```
pip install ucimlrepo
```

```
Collecting ucimlrepo
  Downloading ucimlrepo-0.0.6-py3-none-any.whl (8.0 kB)
Installing collected packages: ucimlrepo
Successfully installed ucimlrepo-0.0.6
```

```
from ucimlrepo import fetch_ucirepo

# fetch dataset
census_income = fetch_ucirepo(id=20)

# data (as pandas dataframes)
X = census_income.data.features
y = census_income.data.targets
```

```
import pandas as pd
import numpy as np
df = pd.concat([X,y], axis=1)
df
```

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-w
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0	
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...
48837	39	Private	215419	Bachelors	13	Divorced	Prof-specialty	Not-in-family	White	Female	0	0	
48838	64	NaN	321403	HS-grad	9	Widowed	NaN	Other-relative	Black	Male	0	0	
48839	38	Private	374983	Bachelors	13	Married-civ-spouse	Prof-specialty	Husband	White	Male	0	0	
48840	44	Private	83891	Bachelors	13	Divorced	Adm-clerical	Own-child	Asian-Pac-Islander	Male	5455	0	
48841	35	Self-emp-inc	182148	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	

48842 rows × 15 columns

Next steps: View recommended plots

```
df = df.replace({'?': None})
```

```
df['workclass'] = df['workclass'].fillna('Never-worked')
```

```
df = df.fillna(value='None')
```

```
df
```

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-w
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	

2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...
48837	39	Private	215419	Bachelors	13	Divorced	Prof-specialty	Not-in-family	White	Female	0	0
48838	64	Never-worked	321403	HS-grad	9	Widowed	None	Other-relative	Black	Male	0	0
48839	38	Private	374983	Bachelors	13	Married-civ-spouse	Prof-specialty	Husband	White	Male	0	0
48840	44	Private	83891	Bachelors	13	Divorced	Adm-clerical	Own-child	Asian-Pac-Islander	Male	5455	0
48841	35	Self-emp-inc	182148	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0

48842 rows × 15 columns

Next steps: View recommended plots

```
1 df.dtypes
```

```
age               int64
workclass        object
fnlwgt            int64
education        object
education-num     int64
marital-status   object
occupation       object
relationship     object
race             object
sex              object
capital-gain      int64
capital-loss      int64
hours-per-week    int64
native-country   object
income           object
dtype: object
```

```
1 df['relationship'].unique()
```

```
array(['Own-child', 'Other-relative', 'Not-in-family', 'Single',
       'Husband', 'Wife'], dtype=object)
```

```
1 df['income'] = df['income'].replace({'>50K.':'Over 50,000',
2                                      '>50K':'Over 50,000',
3                                      '<=50K':'Under 50,000',
4                                      '<=50K.':'Under 50,000'})
```

```
1 df['marital-status'] = df['marital-status'].replace({'Never-Married':'Single',
2                                                      'Married-civ-spouse':'Married',
3                                                      'Married-AF-spouse':'Married',
4                                                      'Married-spouse-absent':'Married'})
```

```
1 df['workclass'] = df['workclass'].replace({'Without-pay':'Interns',
2                                            'Local-gov':'Civil Servant',
3                                            'State-gov':'Civil Servant',
4                                            'Federal-gov':'Civil Servant',
5                                            'Self-emp-inc':'Self Employed',
6                                            'Self-emp-not-inc':'Self Employed',
7                                            'Private':'Office Worker',
8                                            'Never-worked':'Unemployed'})
```

```
1 df['education'] = df['education'].replace({'1st-4th': 'Compulsory',
2                                            '5th-6th': 'Compulsory',
3                                            '7th-8th': 'Compulsory',
4                                            '9th': 'Compulsory',
5                                            '10th': 'Compulsory',
6                                            '11th': 'Compulsory',
7                                            '12th': 'Compulsory',
8                                            'Preschool': 'Compulsory',
```

```
                                    'Bachelors': 'Bachelors',
                                    'Some-college': 'Bachelors',
                                    'Assoc-acdm': 'Associate',
                                    'Assoc-voc': 'Associate',
                                    'Masters': 'Postgraduate',
                                    'Doctorate': 'Postgraduate',
                                    'Prof-school': 'Postgraduate',
                                    'HS-grad': 'HS Graduate'})
```

```
df['relationship'] = df['relationship'].replace({'Unmarried':'Single'})
```

```
df['race'] = df['race'].replace({'Asian-Pac-Islander': 'Asian', 'Amer-Indian-Eskimo': 'Native American'})
```

```
df
```

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hour pe we
28092	17	Office Worker	36877	Compulsory	6	Never-married	Sales	Own-child	White	Female	0	0	
46962	17	Office Worker	73820	Compulsory	8	Never-married	Sales	Own-child	White	Female	0	0	
42922	17	Office Worker	165457	Compulsory	6	Never-married	Other-service	Own-child	White	Male	0	0	
1389	17	Office Worker	46496	Compulsory	7	Never-married	Other-service	Own-child	White	Male	0	0	
32963	17	Office Worker	40299	Compulsory	7	Never-married	Sales	Own-child	White	Female	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	
5104	90	Office Worker	52386	Bachelors	10	Never-married	Other-service	Not-in-family	Asian	Male	0	0	
18725	90	Civil Servant	153602	HS Graduate	9	Married-civ-spouse	Other-service	Husband	White	Male	6767	0	
222	90	Office Worker	51744	HS Graduate	9	Never-married	Other-service	Not-in-family	Black	Male	0	2206	
31696	90	Unemployed	313986	HS Graduate	9	Married-civ-spouse	None	Husband	White	Male	0	0	
6624	90	Office Worker	313986	Compulsory	7	Married-civ-spouse	Craft-repair	Husband	White	Male	0	0	

48842 rows × 15 columns

Next steps: View recommended plots

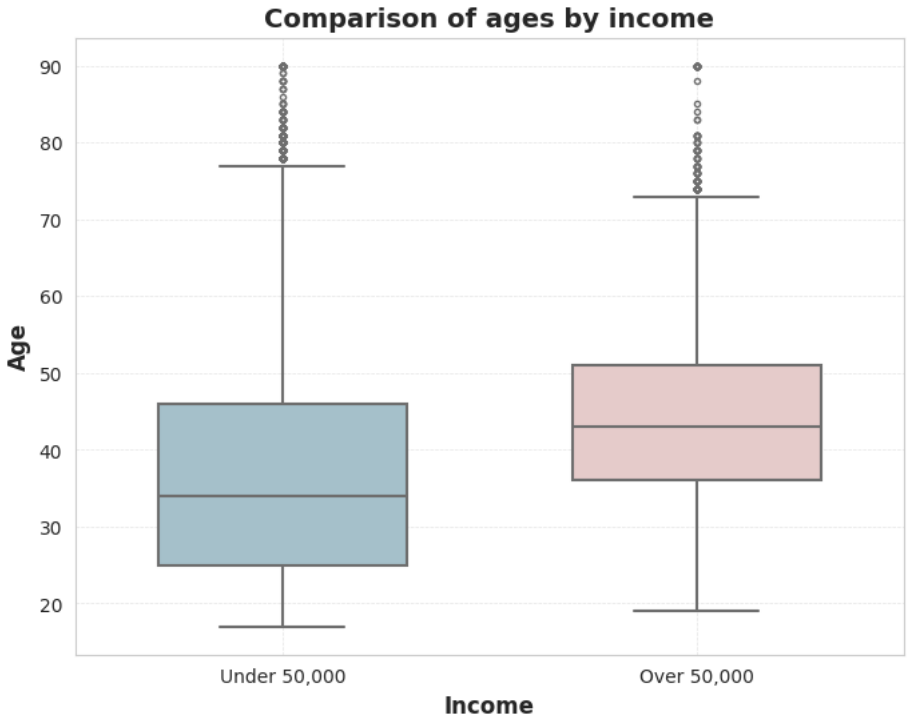
```
maledf = df[df['sex']=='Male']
```

```
femaledf = df[df['sex']=='Female']
```

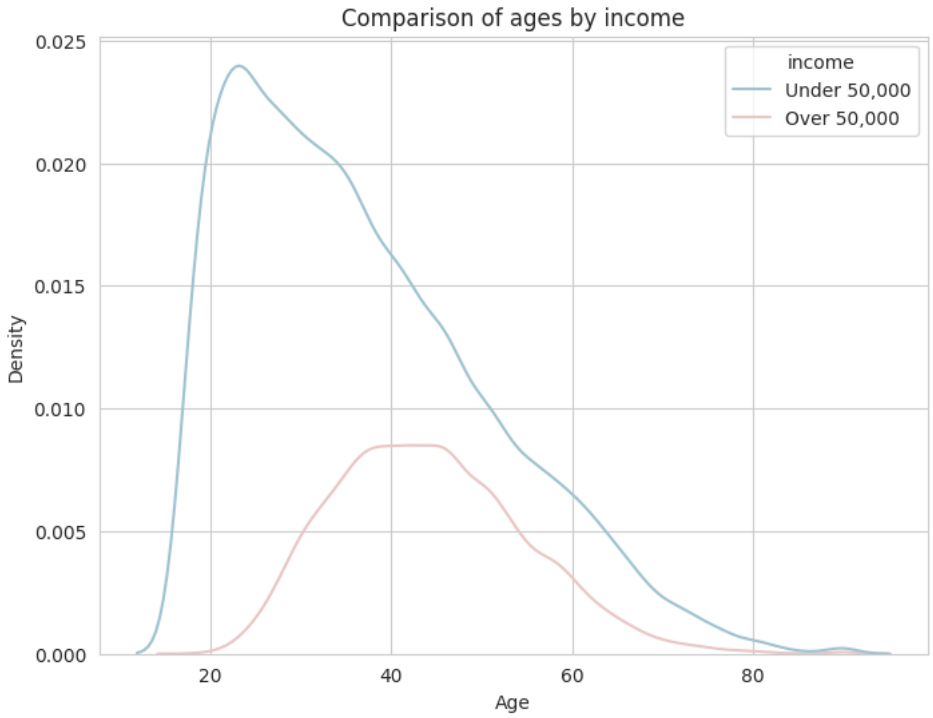
```
over50df = df[df['income']=='Over 50,000']
```

```
import seaborn as sns
import matplotlib.pyplot as plt
sns.boxplot(x='income', y='age', data=df, width=0.6, fliersize=3, linewidth=1.5, palette=['#9FC3D0', '#E9C7C6'])
plt.title('Comparison of ages by income', fontsize=14, fontweight='bold')
plt.xlabel('Income', fontsize=12, fontweight='bold')
plt.ylabel('Age', fontsize=12, fontweight='bold')
plt.rcParams['figure.figsize'] = (8, 6)
plt.grid(True, linestyle='--', linewidth=0.5, alpha=0.5)
plt.tick_params(axis='both', which='major', labelsize=10)
plt.show()
```

```
<ipython-input-261-664bcb675745>:3: FutureWarning:

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

  sns.boxplot(x='income', y='age', data=df, width=0.6, fliersize=3, linewidth=1.5, palette=['#9FC3D0', '#E9C7C6'])
```

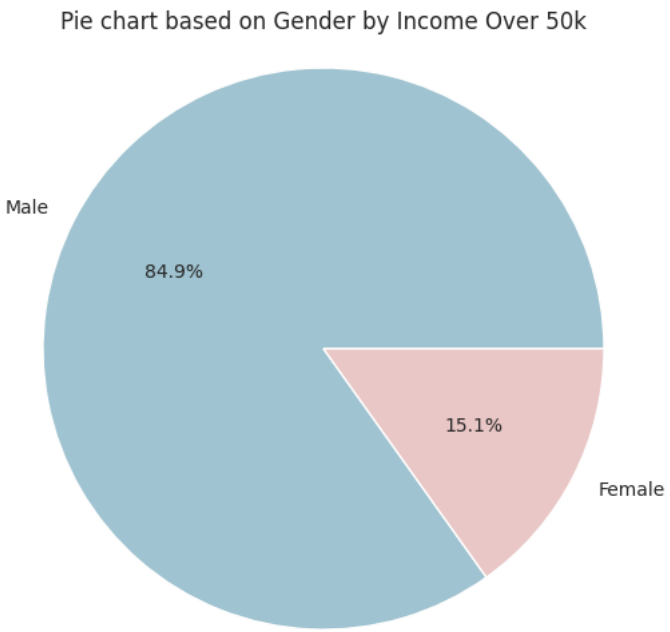



```
sns.kdeplot(data=df, x='age', hue='income', palette=['#9FC3D0', '#E9C7C6'])
plt.title('Comparison of ages by income')
plt.xlabel('Age')
plt.ylabel('Density')
plt.show()
```

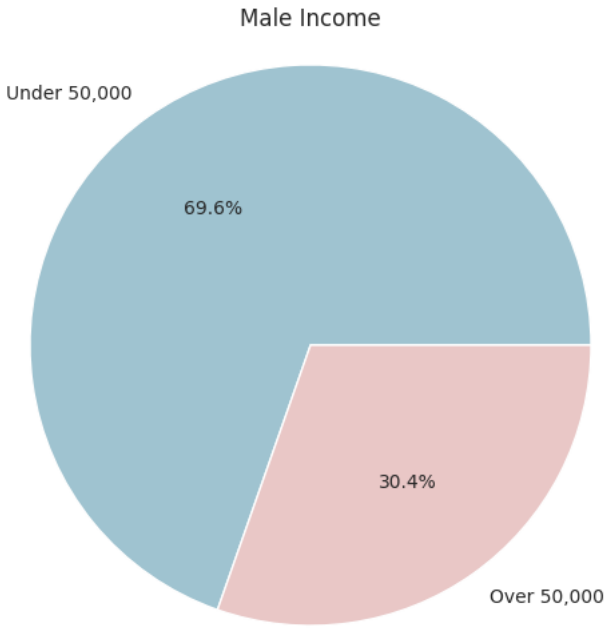



```
# Create a pie chart of the number of cars by cyl
labels = over50df['sex'].value_counts().index
sizes = over50df['sex'].value_counts().values
colors = ['#9FC3D0', '#E9C7C6']

plt.pie(sizes, labels=labels, colors=colors, autopct='%1.1f%%')
plt.axis('equal')
```

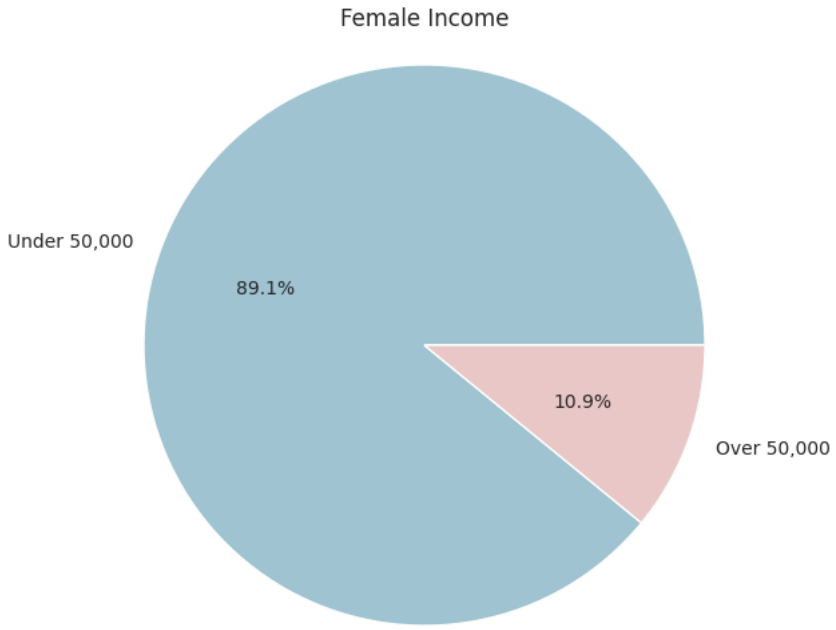
```
plt.title('Pie chart based on Gender by Income Over 50k')
plt.show()
```



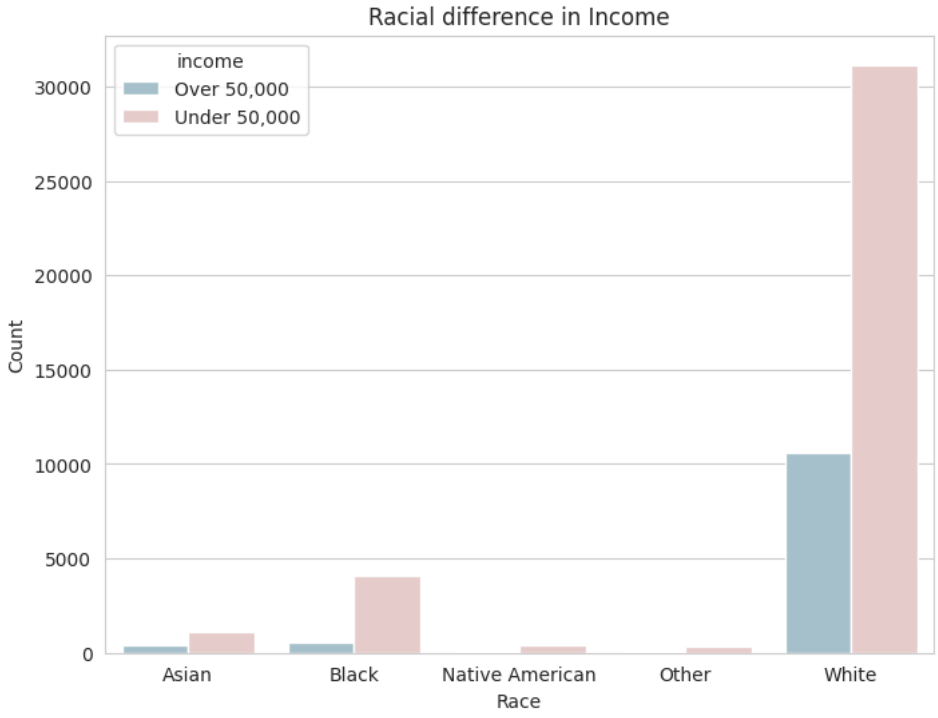

```
# Create a pie chart of the number of cars by cyl
labels = maledf['income'].value_counts().index
sizes = maledf['income'].value_counts().values
colors = ['#9FC3D0', '#E9C7C6']

plt.pie(sizes, labels=labels, colors=colors, autopct='%1.1f%%')
plt.axis('equal')
plt.title('Male Income')
plt.show()
```



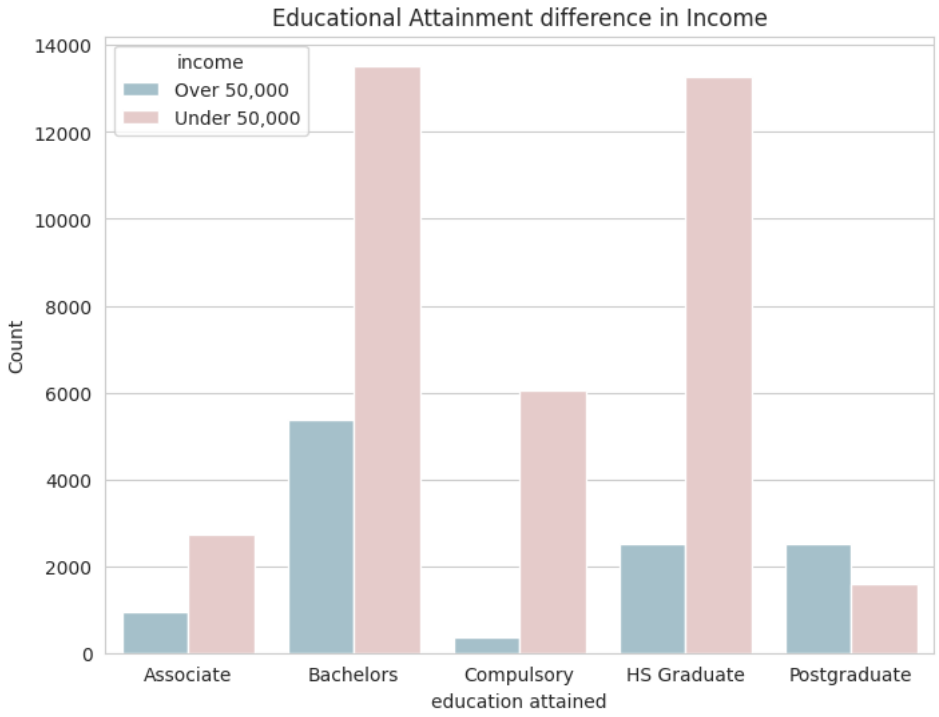

```
# Create a pie chart of the number of cars by cyl
labels = femaledf['income'].value_counts().index
sizes = femaledf['income'].value_counts().values
colors = ['#9FC3D0', '#E9C7C6']

plt.pie(sizes, labels=labels, colors=colors, autopct='%1.1f%%')
plt.axis('equal')
plt.title('Female Income')
plt.show()
```



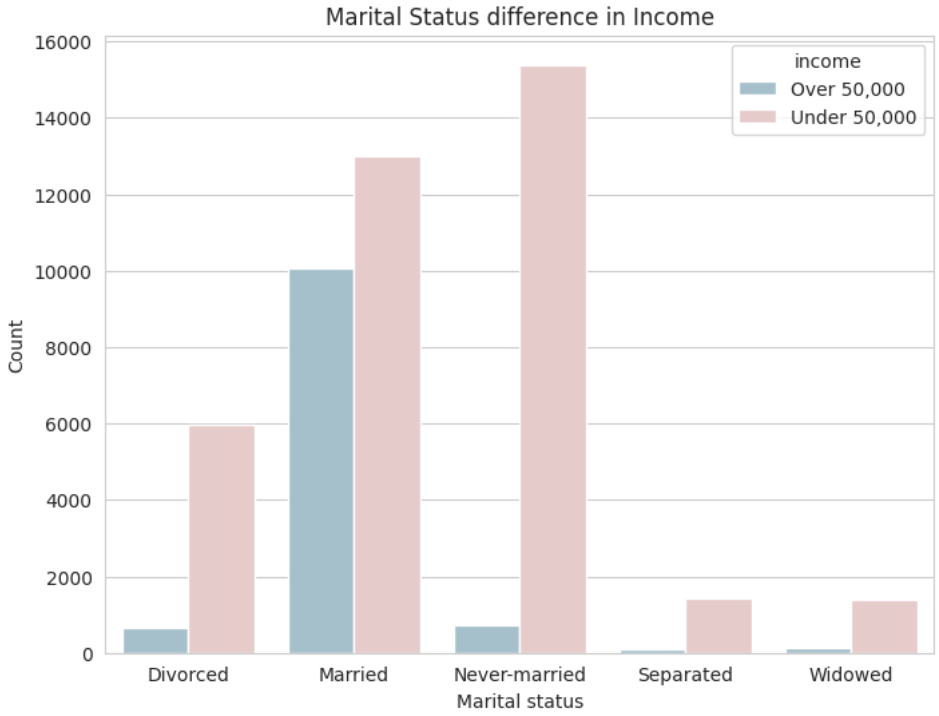

```
grouped_data = df.groupby(['income', 'race']).size().reset_index(name='counts')
sns.barplot(x='race', y='counts', hue='income', data=grouped_data, palette=['#9FC3D0', '#E9C7C6'])
plt.title('Racial difference in Income')
plt.xlabel('Race')
plt.ylabel('Count')
plt.show()
```



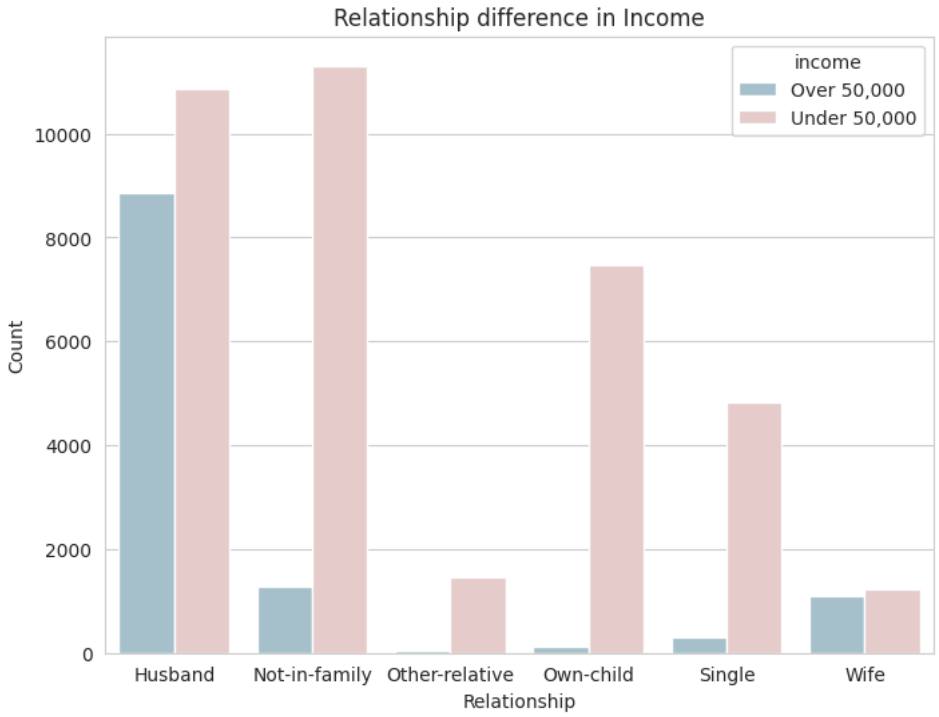

```
grouped_data = df.groupby(['income', 'education']).size().reset_index(name='counts')
sns.barplot(x='education', y='counts', hue='income', data=grouped_data, palette=['#9FC3D0', '#E9C7C6'])
plt.title('Educational Attainment difference in Income')
plt.xlabel('education attained')
plt.ylabel('Count')
plt.show()
```




```
grouped_data = df.groupby(['income', 'marital-status']).size().reset_index(name='counts')
sns.barplot(x='marital-status', y='counts', hue='income', data=grouped_data, palette=['#9FC3D0', '#E9C7C6'])
plt.title('Marital Status difference in Income')
plt.xlabel('Marital status')
plt.ylabel('Count')
plt.show()
```




```
grouped_data = df.groupby(['income', 'relationship']).size().reset_index(name='counts')
sns.barplot(x='relationship', y='counts', hue='income', data=grouped_data, palette=['#9FC3D0', '#E9C7C6'])
plt.title('Relationship difference in Income')
plt.xlabel('Relationship')
plt.ylabel('Count')
plt.show()
```


Relationship difference in Income
income
Over 50,000
Under 50,000
Count
0
2000
4000
6000
8000
10000
Husband
Not-in-family
Other-relative
Own-child
Single
Wife
Relationship