In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:

```
pd.set_option("display.max_rows",None)
pd.set_option("display.max_columns",None)
```

In [3]:

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

In [4]:

```
columns = ['age','workclass','fnlwgt','education','education-num','marital-status','occupation','relationship','race','
```

In [5]:

```
df = pd.read_csv(url,names = columns)
df.head(2)
```

Out[5]:

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	native-country	salary
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

In [6]:

```
df.shape
```

Out[6]:

```
(32561, 15)
```

In [7]:

```
df.describe()
```

Out[7]:

	age	fnlwgt	education-num	capital-gain	capital-loss	hours-per-week
count	32561.000000	3.256100e+04	32561.000000	32561.000000	32561.000000	32561.000000
mean	38.581647	1.897784e+05	10.080679	1077.648844	87.303830	40.437456
std	13.640433	1.055500e+05	2.572720	7385.292085	402.960219	12.347429
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	2.370510e+05	12.000000	0.000000	0.000000	45.000000
max	90.000000	1.484705e+06	16.000000	99999.000000	4356.000000	99.000000

In [8]:

```
df.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  salary          32561 non-null  object
dtypes: int64(6), object(9)
memory usage: 3.7+ MB
```

In [9]:
```
df.isnull().sum()
```

Out[9]:
```
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
hours-per-week    0
native-country    0
salary            0
dtype: int64
```

In [10]:
```
df.workclass.unique()
```

Out[10]:
```
array([' State-gov', ' Self-emp-not-inc', ' Private', ' Federal-gov',
       ' Local-gov', ' ?', ' Self-emp-inc', ' Without-pay',
       ' Never-worked'], dtype=object)
```

In [11]:
```
df.workclass.value_counts()
```

Out[11]:
```
 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
```

In [12]:

```
df =df.replace(' Never-worked',' Without-pay')
df['workclass'].value_counts()
```

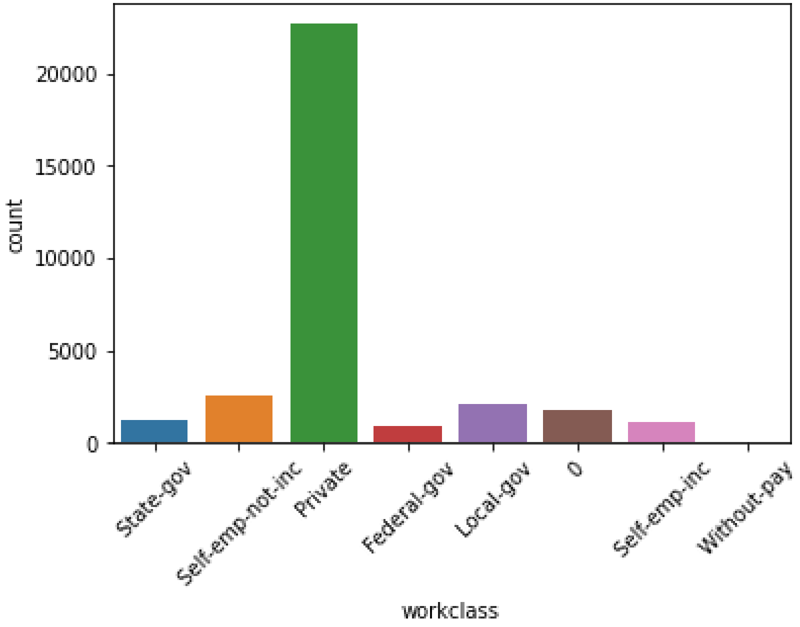
Out[12]:

```
 Private             22696
 Self-emp-not-inc     2541
 Local-gov            2093
 ?                    1836
 State-gov            1298
 Self-emp-inc         1116
 Federal-gov           960
 Without-pay            21
Name: workclass, dtype: int64
```

In [13]:

```
df.replace(' ?',np.nan,inplace= True)
df['workclass'].fillna('0',inplace=True)
```

In [14]:
```
sns.countplot(x = df['workclass'])
plt.xticks(rotation = 45)
plt.show()
```



In [15]:
```
df['salary'].unique()
```

Out[15]:
```
array([' <=50K', ' >50K'], dtype=object)
```

In [16]:

```
salary = {' <=50K': 0 , ' >50K':'1'}
df = df.replace(salary)
df.head(2)
```

Out[16]:

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	native-country	salary
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40	United-States	0
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13	United-States	0

In [17]:

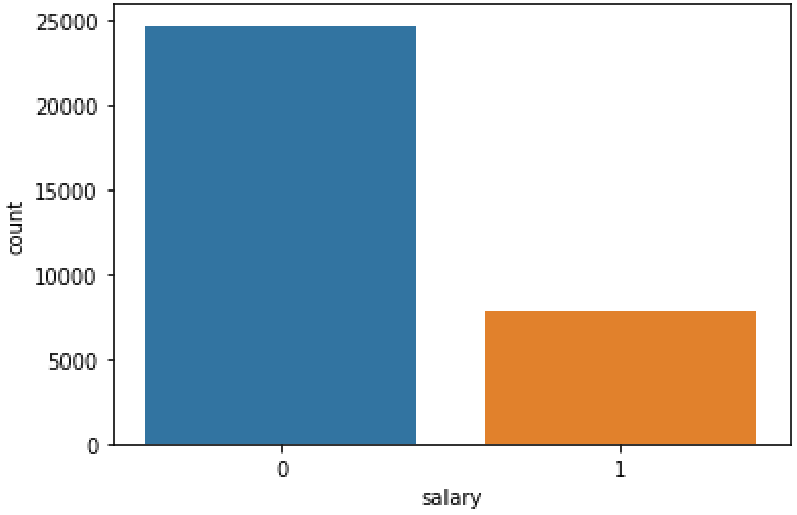
```
df['salary'].value_counts()
```

Out[17]:

```
0    24720
1     7841
Name: salary, dtype: int64
```

```
In [18]: sns.countplot(x=df['salary'])
         plt.xticks(rotation = 0)
```

```
Out[18]: (array([0, 1]), [Text(0, 0, '0'), Text(1, 0, '1')])
```



```
In [19]: df['education'].value_counts()
```

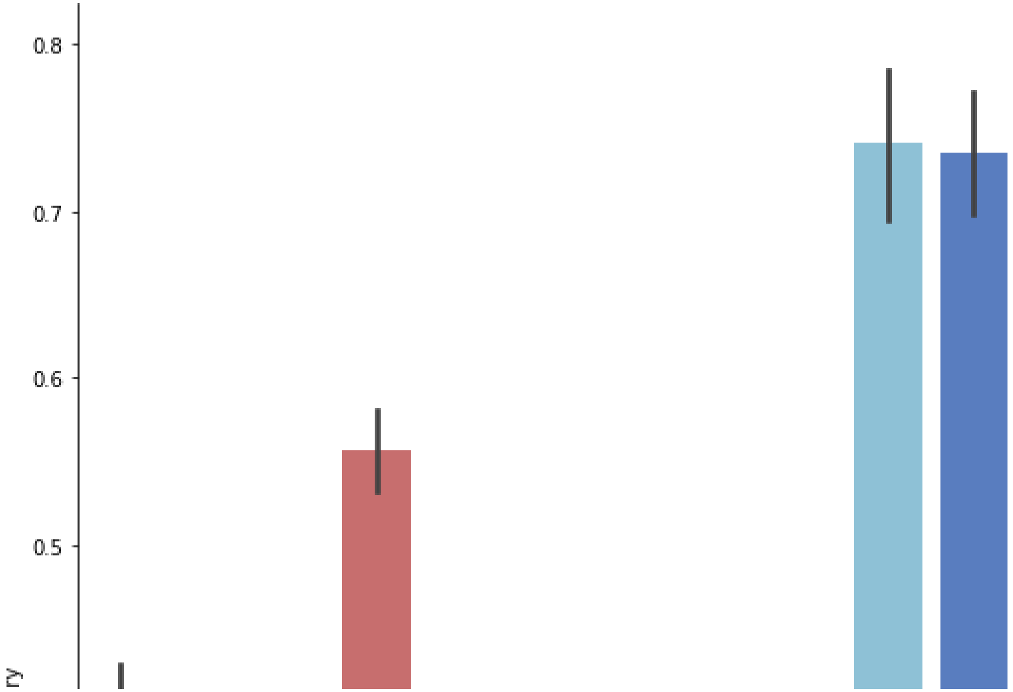
```
Out[19]:  HS-grad         10501
          Some-college     7291
          Bachelors        5355
          Masters          1723
          Assoc-voc        1382
          11th             1175
          Assoc-acdm       1067
          10th              933
          7th-8th           646
          Prof-school       576
          9th               514
          12th              433
          Doctorate         413
          5th-6th           333
          1st-4th           168
          Preschool          51
         Name: education, dtype: int64
```

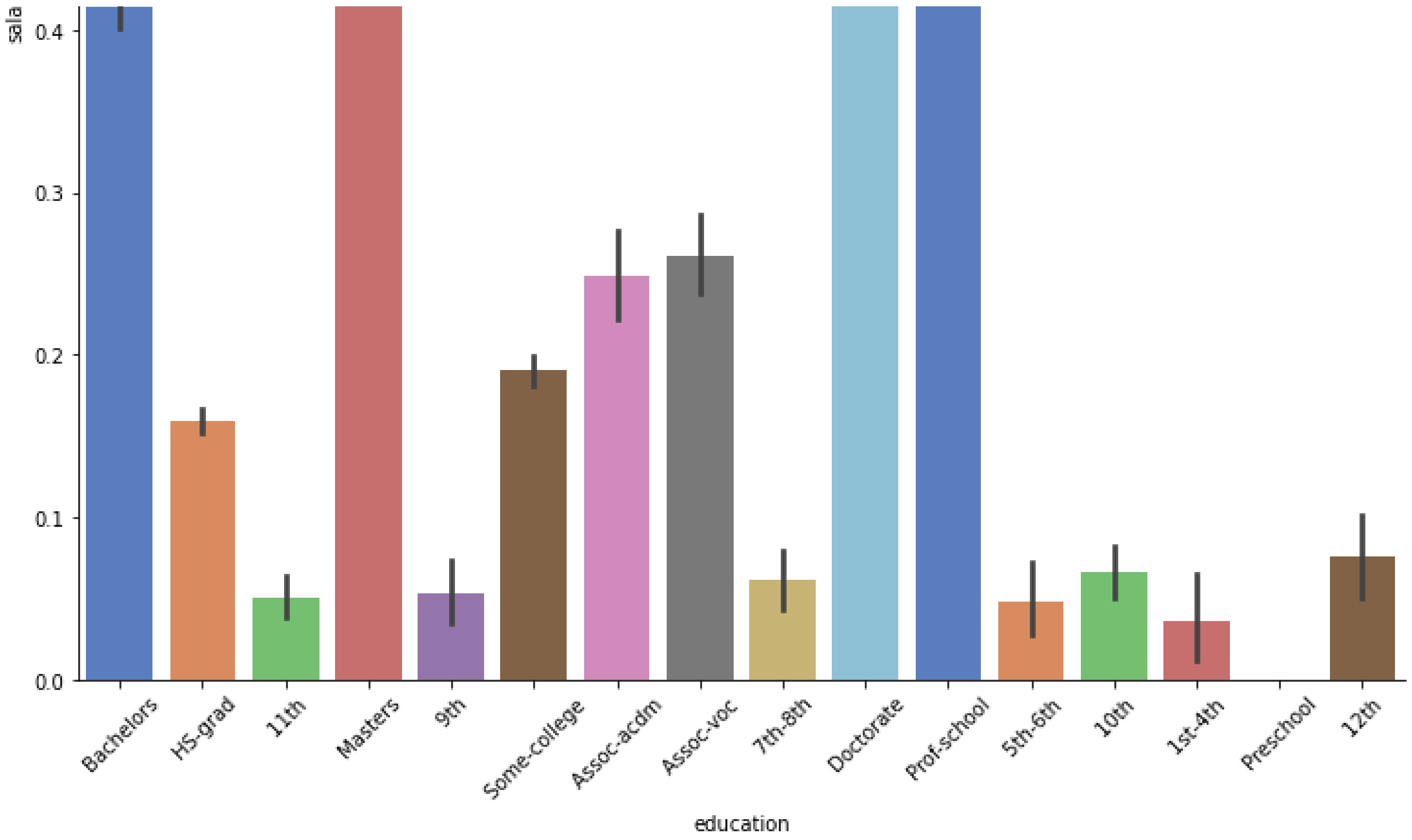
In [20]:

```
sns.catplot(x='education',y=pd.to_numeric(df['salary']),data=df,height=10,palette='muted',kind='bar')
plt.xticks(rotation=45)
```

Out[20]:

```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15]),
 [Text(0, 0, ' Bachelors'),
  Text(1, 0, ' HS-grad'),
  Text(2, 0, ' 11th'),
  Text(3, 0, ' Masters'),
  Text(4, 0, ' 9th'),
  Text(5, 0, ' Some-college'),
  Text(6, 0, ' Assoc-acdm'),
  Text(7, 0, ' Assoc-voc'),
  Text(8, 0, ' 7th-8th'),
  Text(9, 0, ' Doctorate'),
  Text(10, 0, ' Prof-school'),
  Text(11, 0, ' 5th-6th'),
  Text(12, 0, ' 10th'),
  Text(13, 0, ' 1st-4th'),
  Text(14, 0, ' Preschool'),
  Text(15, 0, ' 12th')])
```





In [21]:
```
df['marital-status'].value_counts()
```

Out[21]:
```
 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
```

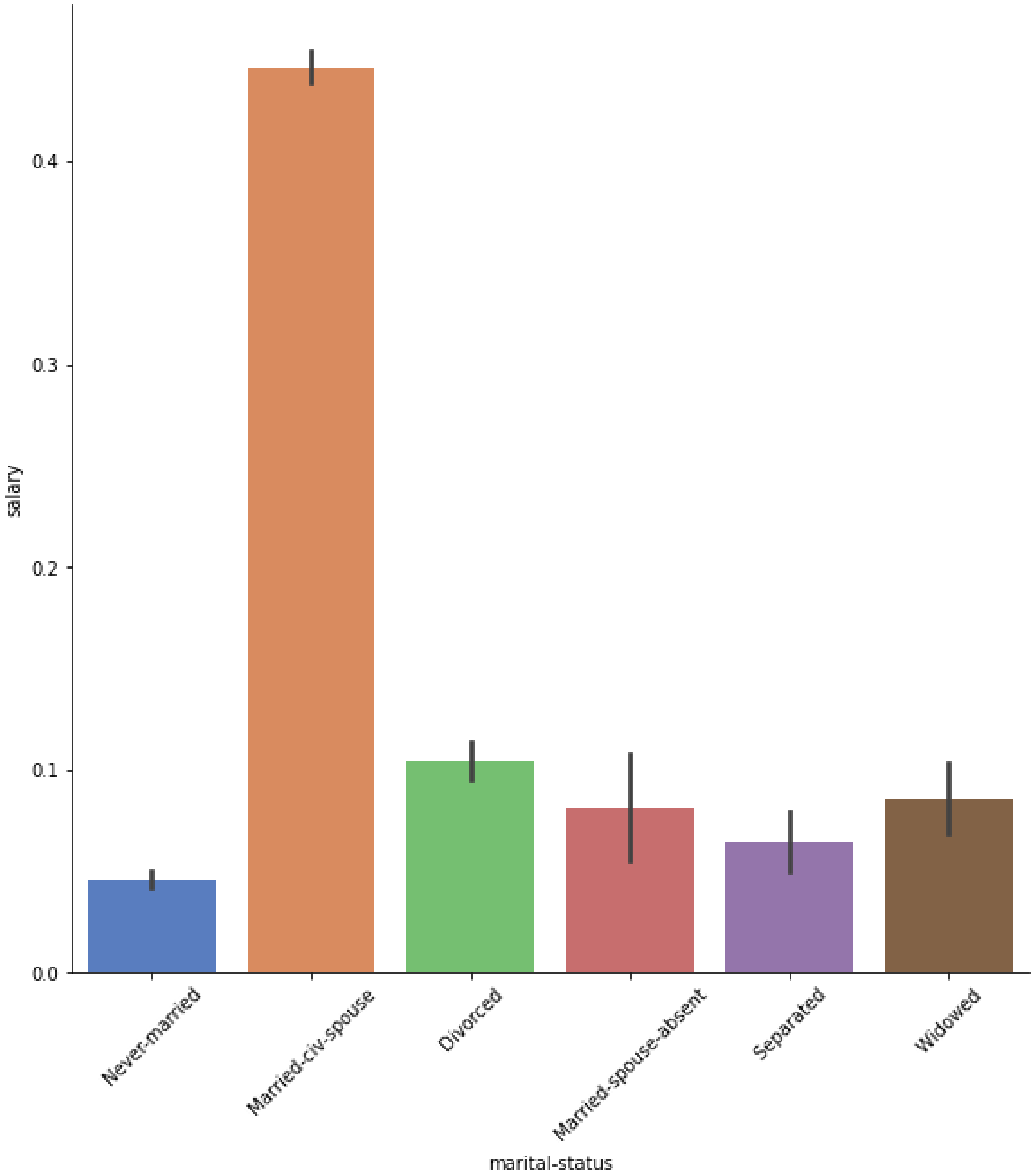
In [22]:
```
df['marital-status'].replace(' Married-AF-spouse', ' Married-civ-spouse',inplace=True)
```

In [23]:

```
sns.catplot(x='marital-status',y=pd.to_numeric(df['salary']),data=df,palette='muted',kind='bar',height=8)
plt.xticks(rotation=45)
```

Out[23]:

```
(array([0, 1, 2, 3, 4, 5]),
 [Text(0, 0, ' Never-married'),
  Text(1, 0, ' Married-civ-spouse'),
  Text(2, 0, ' Divorced'),
  Text(3, 0, ' Married-spouse-absent'),
  Text(4, 0, ' Separated'),
  Text(5, 0, ' Widowed')])
```


0.0
0.1
0.2
0.3
0.4
salary
Never-married
Married-civ-spouse
Divorced
Married-spouse-absent
Separated
Widowed
marital-status

In [24]:

```python
df['occupation'].fillna('0',inplace=True)
df['occupation'].value_counts()
```

Out[24]:

```
 Prof-specialty       4140
 Craft-repair         4099
 Exec-managerial      4066
 Adm-clerical         3770
 Sales                3650
 Other-service        3295
 Machine-op-inspct    2002
0                     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
```

In [25]:

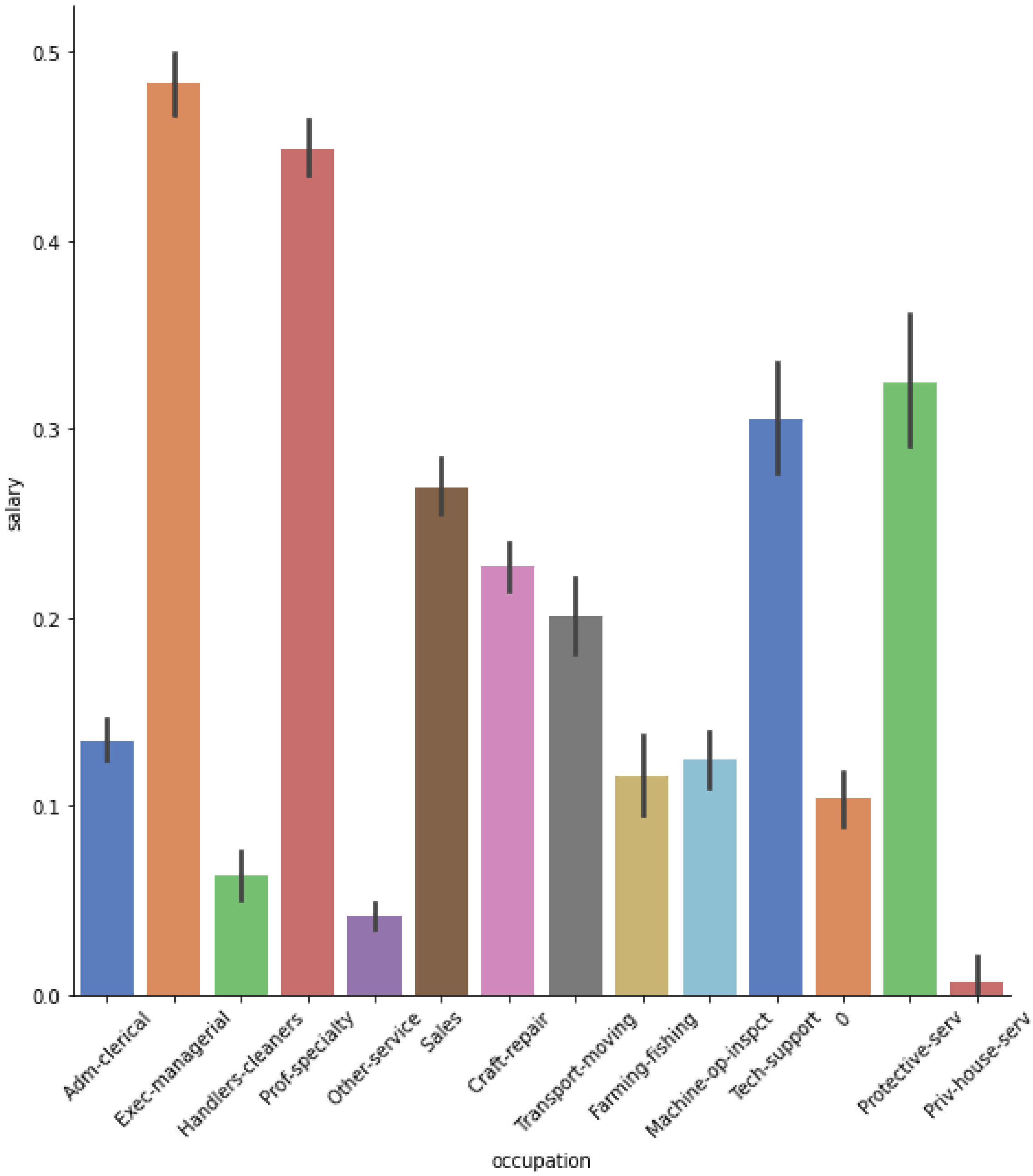
```python
df['occupation'].replace(' Armed-Forces','0',inplace=True)
df['occupation'].value_counts()
```

Out[25]:

```
 Prof-specialty       4140
 Craft-repair         4099
 Exec-managerial      4066
 Adm-clerical         3770
 Sales                3650
 Other-service        3295
 Machine-op-inspct    2002
0                     1852
 Transport-moving     1597
 Handlers-cleaners    1370
 Farming-fishing       994
 Tech-support          928
 Protective-serv       649
 Priv-house-serv       149
Name: occupation, dtype: int64
```

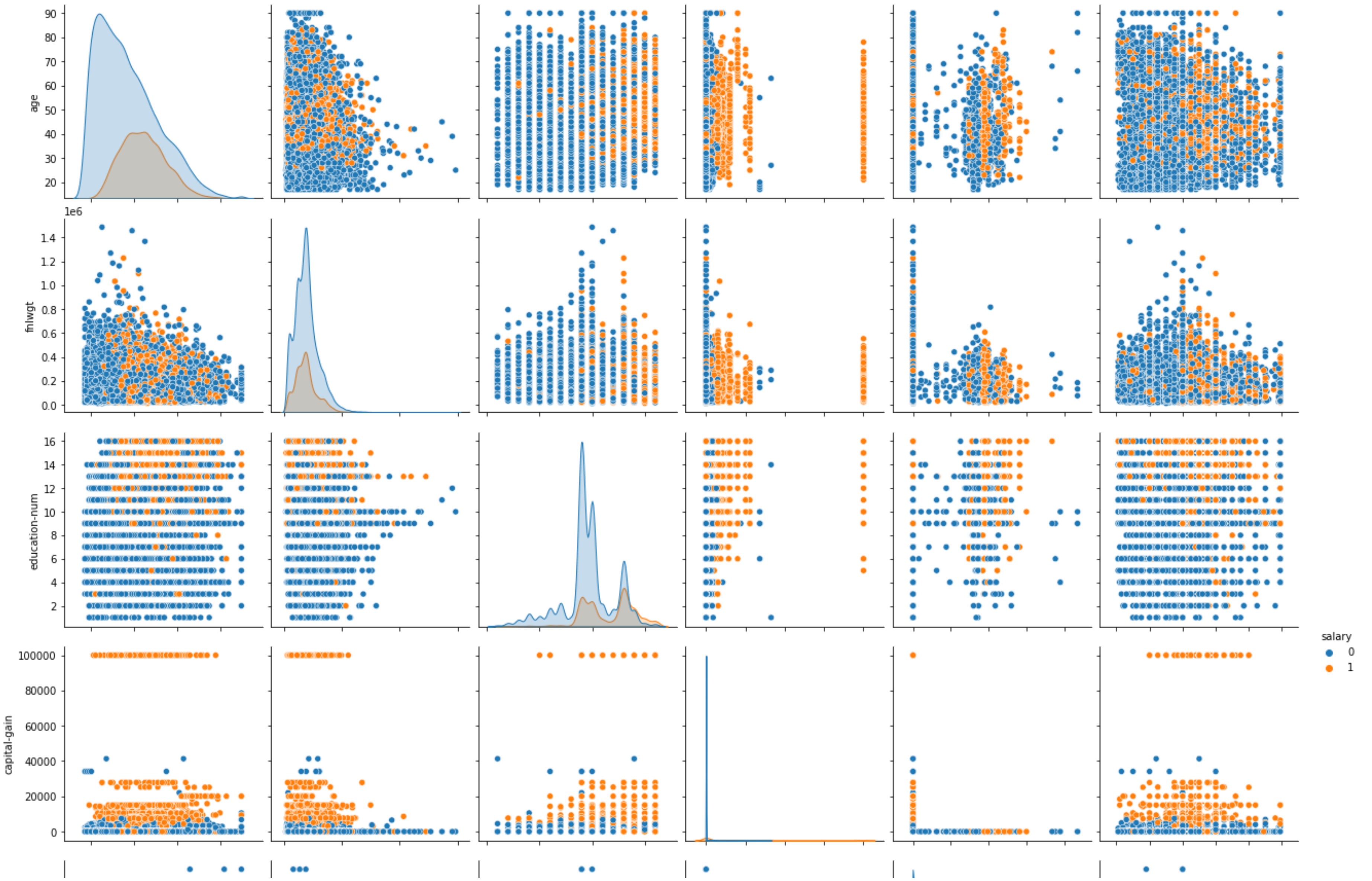
In [26]:
```
sns.catplot(x='occupation',y=pd.to_numeric(df['salary']),data=df,palette='muted',kind='bar',height=8)
plt.xticks(rotation=45)
```

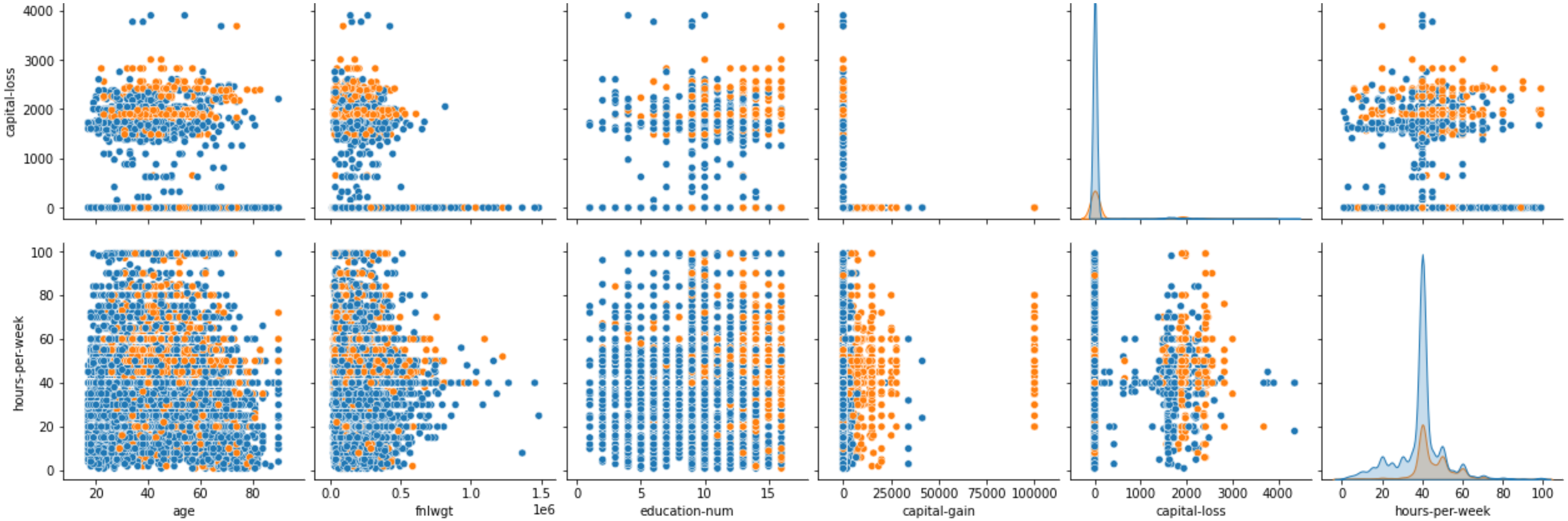
Out[26]:
```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13]),
 [Text(0, 0, ' Adm-clerical'),
  Text(1, 0, ' Exec-managerial'),
  Text(2, 0, ' Handlers-cleaners'),
  Text(3, 0, ' Prof-specialty'),
  Text(4, 0, ' Other-service'),
  Text(5, 0, ' Sales'),
  Text(6, 0, ' Craft-repair'),
  Text(7, 0, ' Transport-moving'),
  Text(8, 0, ' Farming-fishing'),
  Text(9, 0, ' Machine-op-inspct'),
  Text(10, 0, ' Tech-support'),
  Text(11, 0, '0'),
  Text(12, 0, ' Protective-serv'),
  Text(13, 0, ' Priv-house-serv')])
```


0.5
0.4
0.3
0.2
0.1
0.0
salary
Adm-clerical
Exec-managerial
Handlers-cleaners
Prof-specialty
Other-service
Sales
Craft-repair
Transport-moving
Farming-fishing
Machine-op-inspct
Tech-support
0
Protective-serv
Priv-house-serv
occupation

In [28]:
```
sns.pairplot(df,hue='salary',height=3)
plt.plot()
```

Out[28]:
```
[]
```



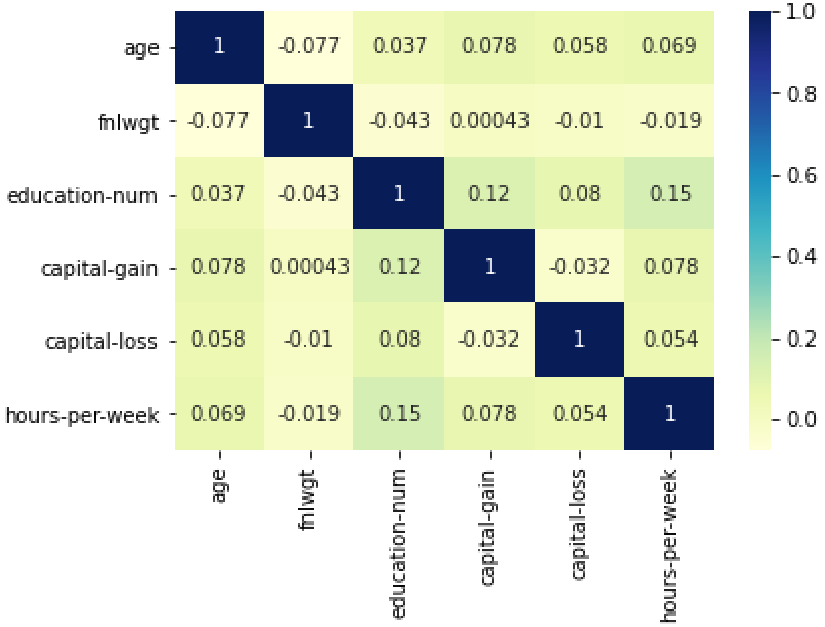

capital-loss
hours-per-week
4000
3000
2000
1000
0
100
80
60
40
20
0
20
40
60
80
age
0.0
0.5
1.0
1.5
1e6
fnlwgt
0
5
10
15
education-num
0
25000
50000
75000
100000
capital-gain
0
1000
2000
3000
4000
capital-loss
0
20
40
60
80
100
hours-per-week

In [34]:

```
corr = df.corr()
sns.heatmap(corr,annot = True,cmap='YlGnBu')
```

Out[34]:

```
<AxesSubplot:>
```



In [35]:

```
df.drop('fnlwgt',axis=1,inplace=True)
```

In [36]:

```
df.head(n=2)
```

Out[36]:

	age	workclass	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	native-country	salary
0	39	State-gov	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40	United-States	0
1	50	Self-emp-not-inc	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13	United-States	0

In [37]:
```
X = df.drop('salary',axis=1)
y = pd.to_numeric(df['salary'])
```

In [38]:
```
X_d = pd.get_dummies(X)
X_d.head(2)
```

Out[38]:

	age	education-num	capital-gain	capital-loss	hours-per-week	workclass_Federal-gov	workclass_Local-gov	workclass_Private	workclass_Self-emp-inc	workclass_Self-emp-not-inc	workclass_State-gov	workclass_Without-pay	workclass_0
0	39	13	2174	0	40	0	0	0	0	0	1	0	0
1	50	13	0	0	13	0	0	0	0	1	0	0	0

In [39]:
```
from sklearn.model_selection import train_test_split,GridSearchCV,StratifiedKFold
x_train,x_test,y_train,y_test = train_test_split(X_d,y,test_size=0.3,random_state=101)
```

In [41]:
```
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
```

In [42]:
```
classifier = [DecisionTreeClassifier(random_state=42),RandomForestClassifier(random_state=42)]
```

In [43]:
```
dt_grid_param = { "min_samples_split" : range(10,500,20),
                "max_depth": range(1,20,2)

}
```

In [46]:
```
rf_grid_param = {"max_features": [1,3,10],
                "min_samples_split":[2,3,10],
                "min_samples_leaf":[1,3,10],
                "bootstrap":[False],
                "n_estimators":[100,300],
                "criterion":["gini"]}
```

In [47]:

```
classifier_param = [dt_grid_param,rf_grid_param]
```

In [48]:

```
cv_result = []
best_estimators = []
for i in range(len(classifier)):
    clf = GridSearchCV(classifier[i], param_grid=classifier_param[i], cv = StratifiedKFold(n_splits = 10), scoring = "a
    clf.fit(x_train,y_train)
    cv_result.append(clf.best_score_)
    best_estimators.append(clf.best_estimator_)
    print(cv_result[i])
```

```
Fitting 10 folds for each of 250 candidates, totalling 2500 fits
0.8584153560733778
Fitting 10 folds for each of 54 candidates, totalling 540 fits
0.8646456971740454
```

In [50]:

```
cv_results = pd.DataFrame({"Cross Validation Means":cv_result, "ML Models":["DecisionTreeClassifier", "RandomForestClas
```

In [55]:

```
cv_results
```

Out[55]:

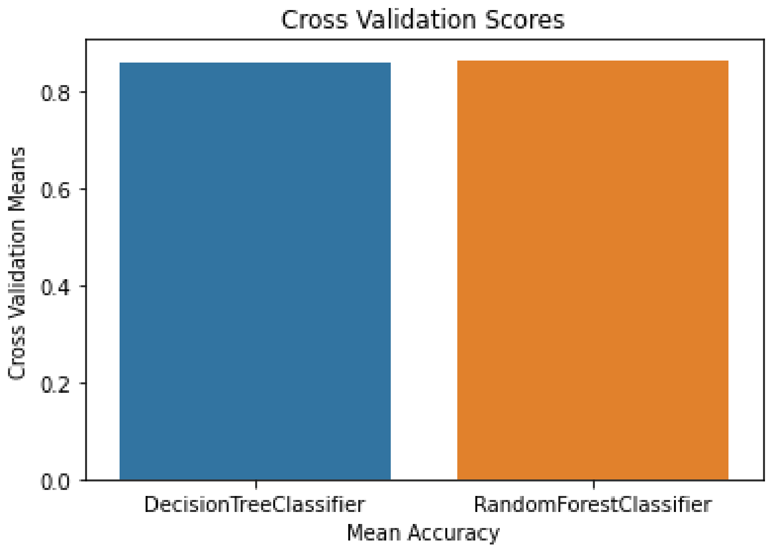
	Cross Validation Means	ML Models
0	0.858415	DecisionTreeClassifier
1	0.864646	RandomForestClassifier

In [56]:

```
g = sns.barplot(y="Cross Validation Means", x="ML Models", data = cv_results)
g.set_xlabel("Mean Accuracy")
g.set_title("Cross Validation Scores")
```

Out[56]:

```
Text(0.5, 1.0, 'Cross Validation Scores')
```



In [ ]: