Adult Cencus Data Analysis

June 16, 2022

1 Import Basic Libraries

[34]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

[35]:
```
#Load the dataset
df=pd.read_csv('adult.csv')
```

[36]:
```
df.head()
```

[36]:
```
   age          workclass  fnlwgt   education  education-num  \
0   39          State-gov   77516   Bachelors             13
1   50   Self-emp-not-inc   83311   Bachelors             13
2   38            Private  215646     HS-grad              9
3   53            Private  234721        11th              7
4   28            Private  338409   Bachelors             13

        marital-status         occupation    relationship   race      sex  \
0        Never-married       Adm-clerical   Not-in-family  White     Male
1   Married-civ-spouse    Exec-managerial         Husband  White     Male
2             Divorced  Handlers-cleaners   Not-in-family  White     Male
3   Married-civ-spouse  Handlers-cleaners         Husband  Black     Male
4   Married-civ-spouse     Prof-specialty            Wife  Black   Female

   capital-gain  capital-loss  hours-per-week         country  salary
0          2174             0              40   United-States   <=50K
1             0             0              13   United-States   <=50K
2             0             0              40   United-States   <=50K
3             0             0              40   United-States   <=50K
4             0             0              40            Cuba   <=50K
```

2 Data cleaning

[37]:
```
#check for null values
df.isnull().sum().sum()
```

[37]:
```
0
```

[38]:
```
df.columns
```

[38]:
```
Index(['age', 'workclass', 'fnlwgt', 'education', 'education-num',
       'marital-status', 'occupation', 'relationship', 'race', 'sex',
       'capital-gain', 'capital-loss', 'hours-per-week', 'country', 'salary'],
      dtype='object')
```

[39]:
```
df['salary'].unique()
```

[39]:
```
array([' <=50K', ' >50K'], dtype=object)
```

[40]:
```
df.groupby('salary').mean()
```

[40]:
```
              age        fnlwgt  education-num  capital-gain  capital-loss  \
salary
 <=50K  36.783738  190340.86517       9.595065    148.752468     53.142921
 >50K   44.249841  188005.00000      11.611657   4006.142456    195.001530

        hours-per-week
salary
 <=50K       38.840210
 >50K        45.473026
```

[41]:
```
df.describe().T
```

[41]:
```
                  count           mean            std      min       25%  \
age             32561.0      38.581647      13.640433     17.0      28.0
fnlwgt          32561.0  189778.366512  105549.977697  12285.0  117827.0
education-num   32561.0      10.080679       2.572720      1.0       9.0
capital-gain    32561.0    1077.648844    7385.292085      0.0       0.0
capital-loss    32561.0      87.303830     402.960219      0.0       0.0
hours-per-week  32561.0      40.437456      12.347429      1.0      40.0

                     50%       75%        max
age                 37.0      48.0       90.0
fnlwgt          178356.0  237051.0  1484705.0
education-num       10.0      12.0       16.0
capital-gain         0.0       0.0    99999.0
capital-loss         0.0       0.0     4356.0
hours-per-week      40.0      45.0       99.0
```

```
[42]: df['workclass'].value_counts()
```

```
[42]:  Private              22696
       Self-emp-not-inc      2541
       Local-gov             2093
       ?                     1836
       State-gov             1298
       Self-emp-inc          1116
       Federal-gov            960
       Without-pay             14
       Never-worked             7
      Name: workclass, dtype: int64
```

Maximum people are working in Private sector

```
[43]: df['education'].value_counts()
```

```
[43]:  HS-grad         10501
       Some-college     7291
       Bachelors        5355
       Masters          1723
       Assoc-voc        1382
       11th             1175
       Assoc-acdm       1067
       10th              933
       7th-8th           646
       Prof-school       576
       9th               514
       12th              433
       Doctorate         413
       5th-6th           333
       1st-4th           168
       Preschool          51
      Name: education, dtype: int64
```

Maximum people has done their High School

```
[44]: df['marital-status'].value_counts()
```

```
[44]:  Married-civ-spouse       14976
       Never-married            10683
       Divorced                  4443
       Separated                 1025
       Widowed                    993
       Married-spouse-absent      418
       Married-AF-spouse           23
      Name: marital-status, dtype: int64
```

Maximum people are married with civilian spouse

[45]:
```
df['relationship'].value_counts()
```

[45]:
```
 Husband           13193
 Not-in-family      8305
 Own-child          5068
 Unmarried          3446
 Wife               1568
 Other-relative      981
Name: relationship, dtype: int64
```

Maximum people are Husband who are working

[46]:
```
df['sex'].value_counts()
```

[46]:
```
 Male      21790
 Female    10771
Name: sex, dtype: int64
```

No. of males is twice than females

[47]:
```
df['occupation'].value_counts()
```

[47]:
```
 Prof-specialty       4140
 Craft-repair         4099
 Exec-managerial      4066
 Adm-clerical         3770
 Sales                3650
 Other-service        3295
 Machine-op-inspct    2002
 ?                    1843
 Transport-moving     1597
 Handlers-cleaners    1370
 Farming-fishing       994
 Tech-support          928
 Protective-serv       649
 Priv-house-serv       149
 Armed-Forces            9
Name: occupation, dtype: int64
```

Maximum people has occupation as prof-speciality(professor in a perticular subject)

There are very less people who are in Armed forces

[48]:
```
df['country'].value_counts()
```

[48]:
```
 United-States                 29170
 Mexico                          643
 ?                               583
 Philippines                     198
 Germany                         137
 Canada                          121
 Puerto-Rico                     114
 El-Salvador                     106
 India                           100
 Cuba                             95
 England                          90
 Jamaica                          81
 South                            80
 China                            75
 Italy                            73
 Dominican-Republic               70
 Vietnam                          67
 Guatemala                        64
 Japan                            62
 Poland                           60
 Columbia                         59
 Taiwan                           51
 Haiti                            44
 Iran                             43
 Portugal                         37
 Nicaragua                        34
 Peru                             31
 Greece                           29
 France                           29
 Ecuador                          28
 Ireland                          24
 Hong                             20
 Cambodia                         19
 Trinadad&Tobago                  19
 Thailand                         18
 Laos                             18
 Yugoslavia                       16
 Outlying-US(Guam-USVI-etc)       14
 Hungary                          13
 Honduras                         13
 Scotland                         12
 Holand-Netherlands                1
Name: country, dtype: int64
```

No. of employment in united satate is maximum than other countries

[49]:

```
#Filling ?
df['workclass']=df['workclass'].replace(' ?','private')
df['country']=df['country'].replace(' ?','United-States')
df['occupation']=df['occupation'].replace(' ?','prof-spaciality')
```

3 Feature engineering

[50]:

```
df.education=df.education.replace([' Preschool',' 1st-4th',' 5th-6th','␣
 ↪7th-8th',' 9th',' 10th',' 11th',' 12th',],'School')
df.education=df.education.replace(' HS-grad','High-School')
df.education=df.education.replace([' Assoc-acdm',' Assoc-voc',' Some-college','␣
 ↪Prof-school'],'Higher')
df.education=df.education.replace(' Bachelors','Graduates')
df.education=df.education.replace(' Doctorate','Doc')
```

[51]:

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

[51]:

```
array(['Graduates', 'High-School', 'School', ' Masters', 'Higher', 'Doc'],
      dtype=object)
```

[52]:

```
df['marital-status']=df['marital-status'].replace([' Married-spouse-absent','␣
 ↪Married-civ-spouse',' Married-AF-spouse'],'married')
df['marital-status']=df['marital-status'].replace([' Divorced',' Separated','␣
 ↪Widowed'],'others')
```

[53]:

```
df['marital-status'].unique()
```

[53]:

```
array([' Never-married', 'married', 'others'], dtype=object)
```

[54]:

```
#what is the marital status whose working hour per week is maximum
df.groupby(['marital-status'])['hours-per-week'].mean()
```

[54]:

```
marital-status
 Never-married    36.939998
married           43.183628
others            39.667544
Name: hours-per-week, dtype: float64
```

4 Visualization

[55]:

```
import matplotlib.pyplot as plt
import seaborn as sns
```

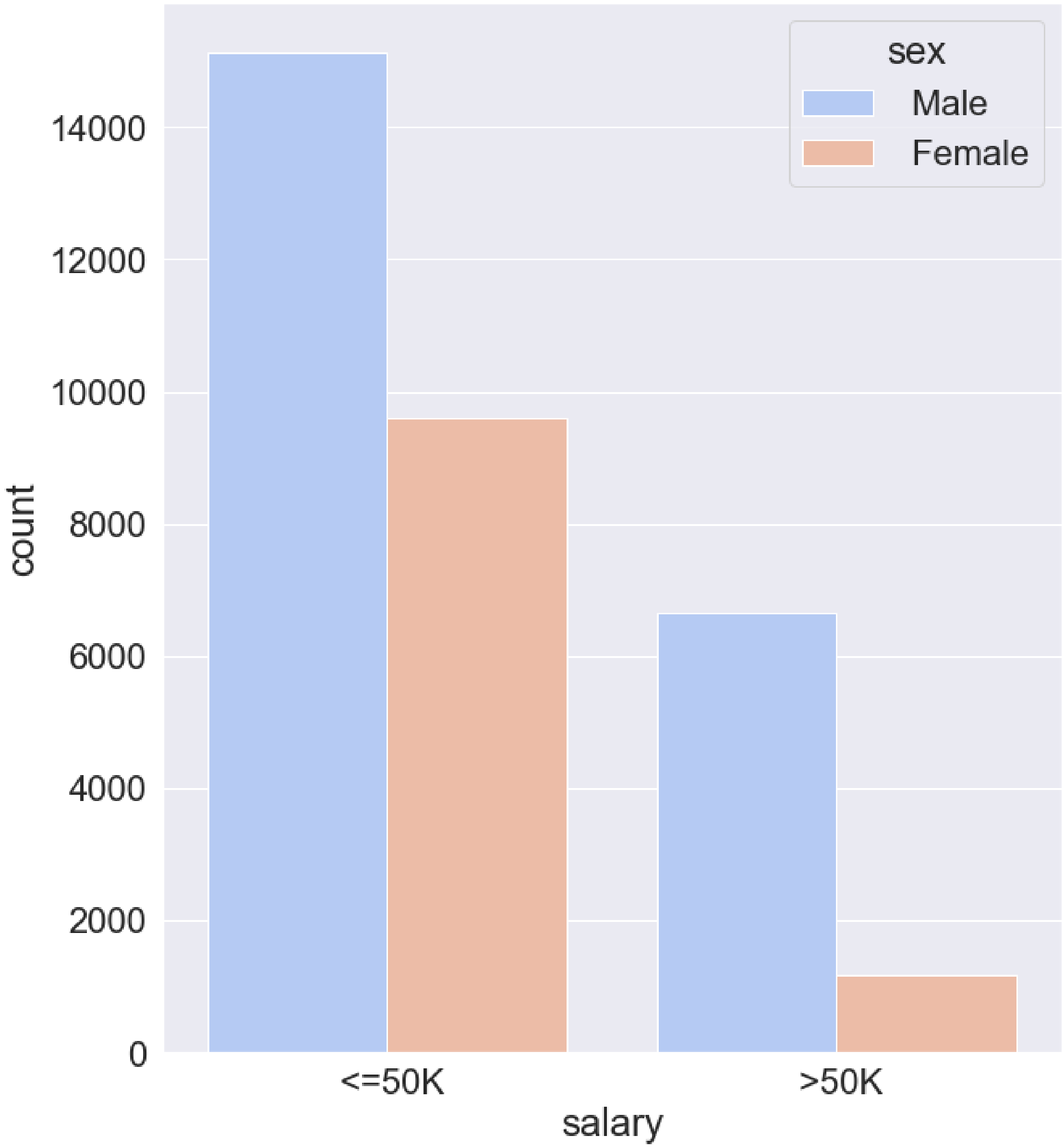
5 Which sex category is earning greater than 50k

[56]:

```
plt.figure(figsize=(10,12))
sns.set(font_scale=2)
sns.countplot(df['salary'],palette='coolwarm',hue='sex',data=df)
plt.show()
```

```
C:\Users\hp\AppData\Roaming\Python\Python38\site-
packages\seaborn\_decorators.py:36: FutureWarning:
```

```
Pass the following variable as a keyword arg: x. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an
explicit keyword will result in an error or misinterpretation.
```



No. of males are earning more than 50k than Females

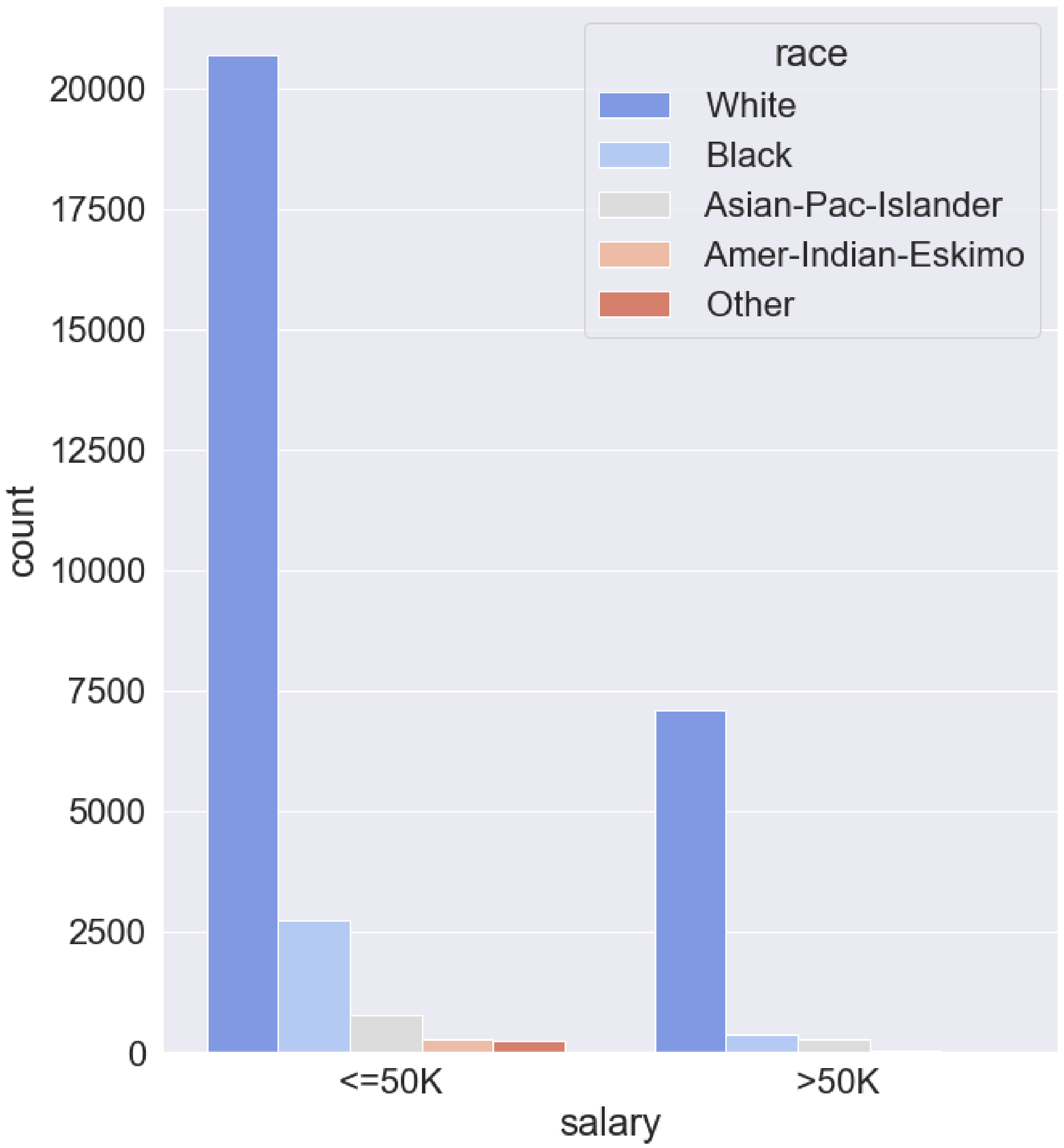
6 Which type of people in race are earning more than others

[57]:

```
plt.figure(figsize=(10,12))
sns.set(font_scale=2)
sns.countplot(df['salary'],palette='coolwarm',hue='race',data=df)
plt.show()
```

```
C:\Users\hp\AppData\Roaming\Python\Python38\site-
packages\seaborn\_decorators.py:36: FutureWarning:

Pass the following variable as a keyword arg: x. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an
explicit keyword will result in an error or misinterpretation.
```



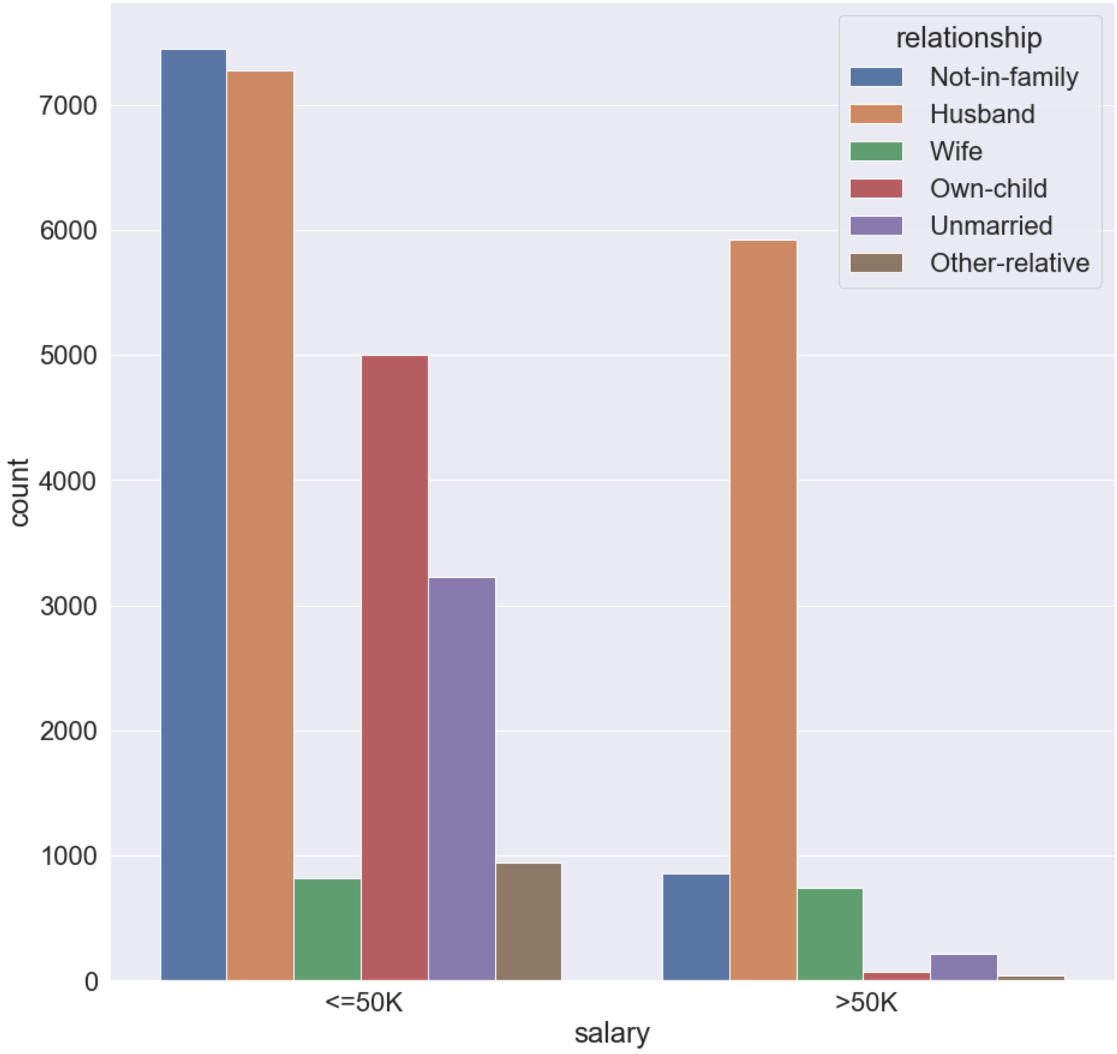
No. of White people are earning more salary than others

7 What is the relationship of people who is earning more than 50k

```
[58]: plt.figure(figsize=(15,15))
sns.set(font_scale=2)
sns.countplot(df['salary'],hue='relationship',data=df);
```

```
C:\Users\hp\AppData\Roaming\Python\Python38\site-
packages\seaborn\_decorators.py:36: FutureWarning:
```

```
Pass the following variable as a keyword arg: x. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an
explicit keyword will result in an error or misinterpretation.
```



People who are Not-in-family are earning more

8 finding the correlation among all the numerical variables

```
[59]: df.corr()
```

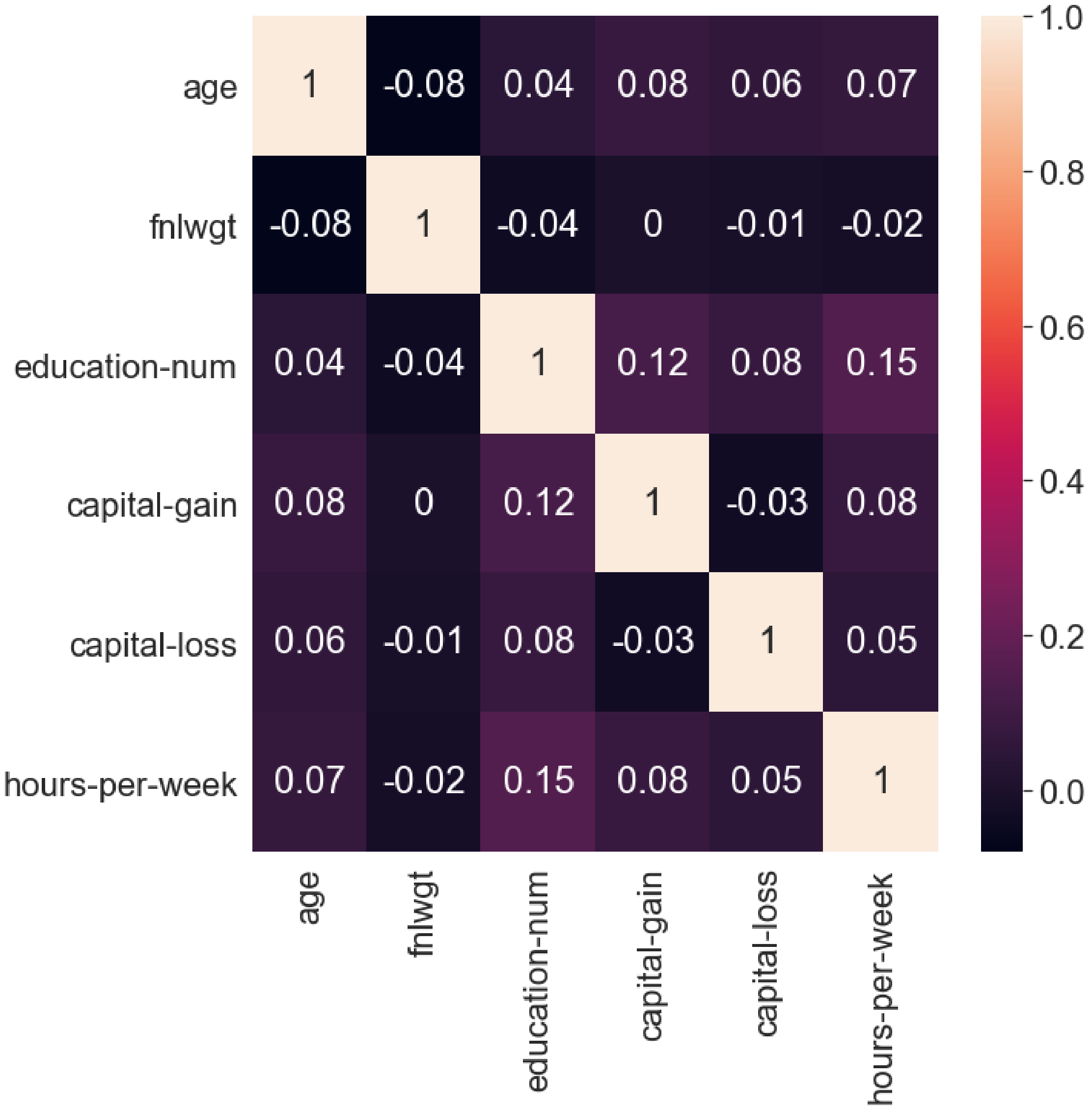
```
[59]:                   age    fnlwgt  education-num  capital-gain  capital-loss  \
      age          1.000000 -0.076646       0.036527      0.077674      0.057775
      fnlwgt      -0.076646  1.000000      -0.043195      0.000432     -0.010252
```

```
education-num   0.036527 -0.043195       1.000000       0.122630       0.079923
capital-gain    0.077674  0.000432       0.122630       1.000000      -0.031615
capital-loss    0.057775 -0.010252       0.079923      -0.031615       1.000000
hours-per-week  0.068756 -0.018768       0.148123       0.078409       0.054256

                hours-per-week
age                   0.068756
fnlwgt               -0.018768
education-num         0.148123
capital-gain          0.078409
capital-loss          0.054256
hours-per-week        1.000000
```

[60]:

```
plt.figure(figsize=(10,10))
sns.set(font_scale=2)
sns.heatmap(np.round(df.corr(),2),annot=True)
plt.show()
```



9 What is the % of education background in the dataset

```
[61]: px.pie(df,values='education-num',names='education',title='% of Education')
```

```
<IPython.core.display.Javascript object>
```

There are maximum people who had done their higher education

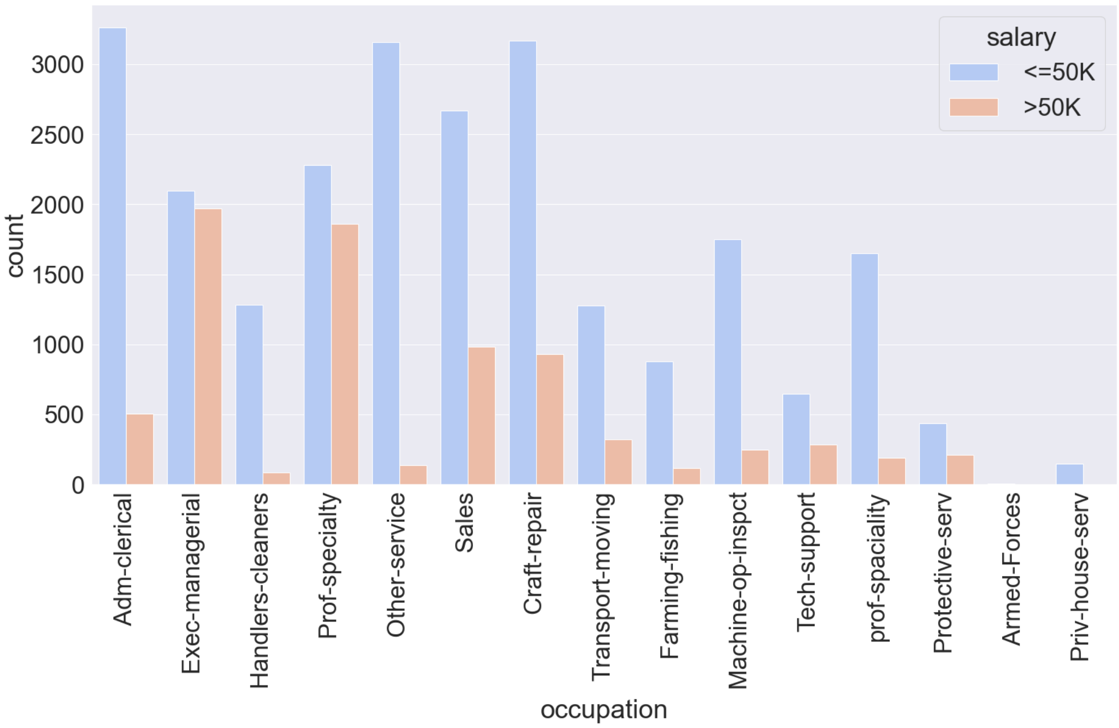
10 Which occupation has maximum salary than other occupations

[62]:

```
plt.figure(figsize=(25,12))
sns.set(font_scale=3)
sns.countplot(df['occupation'],hue='salary',data=df,palette='coolwarm')
plt.xticks(rotation=90);
```

```
C:\Users\hp\AppData\Roaming\Python\Python38\site-
packages\seaborn\_decorators.py:36: FutureWarning:

Pass the following variable as a keyword arg: x. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an
explicit keyword will result in an error or misinterpretation.
```



There are more People whose job role is Executive-manager is earning more than 50k

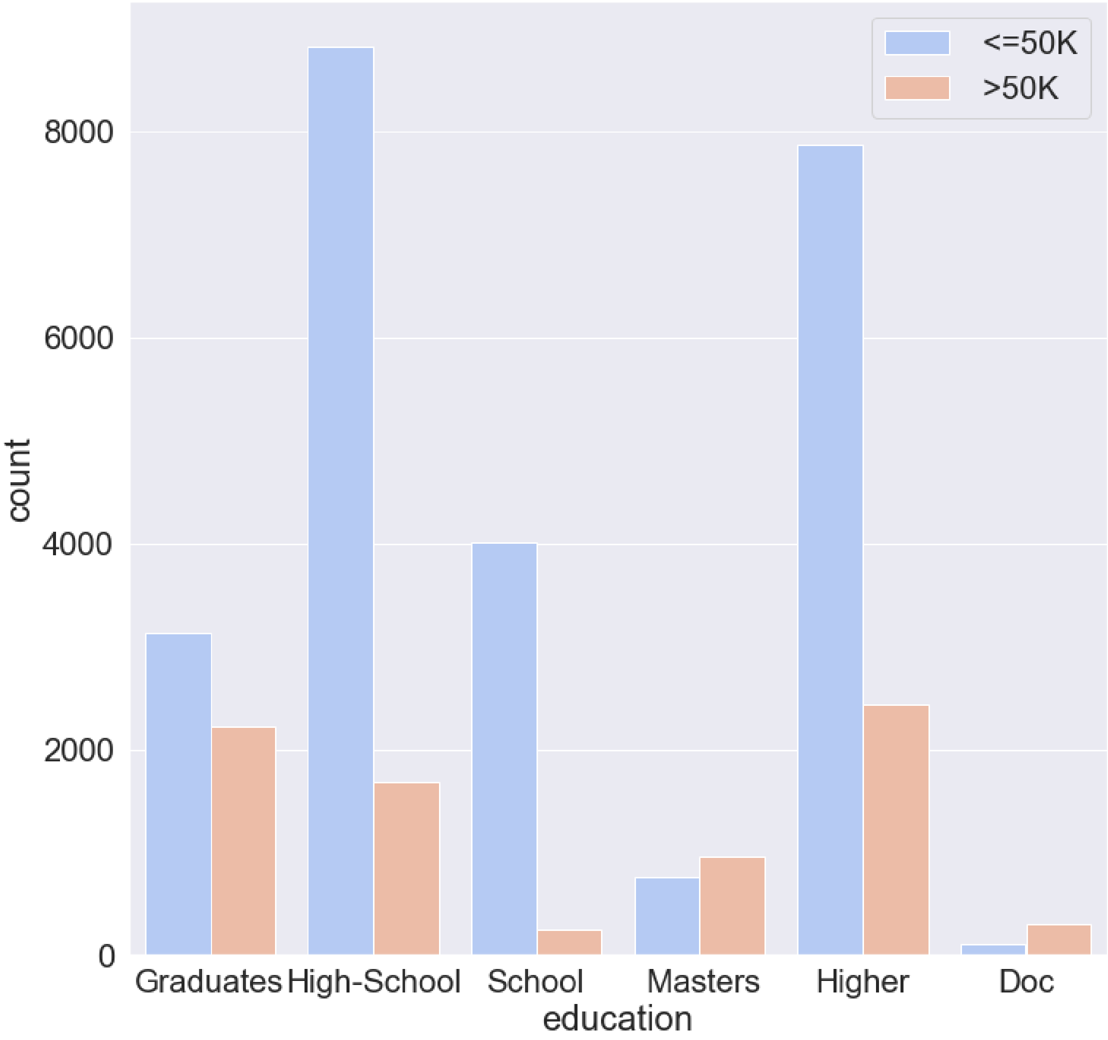
11 How the educational background is related with salary

[63]:

```
plt.figure(figsize=(12,12))
sns.set(font_scale=2)
sns.countplot(df['education'],hue='salary',data=df,palette='coolwarm')
plt.legend(loc='upper right', bbox_to_anchor=(1, 1.0));
```

```
C:\Users\hp\AppData\Roaming\Python\Python38\site-
packages\seaborn\_decorators.py:36: FutureWarning:

Pass the following variable as a keyword arg: x. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an
explicit keyword will result in an error or misinterpretation.
```



People who are from Higher educational background are earning more

people who are from high school educational background are earning less

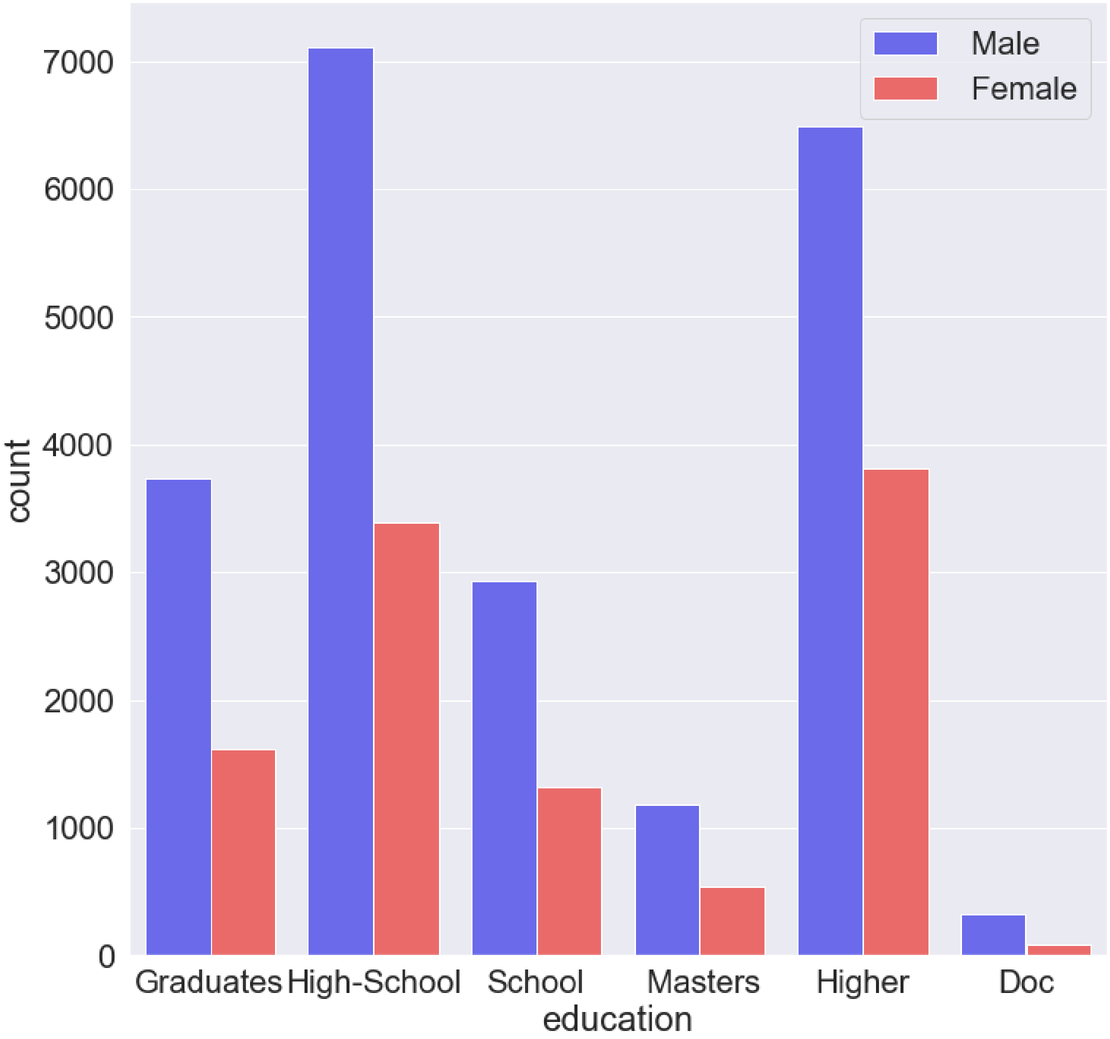
12 Which gender is more educated

[64]:

```
plt.figure(figsize=(12,12))
sns.set(font_scale=2)
sns.countplot(df['education'],hue='sex',data=df,palette='seismic')
plt.legend(loc='upper right', bbox_to_anchor=(1, 1.0));
```

```
C:\Users\hp\AppData\Roaming\Python\Python38\site-
packages\seaborn\_decorators.py:36: FutureWarning:

Pass the following variable as a keyword arg: x. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an
explicit keyword will result in an error or misinterpretation.
```



Males are more educated than Females

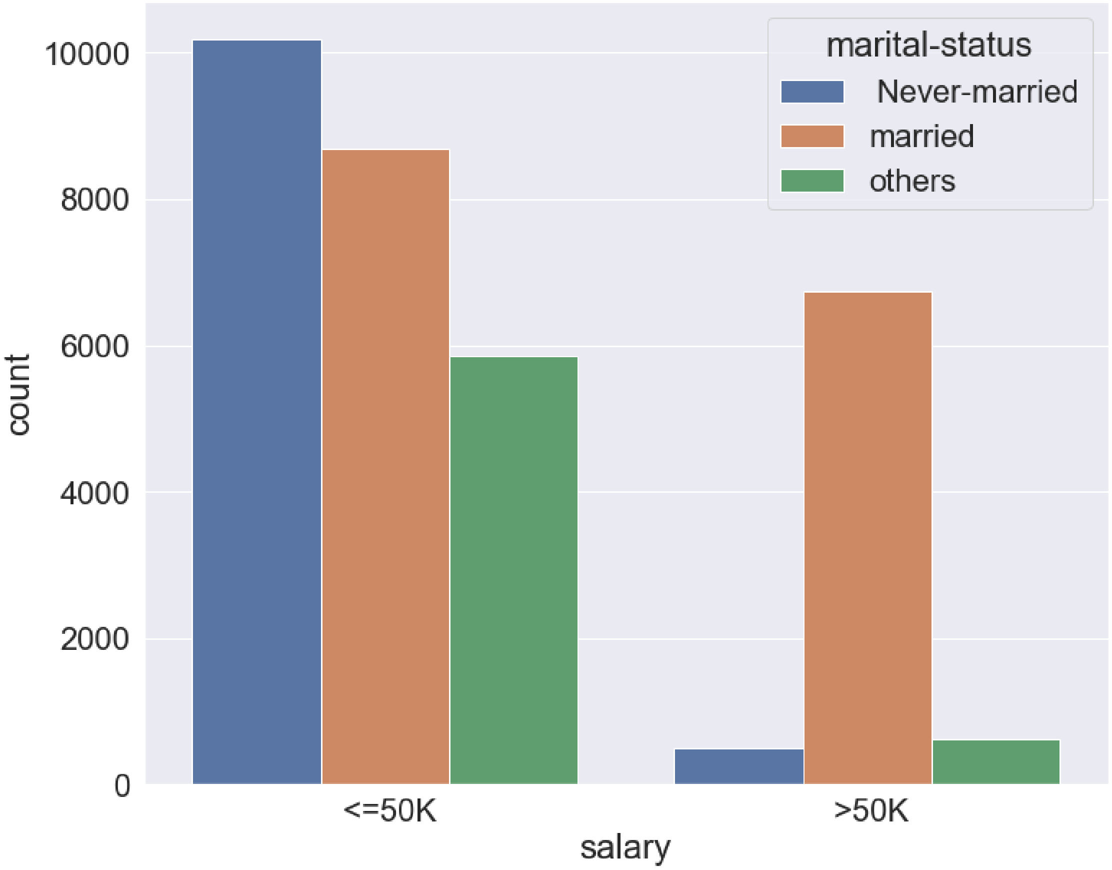
13 Which marital status people are earning more

[65]:

```
#Salary based on martial status
plt.figure(figsize=(12,10))
sns.countplot('salary',data=df,hue='marital-status')
plt.show()
```

```
C:\Users\hp\AppData\Roaming\Python\Python38\site-
packages\seaborn\_decorators.py:36: FutureWarning:

Pass the following variable as a keyword arg: x. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an
explicit keyword will result in an error or misinterpretation.
```

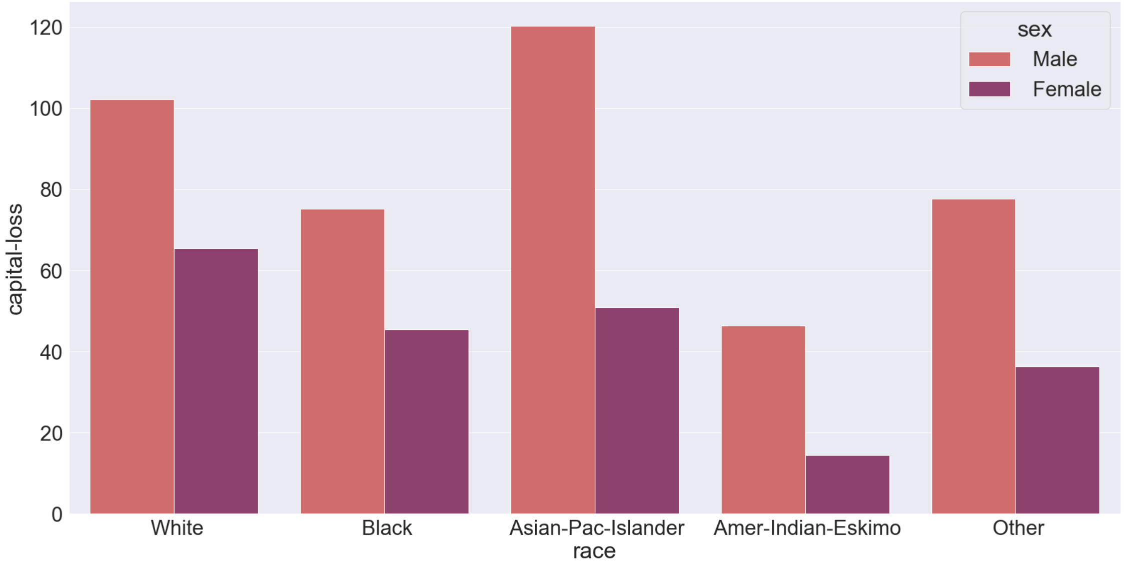


Married people are earning more

14 Which kind of people have maximum capital loss

[67]:

```
plt.figure(figsize=(30,15))
sns.set(font_scale=3)
sns.barplot(x='race',y='capital-loss',data=df,hue='sex',palette='flare',ci=None)
plt.show()
```



Asian-Pac-islander people have more capital loss than others

15 Thank You