

Question 2a: Write a function `plot_categorical_distribution` to plot the distribution of the column 'education' as a histogram.

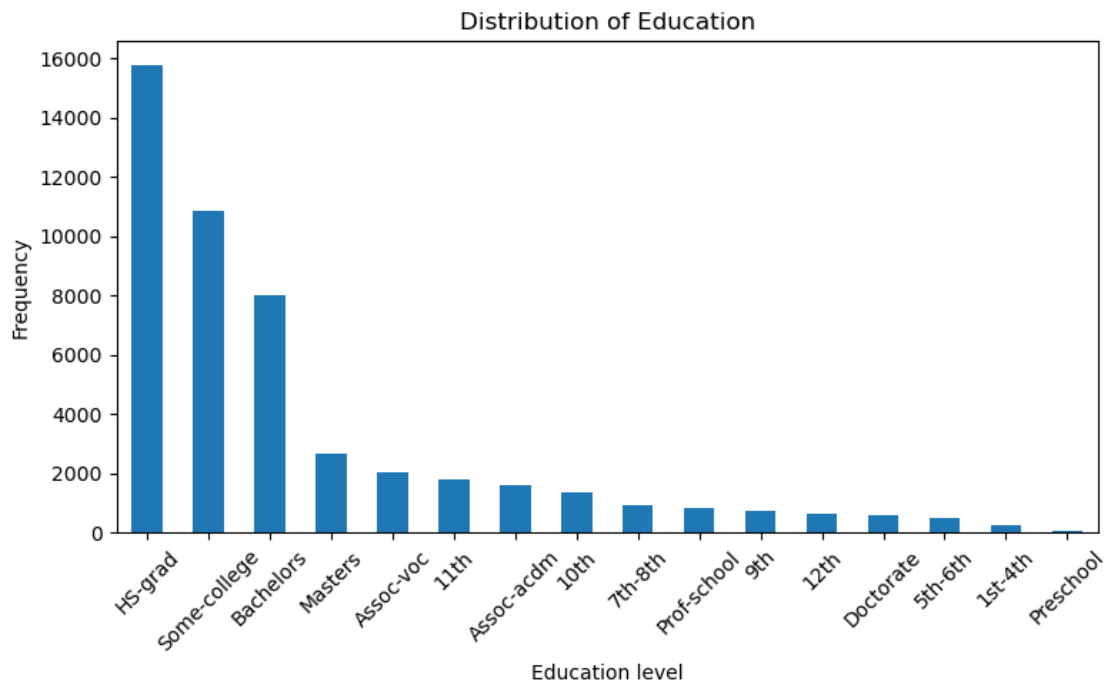
You can use Pandas `.plot()` method for this. Look at the DataFrame `.value_counts` method as well. See class examples for how to add labels and titles.

```
In [18]: def plot_categorical_distribution(df):  
         category = df['education'].value_counts()  
         ax = category.plot(kind = 'bar',figsize =(8,5))  
         plt.xlabel('Education level')  
         plt.ylabel('Frequency')  
         plt.title('Distribution of Education')  
         plt.xticks(rotation=45)  
         plt.tight_layout()  
         plt.show()
```

...

Out[18]: Ellipsis

```
In [19]: plot_categorical_distribution(df)
```



Question 2b: Write a function `plot_age_hours_scatter` that creates a scatter plot of 'age' vs 'hours-per-week', coloring points by 'income'.

You'll want to look at Matplotlib's `pyplot.scatter()` for this one.

```
In [20]: def plot_age_hours_scatter(df):  
    plt.figure(figsize=(8,6))  
  
    for income_class,group in df.groupby('income'):  
        plt.scatter(group['age'],group['hours-per-week'],label = income_class,alpha=0.7)  
    plt.xlabel('Age')  
    plt.ylabel('Hours-per-Week')  
    plt.title('Scatter Plot by Income ')  
    plt.legend(title = 'income')  
    plt.tight_layout()  
    plt.show()
```

```
In [21]: plot_age_hours_scatter(df)
```



Question 2c: Write a function `plot_income_by_marital_status` which plots a stacked bar chart that shows the proportion of income levels for each 'marital-status' category.

```
In [22]: def plot_income_by_marital_status(df):
          ctab = pd.crosstab(df['marital-status'],df['income'])
          proportions = ctab.div(ctab.sum(axis=1),axis=0)
          proportions.plot(kind='bar',stacked=True,figsize=(8,6))

          plt.xlabel('Marital status')
          plt.ylabel('proportion')
          plt.title('Proportion of Income')
          plt.xticks(rotation=45)
          plt.legend(title = 'income')
          plt.tight_layout()
          plt.show()
```

```
In [23]: plot_income_by_marital_status(df)
```

