Demographic Data Analyzer

In this challenge you must analyze demographic data using Pandas. You are given a dataset of demographic data that was extracted from the 1994 Census database. Here is a sample of what the data looks like:

You must use Pandas to answer the following questions:

- How many people of each race are represented in this dataset? This should be a Pandas series with race names as the index labels. (`race` column)
- What is the average age of men?
- What is the percentage of people who have a Bachelor's degree?
- What percentage of people with advanced education (`Bachelors` , `Masters` , or `Doctorate`) make more than 50K?
- What percentage of people without advanced education make more than 50K?
- What is the minimum number of hours a person works per week?
- What percentage of the people who work the minimum number of hours per week have a salary of more than 50K?
- What country has the highest percentage of people that earn >50K and what is that percentage?
- Identify the most popular occupation for those who earn >50K in India.

Use the starter code in the file `demographic_data_anaylizer` . Update the code so all variables set to "None" are set to the appropriate calculation or code. Round all decimals to the nearest tenth.

```
In [1]: import pandas as pd
        import os
```

```
In [2]: os.getcwd()

Out[2]: 'C:\\Users\\ANAND\\FreeCodeCamp\\DA using python projects'
```

```
In [3]: os.chdir('C:\\Users\\ANAND\\FreeCodeCamp')
```

```
In [4]: df = pd.read_csv('csv data/adult.data.csv')
```

In [5]:
```
df.head()
```

Out[5]:

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

In [6]:
```
race_count = df['race'].value_counts()
race_count.index
```

Out[6]:
```
Index(['White', 'Black', 'Asian-Pac-Islander', 'Amer-Indian-Eskimo
', 'Other'], dtype='object')
```

In [7]:
```
df_men = df[df['sex'] == 'Male']
round(df_men['age'].mean(),1)
```

Out[7]:
```
39.4
```

In [8]:
```
c1 = df['education'].count()
```

In [9]:
```
df_BE = df[df['education'] == 'Bachelors']
c2= df_BE['education'].count()
```

In [10]:
```
percentage_bachelors = round((c2/c1)*100,1)
percentage_bachelors
```

Out[10]:
```
16.4
```

In [11]:
```
df_adv_edu = df[(df['education'] == 'Bachelors') | (df['education']
== 'Masters') | (df['education'] == 'Doctorate')]
c3 = df_adv_edu['education'].count()
```

In [12]:
```
df_50K = df_adv_edu[df_adv_edu['salary'] == '>50K']
c4 = df_50K['salary'].count()
```

In [16]:
```
high_edu_rich = round((c4/c3*100),1)
```

In [20]:
```
lower_education = c1-c3
lower_education
```

Out[20]:
```
25070
```

In [34]:

```
df_low_edu = df[(df['education'] != 'Bachelors') & (df['education']
!= 'Masters') & (df['education'] != 'Doctorate')]
df_low_edu['education'].count()
```

Out[34]:

```
25070
```

In [40]:

```
c5 = df_low_edu[df_low_edu['salary'] == '>50K'].salary.count()
```

In [41]:

```
round((c5/lower_education*100),1)
```

Out[41]:

```
17.4
```

In [46]:

```
df_min_hrs = df[(df['hours-per-week'] == df['hours-per-week'].min
())]
```

In [55]:

```
num_min_workers=df_min_hrs['salary'].count()
```

In [56]:

```
c6=df_min_hrs[df_min_hrs['salary'] == '>50K'].salary.count()
```

In [58]:

```
rich_percentage = round((c6/num_min_workers*100),1)
rich_percentage
```

Out[58]:

```
10.0
```

```
In [59]: df
```

Out[59]:

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife
...	...	...	...	...	...	...	...	...
32556	27	Private	257302	Assoc-acdm	12	Married-civ-spouse	Tech-support	Wife
32557	40	Private	154374	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband
32558	58	Private	151910	HS-grad	9	Widowed	Adm-clerical	Unmarried
32559	22	Private	201490	HS-grad	9	Never-married	Adm-clerical	Own-child
32560	52	Self-emp-inc	287927	HS-grad	9	Married-civ-spouse	Exec-managerial	Wife

32561 rows × 15 columns

```
In [65]: df_rich = df[df['salary'] == '>50K']
         df_rich['native-country'].value_counts().index[0]
```

```
Out[65]: 'United-States'
```

```
In [71]: highest_country_percent = round((df_rich['native-country'].value_cou
         nts()[0]/df_rich['salary'].count() * 100),1)
```

```
Out[71]: 91.5
```

```
In [78]: df_rich[df_rich['native-country']=='India'].occupation.value_counts
         ().index[0]
```

```
Out[78]: 'Prof-specialty'
```

```
In [102]: def calculate_demographic_data(print_data=True):
              # Read data from file
              df = pd.read_csv('csv data/adult.data.csv')

              # How many of each race are represented in this dataset? This sh
          ould be a Pandas series with race names as the index labels.
              race_count = df['race'].value_counts()

              # What is the average age of men?
              df_men = df[df['sex'] == 'Male']
              average_age_men = round(df_men['age'].mean(),1)

              # What is the percentage of people who have a Bachelor's degree?
              c1 = df['education'].count()
              df_BE = df[df['education'] == 'Bachelors']
              c2 = df_BE['education'].count()
              percentage_bachelors = round((c2/c1)*100,1)

              # What percentage of people with advanced education (`Bachelors
          `, `Masters`, or `Doctorate`) make more than 50K?
              # What percentage of people without advanced education make more
          than 50K?

              df_adv_edu = df[(df['education'] == 'Bachelors') | (df['educatio
          n'] == 'Masters') | (df['education'] == 'Doctorate')]
              higher_education = df_adv_edu['education'].count()
              df_50K = df_adv_edu[df_adv_edu['salary'] == '>50K']
              c4 = df_50K['salary'].count()
              # with and without `Bachelors`, `Masters`, or `Doctorate`

              df_low_edu = df[(df['education'] != 'Bachelors') & (df['educatio
          n'] != 'Masters') & (df['education'] != 'Doctorate')]
              lower_education = df_low_edu['education'].count()
              c5 = df_low_edu[df_low_edu['salary'] == '>50K'].salary.count()

              # percentage with salary >50K
              higher_education_rich = round((c4/higher_education*100),1)
              lower_education_rich = round((c5/lower_education*100),1)

              # What is the minimum number of hours a person works per week (h
          ours-per-week feature)?
              min_work_hours =  df['hours-per-week'].min()

              # What percentage of the people who work the minimum number of h
          ours per week have a salary of >50K?
              num_min_workers = df_min_hrs['salary'].count()
              c6 = df_min_hrs[df_min_hrs['salary'] == '>50K'].salary.count()
              rich_percentage = round((c6/num_min_workers*100),1)

              # What country has the highest percentage of people that earn >5
          0K?
              df_rich = df[df['salary'] == '>50K']
              abc = df_rich['native-country'].value_counts()
              cab = df['native-country'].value_counts()
              percent = round((abc/cab*100),1)
              highest_earning_country = percent[percent == percent.max()].inde
          x
              highest_earning_country_percentage = percent.max()
```

```
    # Identify the most popular occupation for those who earn >50K in India.
    top_IN_occupation = df_rich[df_rich['native-country']=='India'].occupation.value_counts().index[0]

    # DO NOT MODIFY BELOW THIS LINE

    if print_data:
        print("Number of each race:\n", race_count)
        print("Average age of men:", average_age_men)
        print(f"Percentage with Bachelors degrees: {percentage_bachelors}%")
        print(f"Percentage with higher education that earn >50K: {higher_education_rich}%")
        print(f"Percentage without higher education that earn >50K: {lower_education_rich}%")
        print(f"Min work time: {min_work_hours} hours/week")
        print(f"Percentage of rich among those who work fewest hours: {rich_percentage}%")
        print("Country with highest percentage of rich:", highest_earning_country)
        print(f"Highest percentage of rich people in country: {highest_earning_country_percentage}%")
        print("Top occupations in India:", top_IN_occupation)

    return {
        'race_count': race_count,
        'average_age_men': average_age_men,
        'percentage_bachelors': percentage_bachelors,
        'higher_education_rich': higher_education_rich,
        'lower_education_rich': lower_education_rich,
        'min_work_hours': min_work_hours,
        'rich_percentage': rich_percentage,
        'highest_earning_country': highest_earning_country,
        'highest_earning_country_percentage':
        highest_earning_country_percentage,
        'top_IN_occupation': top_IN_occupation
    }
```

In [103]:

```
calculate_demographic_data()
```

```
Number of each race:
 White                 27816
Black                  3124
Asian-Pac-Islander     1039
Amer-Indian-Eskimo      311
Other                   271
Name: race, dtype: int64
Average age of men: 39.4
Percentage with Bachelors degrees: 16.4%
Percentage with higher education that earn >50K: 46.5%
Percentage without higher education that earn >50K: 17.4%
Min work time: 1 hours/week
Percentage of rich among those who work fewest hours: 10.0%
Country with highest percentage of rich: Index(['Iran'], dtype='ob
ject')
Highest percentage of rich people in country: 41.9%
Top occupations in India: Prof-specialty
```

Out[103]:

```
{'race_count': White                 27816
 Black                  3124
 Asian-Pac-Islander     1039
 Amer-Indian-Eskimo      311
 Other                   271
 Name: race, dtype: int64,
 'average_age_men': 39.4,
 'percentage_bachelors': 16.4,
 'higher_education_rich': 46.5,
 'lower_education_rich': 17.4,
 'min_work_hours': 1,
 'rich_percentage': 10.0,
 'highest_earning_country': Index(['Iran'], dtype='object'),
 'highest_earning_country_percentage': 41.9,
 'top_IN_occupation': 'Prof-specialty'}
```

In [89]:

```
abc = df_rich['native-country'].value_counts()
```

In [91]:

```
cab = df['native-country'].value_counts()
```

In [97]:

```
percent = round((abc/cab*100),1)
```

Out[97]:

```
25.0
```

In [101]:

```
percent.max()
```

Out[101]:

```
41.9
```