comprises 32561 rows and 15 coulmns as follows:

- income_class

```
50K, <=50K.
```

- age: continuous.
- workclass: Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.

- native-country: United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands.

0.2 Importing Dependencies

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
import plotly.express as px
import plotly.graph_objects as go
import plotly.io as pio
pio.templates.default = "plotly_white"

warnings.filterwarnings("ignore")

%matplotlib inline
```

0.3 Data Ingestion:

In [2]:

```
url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data'
```

In [3]:

```
income_df=pd.read_csv(url, header=None)
#income_df=pd.read_csv("adultdata.csv", header=None)
```

In [6]:

```
income_df.head(5)
```

Out[6]:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40	United-States	<=50K
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13	United-States	<=50K
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40	United-States	<=50K
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0	40	United-States	<=50K
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0	40	Cuba	<=50K

In [7]:

```
# Insert column names
col_name = ["age", "workclass","fnlwgt","education","education-num","marital-status","occupation","relationship","race",
            "sex","capital_gain","capital_loss","hour_per_week","native_country","income_class"]
income_df.columns = col_name
income_df.head(5)
```

Out[7]:

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	capital_gain	capital_loss	hour_per_week
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0	40
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0	40

In [8]:

```
# Make a copy of the dataset
income = income_df.copy()
```

0.4 Data Profiling

In [9]:

```
# Check the first 5 rows
income.head(5)
```

Out[9]:

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	capital_gain	capital_loss	hour_per_week
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0	40
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0	40

In [10]:

```
# Check the last 5 rows
income.tail(5)
```

Out[10]:

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	capital_gain	capital_loss	hour_per_w
32556	27	Private	257302	Assoc-acdm	12	Married-civ-spouse	Tech-support	Wife	White	Female	0	0	
32557	40	Private	154374	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	White	Male	0	0	
32558	58	Private	151910	HS-grad	9	Widowed	Adm-clerical	Unmarried	White	Female	0	0	
32559	22	Private	201490	HS-grad	9	Never-married	Adm-clerical	Own-child	White	Male	0	0	
32560	52	Self-emp-inc	287927	HS-grad	9	Married-civ-spouse	Exec-managerial	Wife	White	Female	15024	0	

In [11]:

```
# Insert column names
col_name = ["age", "workclass","fnlwgt","education","education-num","marital-status","occupation","relationship","race",
           "sex","capital_gain","capital_loss","hour_per_week","native_country","income_class"]
income.columns = col_name
income.head(5)
```

Out[11]:

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	capital_gain	capital_loss	hour_per_week
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0	40
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0	40

In [ ]:

In [12]:

```
# Show the no of rows and columns
income.shape
```

Out[12]:

```
(32561, 15)
```

In [13]:

```
# Show more info about the dataset
income.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  hour_per_week   32561 non-null  int64 
 13  native_country  32561 non-null  object
 14  income_class    32561 non-null  object
dtypes: int64(6), object(9)
memory usage: 3.7+ MB
```

In [14]:

```
# Checking missing values
income.isnull().sum()
```

Out[14]:

```
age               0
workclass         0
fnlwgt            0
education         0
education-num     0
marital-status    0
occupation        0
relationship      0
race              0
sex               0
capital_gain      0
capital_loss      0
hour_per_week     0
native_country    0
income_class      0
dtype: int64
```

In [15]:

```
# Checking missing values
income.isna().sum()
```

Out[15]:

```
age               0
workclass         0
fnlwgt            0
education         0
education-num     0
marital-status    0
occupation        0
relationship      0
race              0
sex               0
capital_gain      0
```

In [16]:

```
# Checking for duplicates
income[income.duplicated()]
```

Out[16]:

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	capital_gain	capital_loss	hour_per_
4881	25	Private	308144	Bachelors	13	Never-married	Craft-repair	Not-in-family	White	Male	0	0	
5104	90	Private	52386	Some-college	10	Never-married	Other-service	Not-in-family	Asian-Pac-Islander	Male	0	0	
9171	21	Private	250051	Some-college	10	Never-married	Prof-specialty	Own-child	White	Female	0	0	
11631	20	Private	107658	Some-college	10	Never-married	Tech-support	Not-in-family	White	Female	0	0	
13084	25	Private	195994	1st-4th	2	Never-married	Priv-house-serv	Not-in-family	White	Female	0	0	
15059	21	Private	243368	Preschool	1	Never-married	Farming-fishing	Not-in-family	White	Male	0	0	
17040	46	Private	173243	HS-grad	9	Married-civ-spouse	Craft-repair	Husband	White	Male	0	0	
18555	30	Private	144593	HS-grad	9	Never-married	Other-service	Not-in-family	Black	Male	0	0	
18698	19	Private	97261	HS-grad	9	Never-married	Farming-fishing	Not-in-family	White	Male	0	0	
21318	19	Private	138153	Some-college	10	Never-married	Adm-clerical	Own-child	White	Female	0	0	
21490	19	Private	146679	Some-college	10	Never-married	Exec-managerial	Own-child	Black	Male	0	0	
21875	49	Private	31267	7th-8th	4	Married-civ-spouse	Craft-repair	Husband	White	Male	0	0	
22300	25	Private	195994	1st-4th	2	Never-married	Priv-house-serv	Not-in-family	White	Female	0	0	
22367	44	Private	367749	Bachelors	13	Never-married	Prof-specialty	Not-in-family	White	Female	0	0	
22494	49	Self-emp-not-inc	43479	Some-college	10	Married-civ-spouse	Craft-repair	Husband	White	Male	0	0	
25872	23	Private	240137	5th-6th	3	Never-married	Handlers-cleaners	Not-in-family	White	Male	0	0	
26313	28	Private	274679	Masters	14	Never-married	Prof-specialty	Not-in-family	White	Male	0	0	
28230	27	Private	255582	HS-grad	9	Never-married	Machine-op-inspct	Not-in-family	White	Female	0	0	
28522	42	Private	204235	Some-college	10	Married-civ-spouse	Prof-specialty	Husband	White	Male	0	0	
28846	39	Private	30916	HS-grad	9	Married-civ-spouse	Craft-repair	Husband	White	Male	0	0	
29157	38	Private	207202	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	White	Male	0	0	
30845	46	Private	133616	Some-college	10	Divorced	Adm-clerical	Unmarried	White	Female	0	0	
31993	19	Private	251579	Some-college	10	Never-married	Other-service	Own-child	White	Male	0	0	

In [18]:

```
# Checking for unique values for each column
for col in income.columns:
    print(f"============================================{col}============================================================")
    print(income[col].unique())
    print("========================================================================================================")
```

```
============================================age============================================================
[39 50 38 53 28 37 49 52 31 42 30 23 32 40 34 25 43 54 35 59 56 19 20 45
 22 48 21 24 57 44 41 29 18 47 46 36 79 27 67 33 76 17 55 61 70 64 71 68
 66 51 58 26 60 90 75 65 77 62 63 80 72 74 69 73 81 78 88 82 83 84 85 86
 87]
============================================================================================
============================================workclass============================================================
[' State-gov' ' Self-emp-not-inc' ' Private' ' Federal-gov' ' Local-gov'
 ' ?' ' Self-emp-inc' ' Without-pay' ' Never-worked']
============================================================================================
============================================fnlwgt============================================================
[ 77516  83311 215646 ...  34066  84661 257302]
============================================================================================
============================================education============================================================
[' Bachelors' ' HS-grad' ' 11th' ' Masters' ' 9th' ' Some-college'
 ' Assoc-acdm' ' Assoc-voc' ' 7th-8th' ' Doctorate' ' Prof-school'
 ' 5th-6th' ' 10th' ' 1st-4th' ' Preschool' ' 12th']
============================================================================================
============================================education-num==========================================================
==
[13  9  7 14  5 10 12 11  4 16 15  3  6  2  1  8]
============================================================================================
============================================marital-status=========================================================
===
[' Never-married' ' Married-civ-spouse' ' Divorced'
 ' Married-spouse-absent' ' Separated' ' Married-AF-spouse' ' Widowed']
============================================================================================
============================================occupation============================================================
[' 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']
============================================================================================
============================================relationship===========================================================
=
[' Not-in-family' ' Husband' ' Wife' ' Own-child' ' Unmarried'
 ' Other-relative']
============================================================================================
============================================race============================================================
[' White' ' Black' ' Asian-Pac-Islander' ' Amer-Indian-Eskimo' ' Other']
============================================================================================
============================================sex============================================================
[' Male' ' Female']
============================================================================================
============================================capital_gain===========================================================
=
[ 2174     0 14084  5178  5013  2407 14344 15024  7688 34095  4064  4386
  7298  1409  3674  1055  3464  2050  2176   594 20051  6849  4101  1111
  8614  3411  2597 25236  4650  9386  2463  3103 10605  2964  3325  2580
  3471  4865 99999  6514  1471  2329  2105  2885 25124 10520  2202  2961
 27828  6767  2228  1506 13550  2635  5556  4787  3781  3137  3818  3942
   914   401  2829  2977  4934  2062  2354  5455 15020  1424  3273 22040
  4416  3908 10566   991  4931  1086  7430  6497   114  7896  2346  3418
  3432  2907  1151  2414  2290 15831 41310  4508  2538  3456  6418  1848
  3887  5721  9562  1455  2036  1831 11678  2936  2993  7443  6360  1797
  1173  4687  6723  2009  6097  2653  1639 18481  7978  2387  5060]
============================================================================================
============================================capital_loss===========================================================
=
[   0 2042 1408 1902 1573 1887 1719 1762 1564 2179 1816 1980 1977 1876
 1340 2206 1741 1485 2339 2415 1380 1721 2051 2377 1669 2352 1672  653
 2392 1504 2001 1590 1651 1628 1848 1740 2002 1579 2258 1602  419 2547
 2174 2205 1726 2444 1138 2238  625  213 1539  880 1668 1092 1594 3004
 2231 1844  810 2824 2559 2057 1974  974 2149 1825 1735 1258 2129 2603
 2282  323 4356 2246 1617 1648 2489 3770 1755 3683 2267 2080 2457  155
 3900 2201 1944 2467 2163 2754 2472 1411]
============================================================================================
============================================hour_per_week==========================================================
==
```

```
=
[' <=50K' ' >50K']
=========================================================================================================
```

In [19]:

```
# Seperate both the dataset columns into numeric and categorical columns
numeric_cols = [feature for feature in income.columns if income[feature].dtype != 'O']
categorical_cols =[feature for feature in income.columns if income[feature].dtype == 'O']
```

In [20]:

```
print('Categorical Features')
print(categorical_cols)
print('==============================================================================================================')
print('Numeric Features')
print(numeric_cols)
```

```
Categorical Features
['workclass', 'education', 'marital-status', 'occupation', 'relationship', 'race', 'sex', 'native_country', 'income
_class']
====================================================================================================================
=
Numeric Features
['age', 'fnlwgt', 'education-num', 'capital_gain', 'capital_loss', 'hour_per_week']
```

In [24]:

```
# Removing trailing spaces from all the categorical columns
for col in categorical_cols:
    income[col]=income[col].str.strip()
    print(income[col].unique())
```

```
['State-gov' 'Self-emp-not-inc' 'Private' 'Federal-gov' 'Local-gov' '?'
 'Self-emp-inc' 'Without-pay' 'Never-worked']
['Bachelors' 'HS-grad' '11th' 'Masters' '9th' 'Some-college' 'Assoc-acdm'
 'Assoc-voc' '7th-8th' 'Doctorate' 'Prof-school' '5th-6th' '10th'
 '1st-4th' 'Preschool' '12th']
['Never-married' 'Married-civ-spouse' 'Divorced' 'Married-spouse-absent'
 'Separated' 'Married-AF-spouse' 'Widowed']
['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']
['Not-in-family' 'Husband' 'Wife' 'Own-child' 'Unmarried' 'Other-relative']
['White' 'Black' 'Asian-Pac-Islander' 'Amer-Indian-Eskimo' 'Other']
['Male' 'Female']
['United-States' 'Cuba' 'Jamaica' 'India' '?' 'Mexico' 'South'
 'Puerto-Rico' 'Honduras' 'England' 'Canada' 'Germany' 'Iran'
 'Philippines' 'Italy' 'Poland' 'Columbia' 'Cambodia' 'Thailand' 'Ecuador'
 'Laos' 'Taiwan' 'Haiti' 'Portugal' 'Dominican-Republic' 'El-Salvador'
 'France' 'Guatemala' 'China' 'Japan' 'Yugoslavia' 'Peru'
 'Outlying-US(Guam-USVI-etc)' 'Scotland' 'Trinadad&Tobago' 'Greece'
 'Nicaragua' 'Vietnam' 'Hong' 'Ireland' 'Hungary' 'Holand-Netherlands']
['<=50K' '>50K']
```

In [33]:

```
# Cleaning up the some categorical columns by replacing '?' with nan
income['workclass'] = income['workclass'].str.replace('?', str(np.nan))
income['occupation'] = income['occupation'].str.replace('?', str(np.nan))
income['native_country'] = income['native_country'].str.replace('?', str(np.nan))
income['education'].replace({'11th':str(np.nan), '9th':str(np.nan), '7th-8th':str(np.nan), '1st-4th':str(np.nan),'12th':str(np.na
```

In [34]:

```
income['workclass'].unique()
```

In [28]:

```
income['education'].unique()
```

Out[28]:

```
array(['Bachelors', 'HS-grad', 'NaN', 'Masters', 'Some-college',
       'Assoc-acdm', 'Assoc-voc', 'Doctorate', 'Prof-school', '5th-6th',
       '10th', 'Preschool'], dtype=object)
```

In [28]:

```
# Statistical analysis about the dataset
income.describe()
```

Out[28]:

	age	fnlwgt	education-num	capital_gain	capital_loss	hour_per_week
count	32537.000000	3.253700e+04	32537.000000	32537.000000	32537.000000	32537.000000
mean	38.585549	1.897808e+05	10.081815	1078.443741	87.368227	40.440329
std	13.637984	1.055565e+05	2.571633	7387.957424	403.101833	12.346889
min	17.000000	1.228500e+04	1.000000	0.000000	0.000000	1.000000
25%	28.000000	1.178270e+05	9.000000	0.000000	0.000000	40.000000
50%	37.000000	1.783560e+05	10.000000	0.000000	0.000000	40.000000
75%	48.000000	2.369930e+05	12.000000	0.000000	0.000000	45.000000
max	90.000000	1.484705e+06	16.000000	99999.000000	4356.000000	99.000000

In [29]:

```
income.describe(include='all').T
```

Out[29]:

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
age	32537.0	NaN	NaN	NaN	38.585549	13.637984	17.0	28.0	37.0	48.0	90.0
workclass	32537	9	Private	22673	NaN	NaN	NaN	NaN	NaN	NaN	NaN
fnlwgt	32537.0	NaN	NaN	NaN	189780.848511	105556.471009	12285.0	117827.0	178356.0	236993.0	1484705.0
education	32537	16	HS-grad	10494	NaN	NaN	NaN	NaN	NaN	NaN	NaN
education-num	32537.0	NaN	NaN	NaN	10.081815	2.571633	1.0	9.0	10.0	12.0	16.0
marital-status	32537	7	Married-civ-spouse	14970	NaN	NaN	NaN	NaN	NaN	NaN	NaN
occupation	32537	15	Prof-specialty	4136	NaN	NaN	NaN	NaN	NaN	NaN	NaN
relationship	32537	6	Husband	13187	NaN	NaN	NaN	NaN	NaN	NaN	NaN
race	32537	5	White	27795	NaN	NaN	NaN	NaN	NaN	NaN	NaN
sex	32537	2	Male	21775	NaN	NaN	NaN	NaN	NaN	NaN	NaN
capital_gain	32537.0	NaN	NaN	NaN	1078.443741	7387.957424	0.0	0.0	0.0	0.0	99999.0
capital_loss	32537.0	NaN	NaN	NaN	87.368227	403.101833	0.0	0.0	0.0	0.0	4356.0
hour_per_week	32537.0	NaN	NaN	NaN	40.440329	12.346889	1.0	40.0	40.0	45.0	99.0
native_country	32537	42	United-States	29153	NaN	NaN	NaN	NaN	NaN	NaN	NaN
income_class	32537	2	<=50K	24698	NaN	NaN	NaN	NaN	NaN	NaN	NaN

In [30]:

```
# Checking the correlation among the numeric columns
income[numeric_cols]
```

Out[30]:

	age	fnlwgt	education-num	capital_gain	capital_loss	hour_per_week
0	39	77516	13	2174	0	40
1	50	83311	13	0	0	13
2	38	215646	9	0	0	40
3	53	234721	7	0	0	40
4	28	338409	13	0	0	40
...	...	...	...	...	...	...
32556	27	257302	12	0	0	38
32557	40	154374	9	0	0	40
32558	58	151910	9	0	0	40
32559	22	201490	9	0	0	20
32560	52	287927	9	15024	0	40

32537 rows × 6 columns

In [31]:

```
# Correlation among the numeric columns
income[numeric_cols].corr()
```

Out[31]:

	age	fnlwgt	education-num	capital_gain	capital_loss	hour_per_week
age	1.000000	-0.076447	0.036224	0.077676	0.057745	0.068515
fnlwgt	-0.076447	1.000000	-0.043388	0.000429	-0.010260	-0.018898
education-num	0.036224	-0.043388	1.000000	0.122664	0.079892	0.148422
capital_gain	0.077676	0.000429	0.122664	1.000000	-0.031639	0.078408
capital_loss	0.057745	-0.010260	0.079892	-0.031639	1.000000	0.054229
hour_per_week	0.068515	-0.018898	0.148422	0.078408	0.054229	1.000000

In [36]:

```
# Checking missing values again after columns cleaning
income.isnull().sum()
#income['workclass'].isnull().sum()
#income[income['workclass'] == str(np.nan)]
```

Out[36]:

```
age               0
workclass         0
fnlwgt            0
education         0
education-num     0
marital-status    0
occupation        0
relationship      0
race              0
sex               0
capital_gain      0
capital_loss      0
```

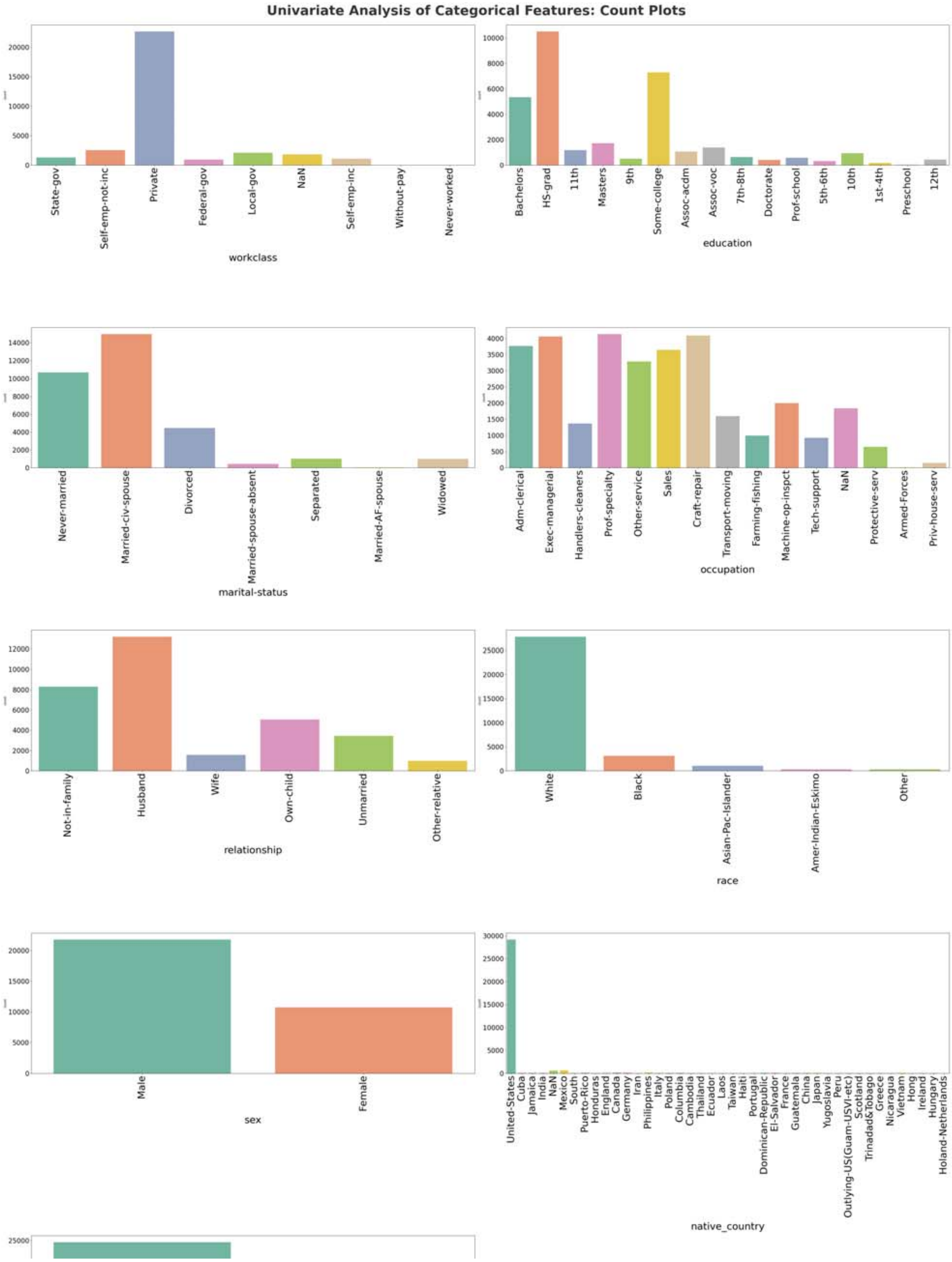
In [33]:

```
for col in categorical_cols:
    print(f"{col}:{income[col].value_counts(normalize=True)*100}")
    print("=============================================")
```

```
workclass: Private             69.683745
 Self-emp-not-inc     7.806497
 Local-gov            6.432677
 NaN                  5.642807
 State-gov            3.989304
 Self-emp-inc         3.429941
 Federal-gov          2.950487
 Without-pay          0.043028
 Never-worked         0.021514
Name: workclass, dtype: float64
=============================================
education: HS-grad         32.252513
 Some-college    22.380674
 Bachelors       16.452039
 Masters          5.292436
 Assoc-voc        4.247472
 11th             3.611273
 Assoc-acdm       3.279344
 10th             2.867505
```

In [34]:

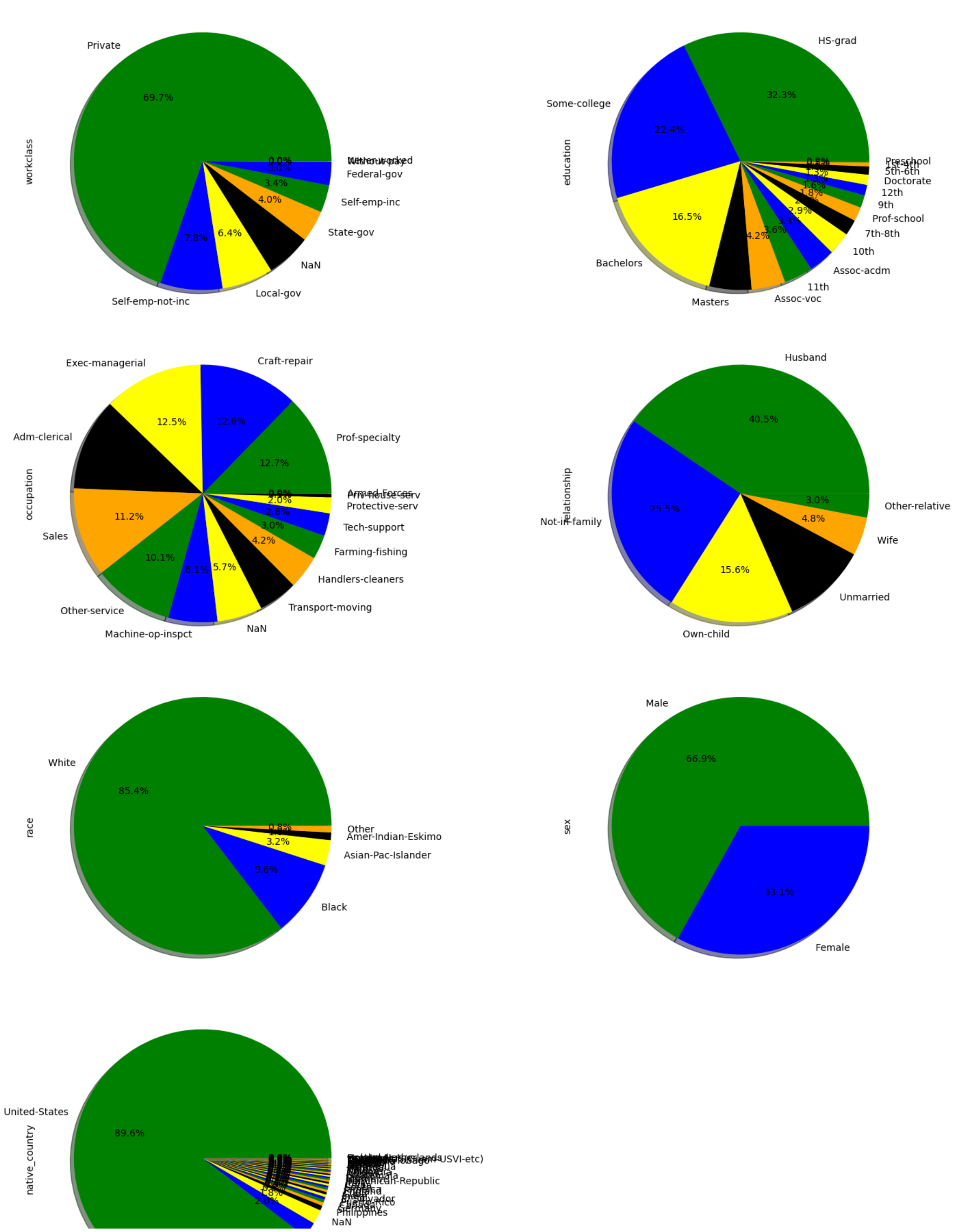
```
plt.figure(figsize=(40, 60))
plt.suptitle('Univariate Analysis of Categorical Features: Count Plots', fontsize=40, fontweight='bold', alpha=0.8, y=1.)
for i in range(0, len(categorical_cols)):
    plt.subplot(5, 2, i+1)
    sns.countplot(x=income[categorical_cols[i]],palette="Set2")
    plt.xlabel(categorical_cols[i], fontsize = 30)
    plt.xticks(rotation=90, fontsize = 30)
    plt.yticks(fontsize = 20)
    plt.tight_layout()
```


Univariate Analysis of Categorical Features: Count Plots
State-gov
Self-emp-not-inc
Private
Federal-gov
Local-gov
NaN
Self-emp-inc
Without-pay
Never-worked
workclass
Bachelors
HS-grad
11th
Masters
9th
Some-college
Assoc-acdm
Assoc-voc
7th-8th
Doctorate
Prof-school
5th-6th
10th
1st-4th
Preschool
12th
education
Never-married
Married-civ-spouse
Divorced
Married-spouse-absent
Separated
Married-AF-spouse
Widowed
marital-status
Adm-clerical
Exec-managerial
Handlers-cleaners
Prof-specialty
Other-service
Sales
Craft-repair
Transport-moving
Farming-fishing
Machine-op-inspct
Tech-support
NaN
Protective-serv
Armed-Forces
Priv-house-serv
occupation
Not-in-family
Husband
Wife
Own-child
Unmarried
Other-relative
relationship
White
Black
Asian-Pac-Islander
Amer-Indian-Eskimo
Other
race
Male
Female
sex
United-States
Cuba
Jamaica
India
NaN
Mexico
South
Puerto-Rico
Honduras
England
Canada
Germany
Iran
Philippines
Italy
Poland
Columbia
Cambodia
Thailand
Ecuador
Laos
Taiwan
Haiti
Portugal
Dominican-Republic
El-Salvador
France
Guatemala
China
Japan
Yugoslavia
Peru
Outlying-US(Guam-USVI-etc)
Scotland
Trinadad&Tobago
Greece
Nicaragua
Vietnam
Hong
Ireland
Hungary
Holand-Netherlands
native_country

In [64]:

```
plt.figure(figsize=(15, 20))
plt.suptitle('Univariate Analysis of Categorical Features: ', fontsize=20, fontweight='bold', alpha=0.8, y=1.)
plt.subplot(421)
income['workclass'].value_counts().plot.pie(y=income['workclass'],autopct='%1.1f%%',shadow=True,colors=['green','blue','yellow','
plt.subplot(422)
income['education'].value_counts().plot.pie(y=income['education'],autopct='%1.1f%%',shadow=True,colors=['green','blue','yellow','
plt.subplot(423)
income['occupation'].value_counts().plot.pie(y=income['occupation'],autopct='%1.1f%%',shadow=True,colors=['green','blue','yellow'
plt.subplot(424)
income['relationship'].value_counts().plot.pie(y=income['relationship'],autopct='%1.1f%%',shadow=True,colors=['green','blue','yel
plt.subplot(425)
income['race'].value_counts().plot.pie(y=income['race'],autopct='%1.1f%%',shadow=True,colors=['green','blue','yellow','black', 'o
plt.subplot(426)
income['sex'].value_counts().plot.pie(y=income['sex'],autopct='%1.1f%%',shadow=True,colors=['green','blue','yellow','black', 'ora
plt.subplot(427)
income['native_country'].value_counts().plot.pie(y=income['native_country'],autopct='%1.1f%%',shadow=True,colors=['green','blue',
plt.tight_layout()
plt.show()
```

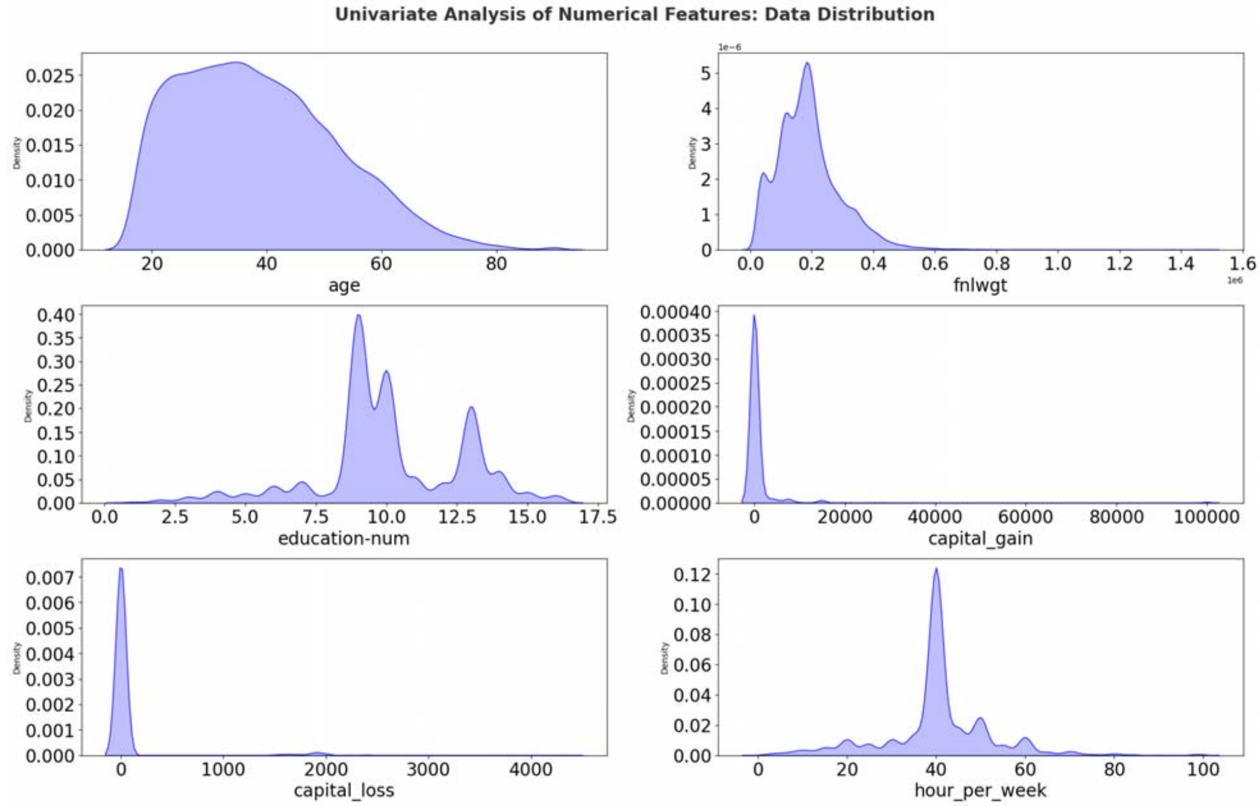
Univariate Analysis of Categorical Features:



In [36]:

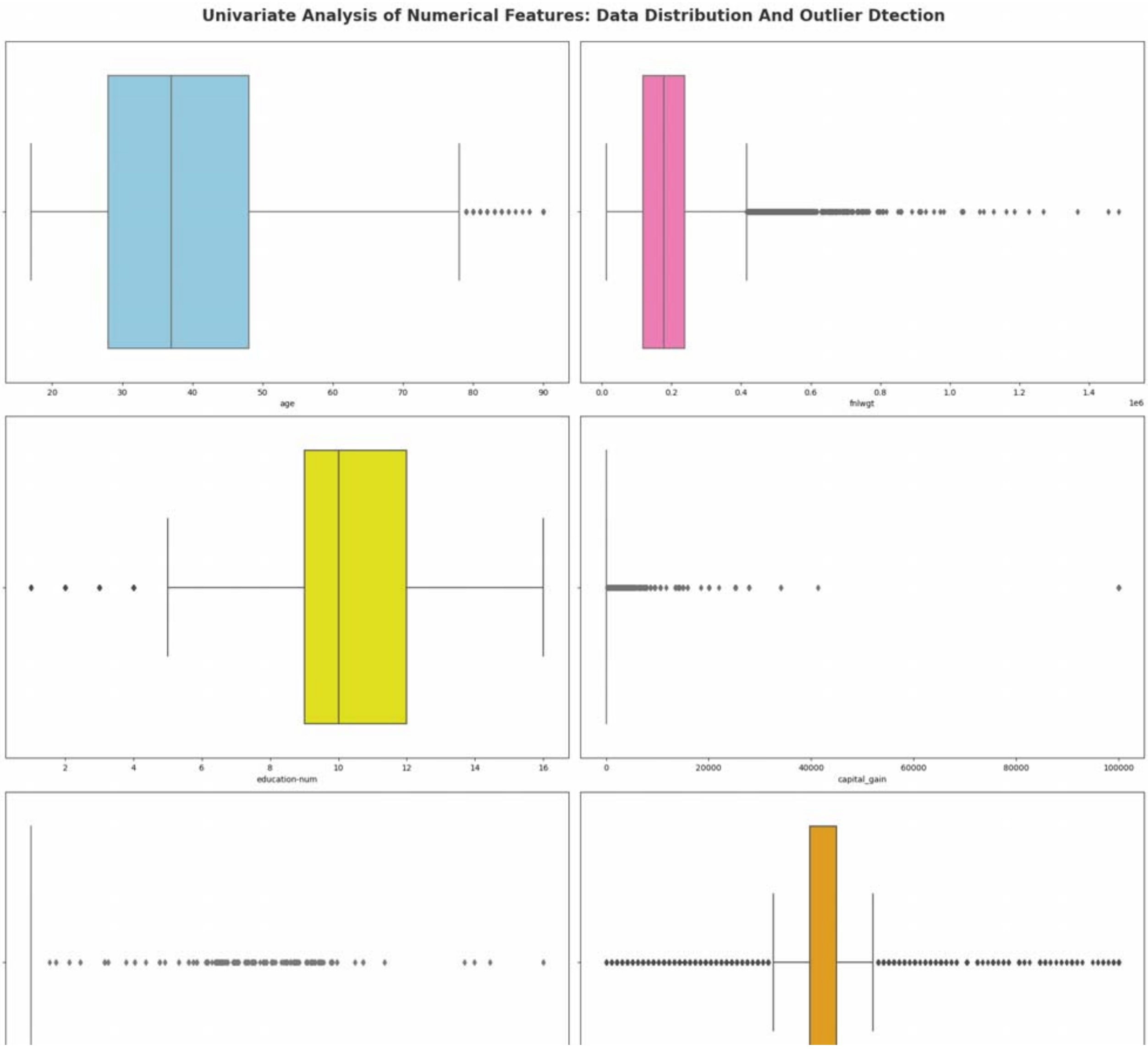
```
plt.figure(figsize=(20, 20))
plt.suptitle('Univariate Analysis of Numerical Features: Data Distribution', fontsize=20, fontweight='bold', alpha=0.8, y=1.)

for i in range(0, len(numeric_cols)):
    plt.subplot(5, 2, i+1)
    sns.kdeplot(x=income[numeric_cols[i]],shade=True, color='b')
    plt.xlabel(numeric_cols[i], fontsize = 20)
    plt.xticks(fontsize = 20)
    plt.yticks(fontsize = 20)
    plt.tight_layout()
```



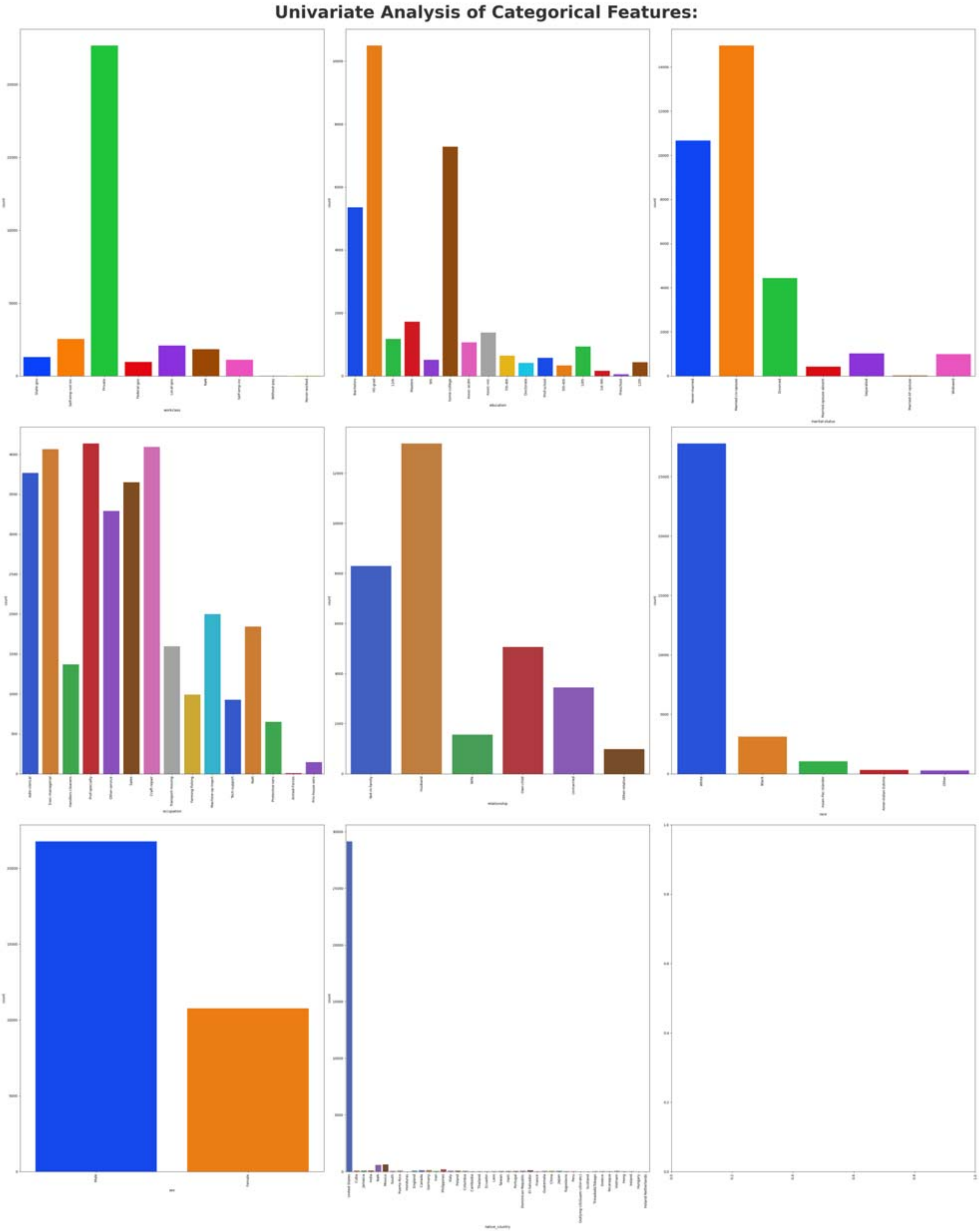
In [37]:

```
plt.subplots(3,2,figsize=(20,20))
plt.suptitle('Univariate Analysis of Numerical Features: Data Distribution And Outlier Dtection', fontsize=20, fontweight='bold',
plt.subplot(321)
sns.boxplot(income['age'],color='skyblue')
plt.subplot(322)
sns.boxplot(income['fnlwgt'],color='hotpink')
plt.subplot(323)
sns.boxplot(income['education-num'],color='yellow')
plt.subplot(324)
sns.boxplot(income['capital_gain'],color='lightgreen')
plt.subplot(325)
sns.boxplot(income['capital_loss'],color='lightblue')
plt.subplot(326)
sns.boxplot(income['hour_per_week'],color='orange')
plt.tight_layout()
plt.show()
```



In [43]:

```
#
plt.subplots(3,3,figsize=(40,50))
plt.suptitle('Univariate Analysis of Categorical Features: ', fontsize=50, fontweight='bold', alpha=0.8, y=1.)
plt.xticks(rotation=90)
plt.subplot(331)
sns.countplot(x=income['workclass'],data=income,palette = 'bright',saturation=0.95)
plt.xticks(rotation=90)
plt.subplot(332)
sns.countplot(x=income['education'],data=income,palette = 'bright',saturation=0.80)
plt.xticks(rotation=90)
plt.subplot(333)
sns.countplot(x=income['marital-status'],data=income,palette = 'bright',saturation=0.90)
plt.xticks(rotation=90)
plt.subplot(334)
sns.countplot(x=income['occupation'],data=income,palette = 'bright',saturation=0.60)
plt.xticks(rotation=90)
plt.subplot(335)
sns.countplot(x=income['relationship'],data=income,palette = 'bright',saturation=0.50)
plt.xticks(rotation=90)
plt.subplot(336)
sns.countplot(x=income['race'],data=income,palette = 'bright',saturation=0.70)
plt.xticks(rotation=90)
plt.subplot(337)
sns.countplot(x=income['sex'],data=income,palette = 'bright',saturation=0.85)
plt.xticks(rotation=90)
plt.subplot(338)
sns.countplot(x=income['native_country'],data=income,palette = 'bright',saturation=0.40)
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```


Univariate Analysis of Categorical Features:

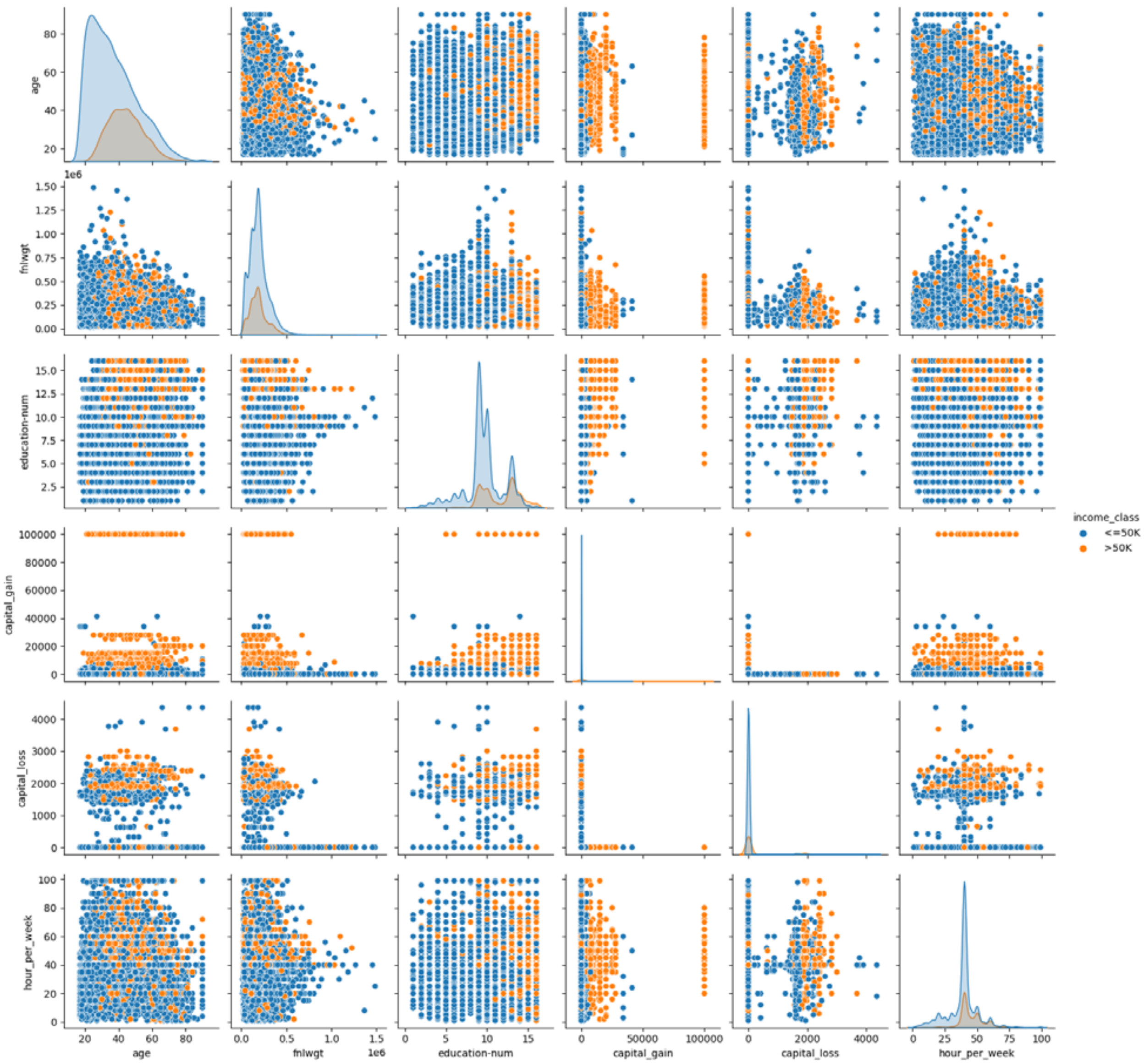
In [40]:

```
# Correlation among numeric features
plt.figure(figsize=(40, 40))
plt.title('MultiVariate Analysis: Feature Correlation', fontsize=20, fontweight='bold', alpha=0.8, y=1.)
sns.pairplot(income, hue='income_class')
plt.tight_layout()
plt.show()
```

Out[40]:

```
<seaborn.axisgrid.PairGrid at 0x2a110a01670>
```

```
<Figure size 4000x4000 with 0 Axes>
```

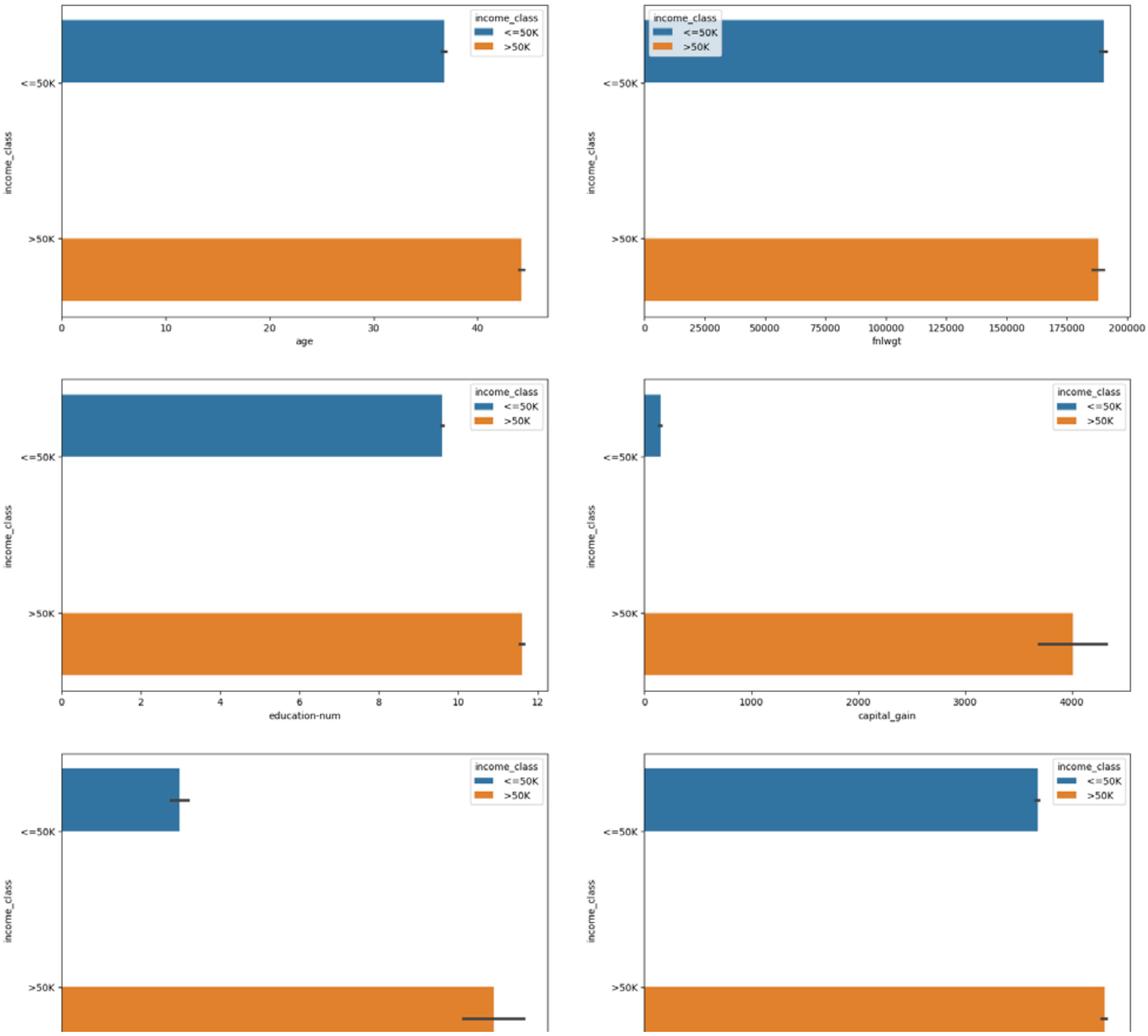


In [49]:

```
# Checkin the relationship between numeric columns and the label feature
plt.figure(figsize=(20,20))
plt.subplot(3,2,1)
sns.barplot (x=income['age'], y=income['income_class'], hue=income['income_class'])
plt.subplot(3,2,2)
sns.barplot (x=income['fnlwgt'], y=income['income_class'], hue=income['income_class'])
plt.subplot(3,2,3)
sns.barplot (x=income['education-num'], y=income['income_class'], hue=income['income_class'])
plt.subplot(3,2,4)
sns.barplot (x=income['capital_gain'], y=income['income_class'], hue=income['income_class'])
plt.subplot(3,2,5)
sns.barplot (x=income['capital_loss'], y=income['income_class'], hue=income['income_class'])
plt.subplot(3,2,6)
sns.barplot (x=income['hour_per_week'], y=income['income_class'], hue=income['income_class'])
```

Out[49]:

```
<AxesSubplot:xlabel='hour_per_week', ylabel='income_class'>
```

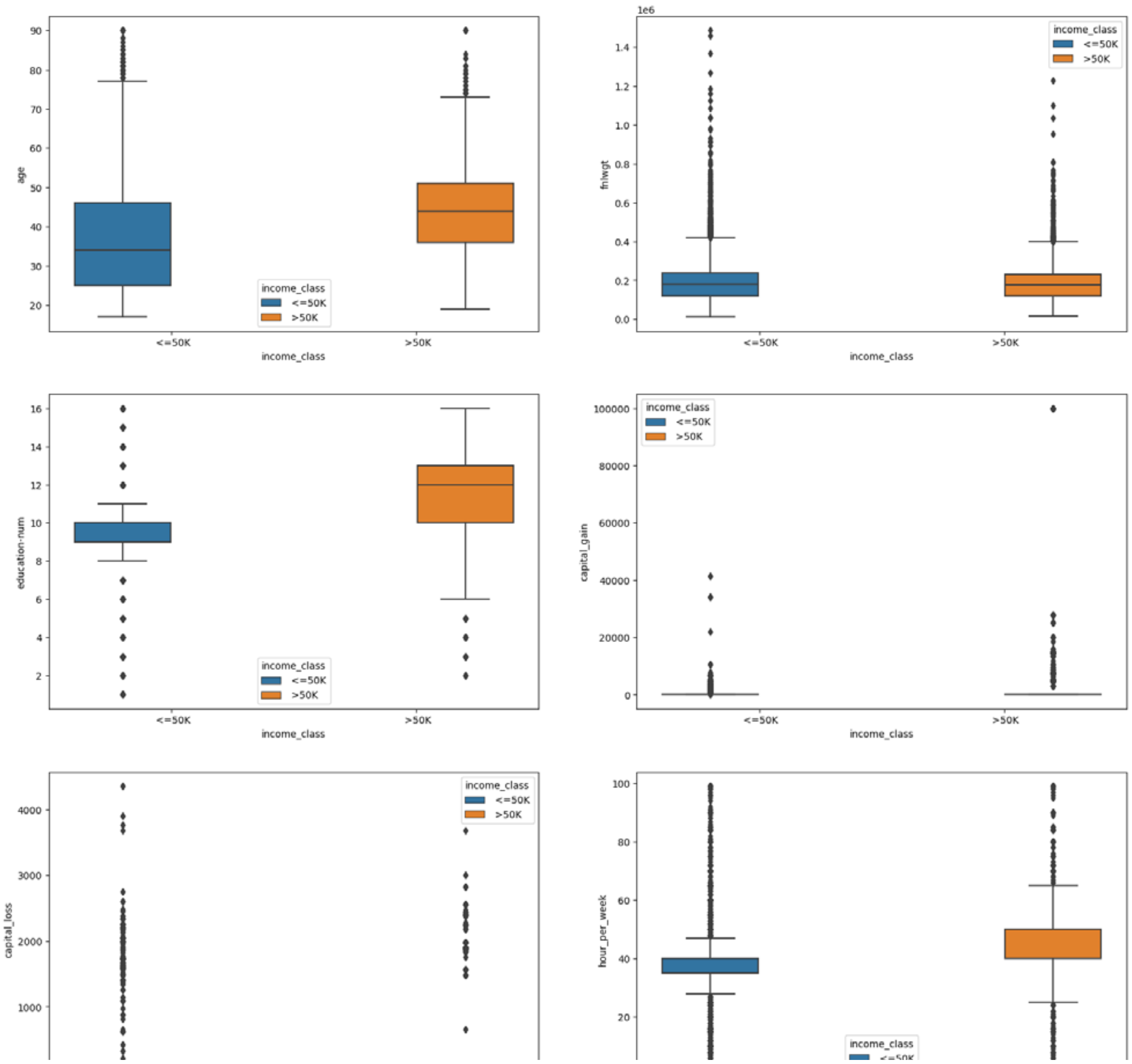


In [48]:

```
# Checkin the relationship between numeric columns and the label feature
plt.figure(figsize=(20,20))
plt.subplot(3,2,1)
sns.boxplot (y=income['age'], x=income['income_class'], hue=income['income_class'])
plt.subplot(3,2,2)
sns.boxplot (y=income['fnlwgt'], x=income['income_class'], hue=income['income_class'])
plt.subplot(3,2,3)
sns.boxplot (y=income['education-num'], x=income['income_class'], hue=income['income_class'])
plt.subplot(3,2,4)
sns.boxplot (y=income['capital_gain'], x=income['income_class'], hue=income['income_class'])
plt.subplot(3,2,5)
sns.boxplot (y=income['capital_loss'], x=income['income_class'], hue=income['income_class'])
plt.subplot(3,2,6)
sns.boxplot (y=income['hour_per_week'], x=income['income_class'], hue=income['income_class'])
```

Out[48]:

```
<AxesSubplot:xlabel='income_class', ylabel='hour_per_week'>
```

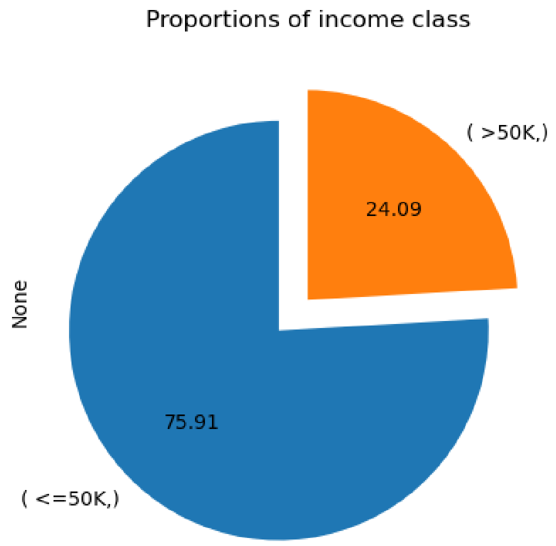


In [451]:

```
# Proportion of people that earn more than 50k
income.value_counts(['income_class']).plot.pie(y='income_class',startangle=90, explode=(0.2,0), title='Proportions of income clas
```

Out[451]:

```
<AxesSubplot:title={'center':'Proportions of income class'}, ylabel='None'>
```



In [460]:

```
# Top 5 workclass earning highest hour per week
income.groupby(['workclass'])['hour_per_week'].sum().sort_values(ascending = False).reset_index()
```

Out[460]:

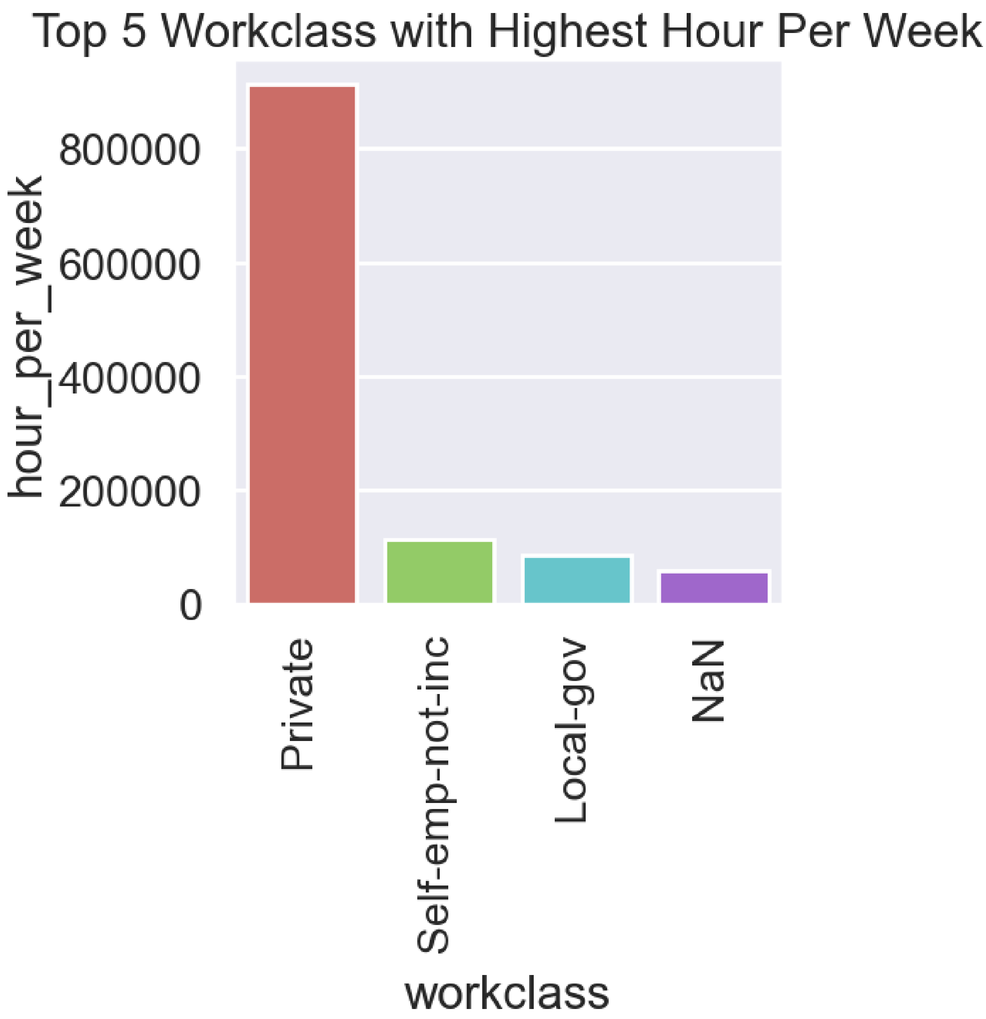
	workclass	hour_per_week
0	Private	913065
1	Self-emp-not-inc	112836
2	Local-gov	85777
3	NaN	58604
4	Self-emp-inc	54481
5	State-gov	50663
6	Federal-gov	39724
7	Without-pay	458
8	Never-worked	199

In [35]:

```
plt.figure(figsize=(5,5))
Top_5 =income.groupby(['workclass'])['hour_per_week'].sum().sort_values(ascending = False).reset_index()[:4]
#sns.set_context("poster")
sns.set_style("darkgrid")
plt.title('Top 5 Workclass with Highest Hour Per Week')
sns.barplot(data = Top_5, y='hour_per_week', x='workclass', palette='hls')
plt.xticks(rotation=90)
```

Out[35]:

```
(array([0, 1, 2, 3]),
 [Text(0, 0, ' Private'),
  Text(1, 0, ' Self-emp-not-inc'),
  Text(2, 0, ' Local-gov'),
  Text(3, 0, ' NaN')])
```



In [19]:

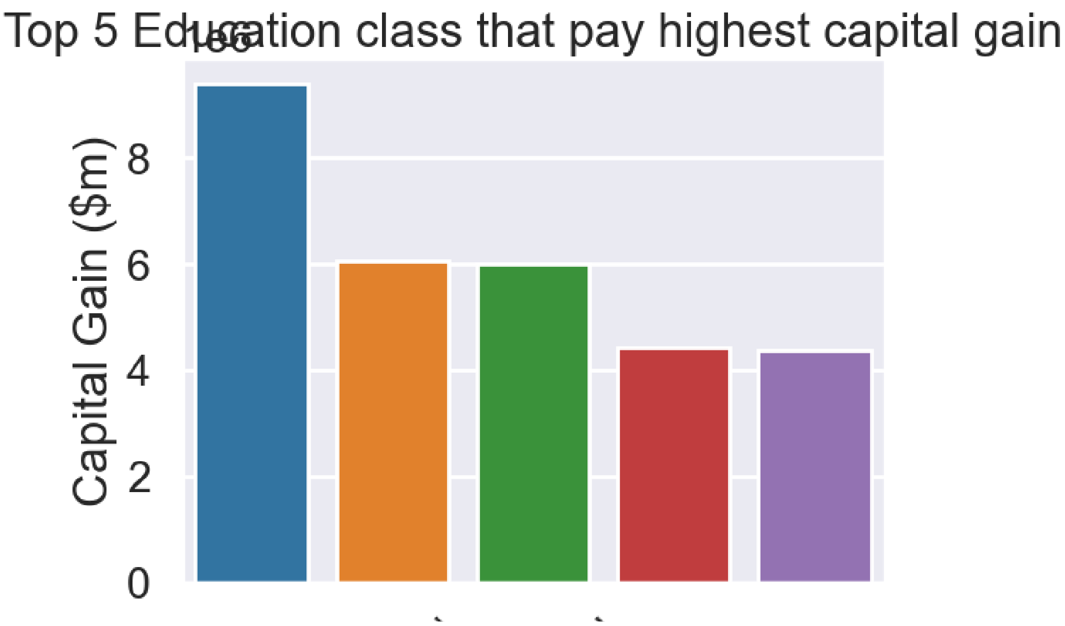
```
# Top 5 Education class that pay highest capital gain
income.groupby(['education'])['capital_gain'].sum().sort_values(ascending=False).reset_index()
```

Out[19]:

	education	capital_gain
0	Bachelors	9404984
1	HS-grad	6056978
2	Prof-school	5998704
3	Masters	4415297
4	Some-college	4366027
5	Doctorate	1970070
6	Assoc-voc	988201
7	Assoc-acdm	683306
8	10th	377468
9	11th	252740
10	9th	175834
11	7th-8th	151125
12	12th	123010
13	5th-6th	58615
14	Preschool	45818
15	1st-4th	21147

In [39]:

```
Top_5_Cgain=income.groupby(['education'])['capital_gain'].sum().sort_values(ascending=False).reset_index()[:5]
plt.title('Top 5 Education class that pay highest capital gain')
sns.barplot(data=Top_5_Cgain, x = 'education', y='capital_gain')
plt.xticks(rotation=45)
plt.ylabel('Capital Gain ($m)')
plt.show()
```



In [ ]:

```

```