task_20

November 20, 2023

1 Data Structure Analysis

[1]:
```
import pandas as pd
import numpy as np
```

[2]:
```
url = "https://raw.githubusercontent.com/dsuprunov/
 ↪python-statista-programming-challenge/main/docs/Input_Dataset.csv"
df = pd.read_csv(url)
df.replace('?', np.nan, inplace=True)
```

[3]:
```
df
```

[3]:

	age	workclass	fnlwgt	education	educational-num	\
0	25	Private	226802	11th	7	
1	38	Private	89814	HS-grad	9	
2	28	Local-gov	336951	Assoc-acdm	12	
3	44	Private	160323	Some-college	10	
4	18	NaN	103497	Some-college	10	
…	…	…	…	…	…	
48837	27	Private	257302	Assoc-acdm	12	
48838	40	Private	154374	HS-grad	9	
48839	58	Private	151910	HS-grad	9	
48840	22	Private	201490	HS-grad	9	
48841	52	Self-emp-inc	287927	HS-grad	9	

	marital-status	occupation	relationship	race	gender	\
0	Never-married	Machine-op-inspct	Own-child	Black	Male	
1	Married-civ-spouse	Farming-fishing	Husband	White	Male	
2	Married-civ-spouse	Protective-serv	Husband	White	Male	
3	Married-civ-spouse	Machine-op-inspct	Husband	Black	Male	
4	Never-married	NaN	Own-child	White	Female	
…	…	…	…	…	…	
48837	Married-civ-spouse	Tech-support	Wife	White	Female	
48838	Married-civ-spouse	Machine-op-inspct	Husband	White	Male	
48839	Widowed	Adm-clerical	Unmarried	White	Female	
48840	Never-married	Adm-clerical	Own-child	White	Male	
48841	Married-civ-spouse	Exec-managerial	Wife	White	Female	

```
       capital-gain  capital-loss  hours-per-week native-country income
0                 0             0              40  United-States  <=50K
1                 0             0              50  United-States  <=50K
2                 0             0              40  United-States   >50K
3              7688             0              40  United-States   >50K
4                 0             0              30  United-States  <=50K
…               …           …             …            …    …
48837             0             0              38  United-States  <=50K
48838             0             0              40  United-States   >50K
48839             0             0              40  United-States  <=50K
48840             0             0              20  United-States  <=50K
48841         15024             0              40  United-States   >50K

[48842 rows x 15 columns]
```

2 Data structure conclusions

- The provided data is census data
- The data consists of 48842 rows
- Each row represents a unit of people in the target population that this row represents

Key	Type	NaN	nunique	Descriptiom
age	int	No	74	age
workclass	string	Yes	8	employment status
fnlwgt	int	No	28523	the number of units in the target population that the responding unit represents
education	string	No	16	education degree
educational-num	int	No	16	number of years of education in total
marital-status	string	No	7	marital status
occupation	string	Yes	14	occupation

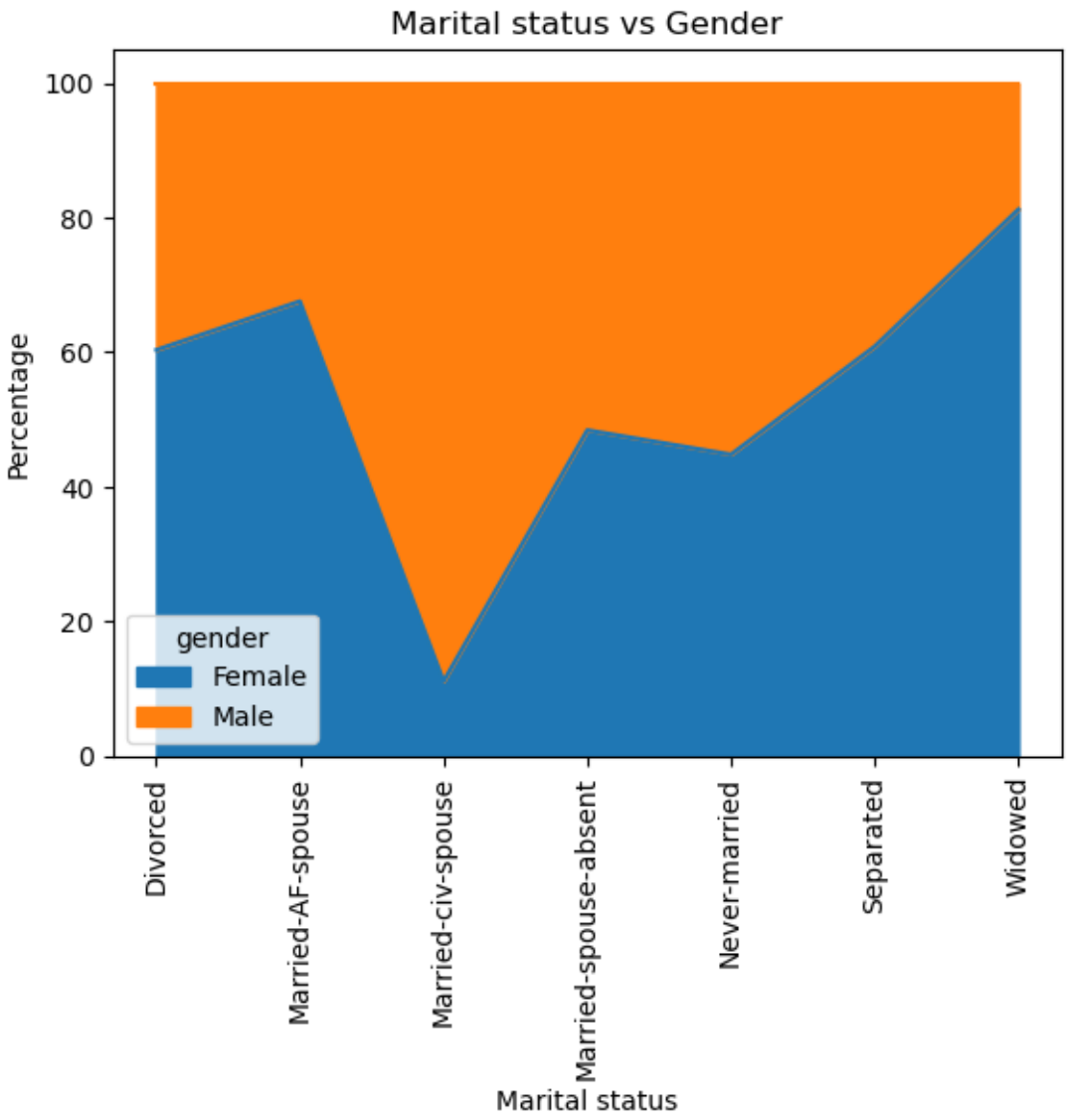
Key	Type	NaN	nunique	Descriptiom
relationship	string	No	6	represents the respondent's role in the family
race	string	No	5	race
gender	string	No	2	gender
capital-gain	int	No	123	income gain from investment sources other than wage/salary
capital-loss	int	No	99	income loss from investment sources other than wage/salary
hours-per-week	int	No	96	the hours to work per week
native-country	string	Yes	41	country of origin
income	string	No	2	annual income

2.1 Insight #1: In marriage, husbands tend to die earlier than wives.

- **Importance**: This insight is crucial for understanding the gender distribution among widowed individuals. The observed gender imbalance can be important for social support systems and healthcare services tailored to the specific needs of widowed individuals.
- **Discovery**: From the graph below, we observe that for the **Widowed** group, there are 81% females and only 19% males.

[4]:
```
marital_vs_gender = df.groupby(by=['marital-status', 'gender'], dropna=True).
↪size().unstack()
marital_vs_gender = marital_vs_gender.div(marital_vs_gender.
↪sum(axis='columns'), axis='index') * 100
ax = marital_vs_gender.plot(kind='area', title='Marital status vs Gender',␣
↪rot=90)
ax.set_xlabel('Marital status')
ax.set_ylabel('Percentage')
```

[4]: Text(0, 0.5, 'Percentage')



[5]:
```
# marital_vs_gender.round(0).astype(int).astype('string') + ' %'
```

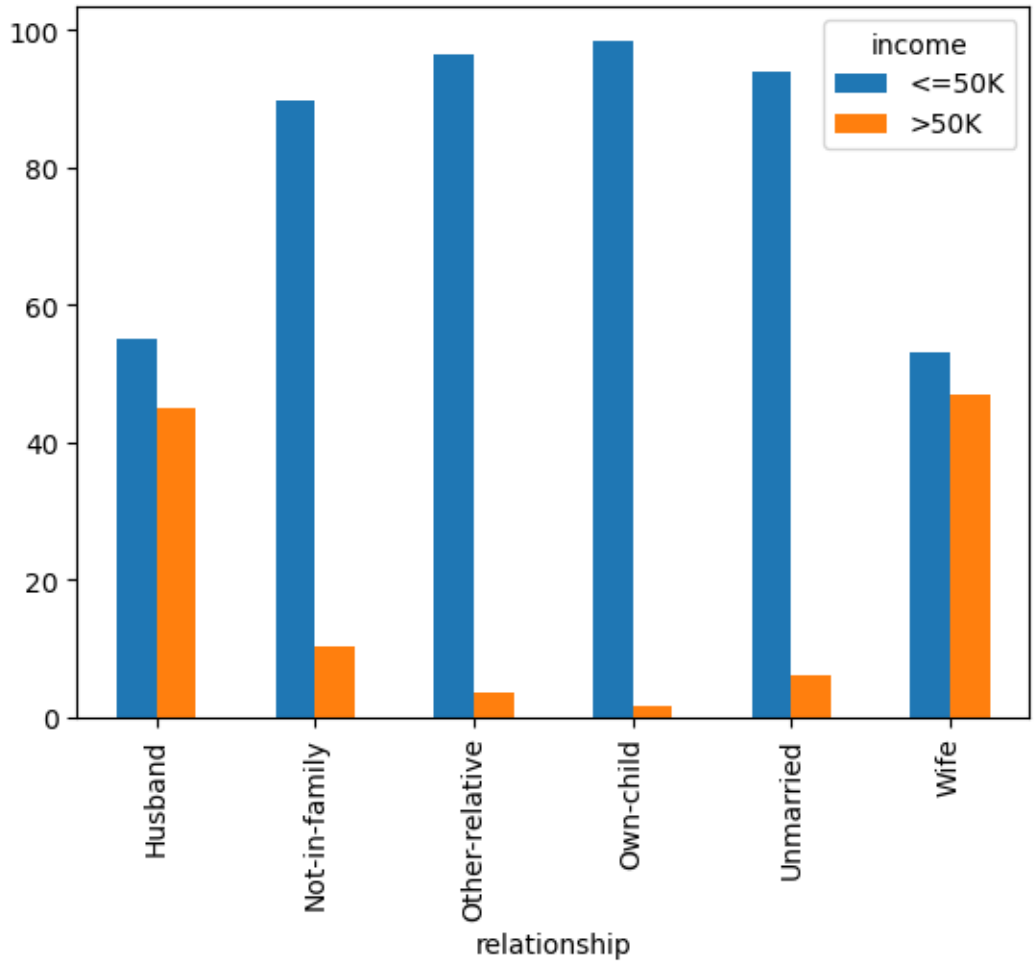
2.2 Insight #2: In the family, the incomes of both partners are approximately at the same level.

- **Importance**: Understanding that incomes of both partners in a family are generally at the same level is important for addressing issues of financial equality and joint decision-making within households.

- **Discovery**: In the graph below, we are interested in only two parameters: wife and husband. All other options are not relevant in relation to the family.

[6]:
```
income_by_relationship = df.groupby('relationship')['income'].
↪value_counts(normalize=True).unstack() * 100
income_by_relationship.plot(kind='bar')
```

[6]: <Axes: xlabel='relationship'>

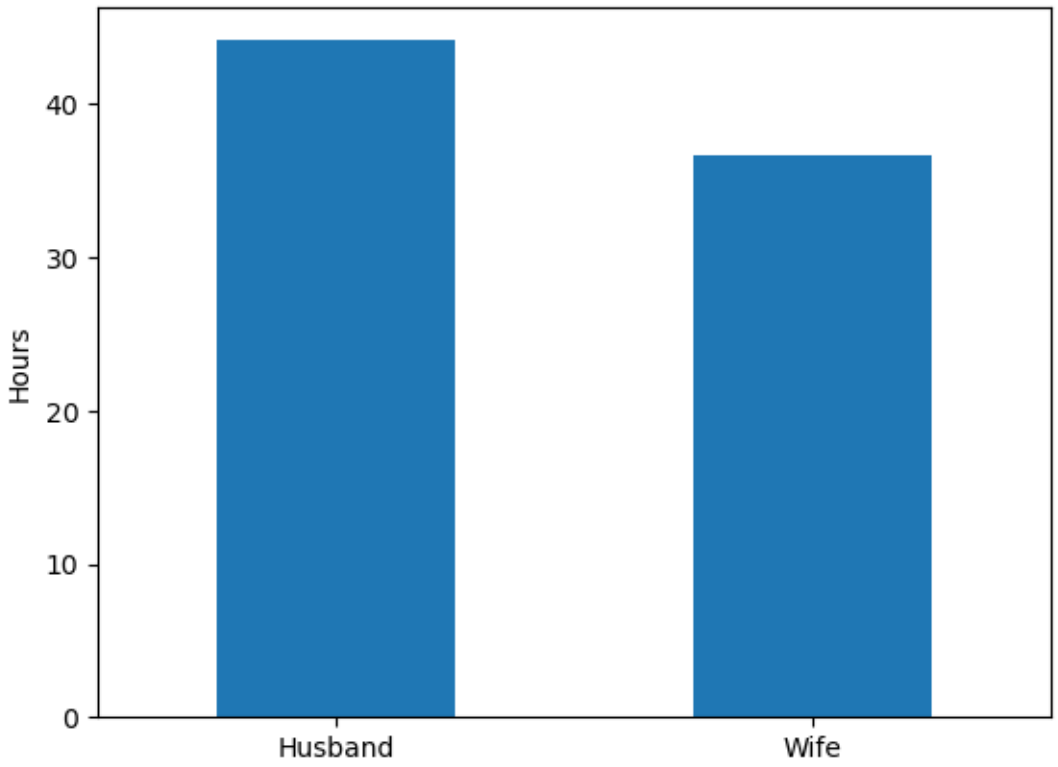


2.3 Insight #3: In the family the husband works more.

- **Importance**: Recognizing that husbands tend to work more hours sheds light on gender roles and responsibilities within households, contributing to discussions on work-life balance.
- **Discovery**: In the graph below, we can see that husbands have a higher number of hours per week than wives.

[7]:
```
family = df[(df['relationship'] == 'Wife') | (df['relationship'] == 'Husband')]
average_hours_by_relationship = family.
 ↪groupby('relationship')['hours-per-week'].mean()
ax = average_hours_by_relationship.plot(kind='bar', rot=0)
ax.set_xlabel('')
ax.set_ylabel('Hours')
```

[7]:
```
Text(0, 0.5, 'Hours')
```



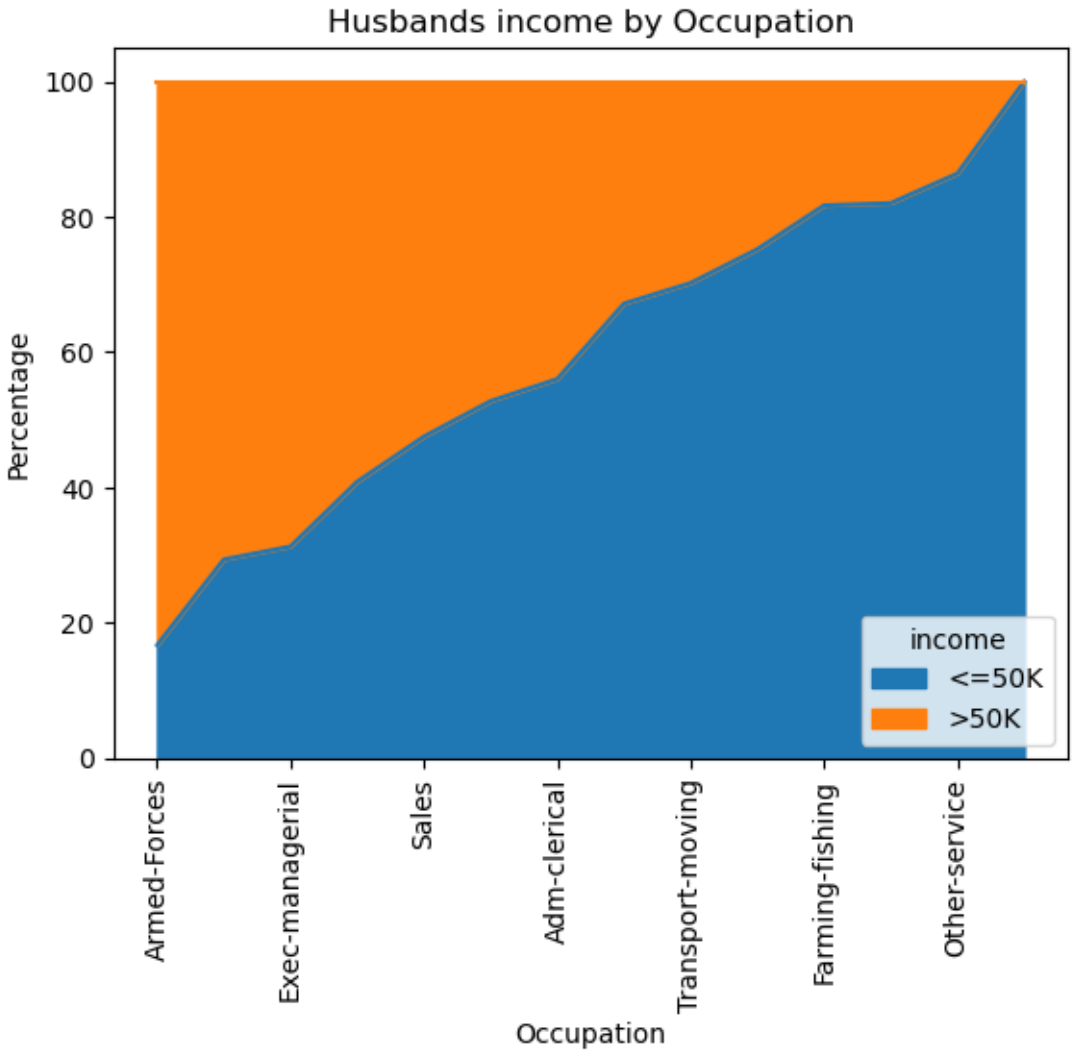
2.4 Insight #4: The highest-paying jobs for husbands.

- **Importance**: Identifying the highest-paying jobs for husbands is significant for understanding ways to meet the financial needs of the family.
- **Discovery**: In the graph below, we can see that for husbands, the highest-paid professions are government jobs or executive-managerial positions in business.

[8]:
```
husbands = df[df['relationship'] == 'Husband']
income_by_occupation_husbands = husbands.groupby('occupation')['income'].
 ↪value_counts(normalize=True).unstack() * 100
income_by_occupation_husbands = income_by_occupation_husbands.
 ↪sort_values(by='>50K', ascending=False)
```

```
ax = income_by_occupation_husbands.plot(kind='area', title = 'Husbands income␣
 ↪by Occupation', rot=90)
ax.set_xlabel('Occupation')
ax.set_ylabel('Percentage')
```

[8]: Text(0, 0.5, 'Percentage')

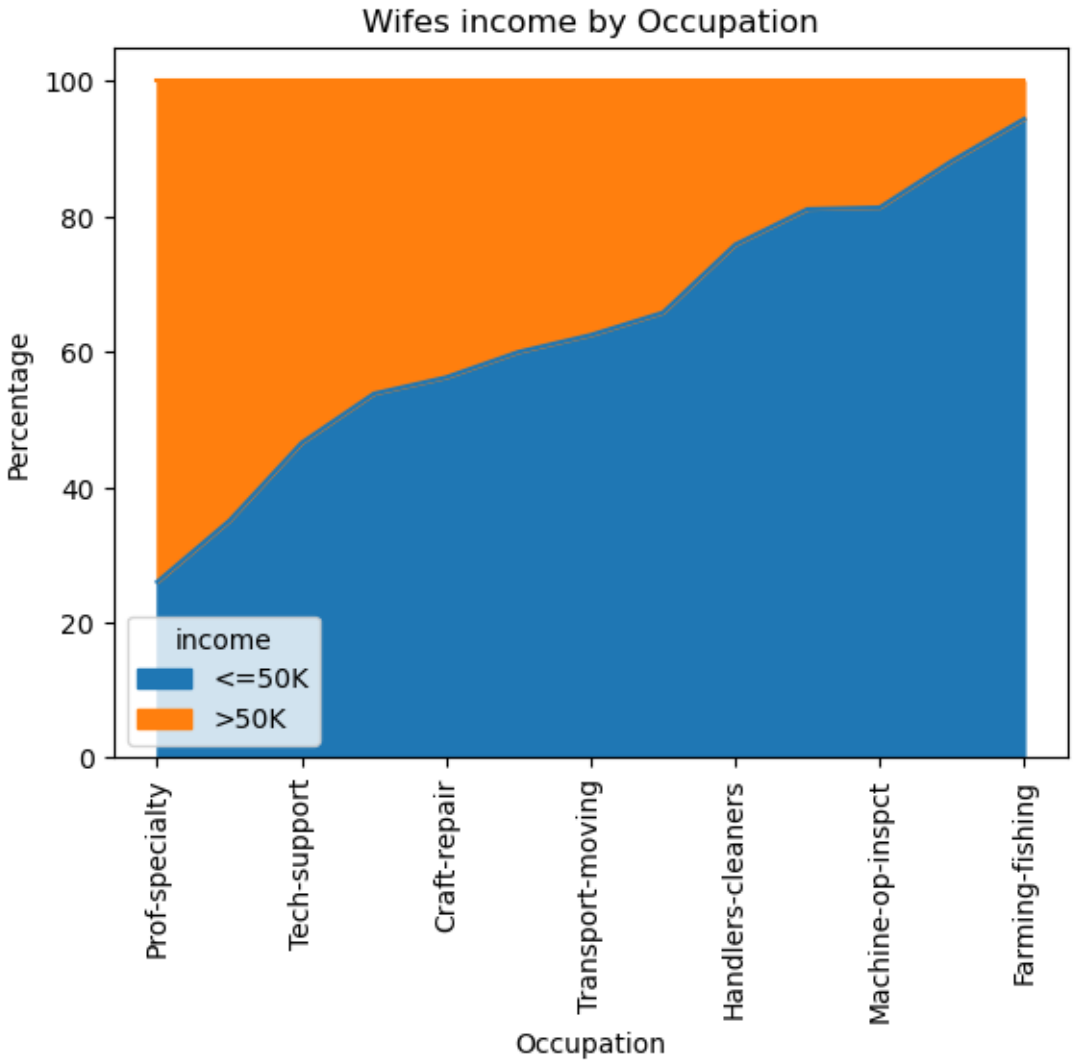


2.5 Insight #5: The highest-paying jobs for wives.

- **Importance**: Knowing the highest-paying jobs for wives contributes to discussions on gender equality in the workforce, supporting efforts to break down barriers and promote equal opportunities for all.
- **Discovery**: In the graph below, we can see that for wives, the highest-paid professions are in professional specializations and support roles (business and technical support).

[9]:
```
wifes = df[df['relationship'] == 'Wife']
income_by_occupation_wifes = wifes.groupby('occupation')['income'].
 ↪value_counts(normalize=True).unstack() * 100
income_by_occupation_wifes = income_by_occupation_wifes.sort_values(by='>50K',␣
 ↪ascending=False)
ax = income_by_occupation_wifes.plot(kind='area', title = 'Wifes income by␣
 ↪Occupation', rot=90)
ax.set_xlabel('Occupation')
ax.set_ylabel('Percentage')
```

[9]:
```
Text(0, 0.5, 'Percentage')
```



2.6 Insight #6: Invest in your education for a higher income.

- **Importance**: Understanding the correlation between education and income is crucial for individuals making educational and career choices and establishing a good and solid family structure.
- **Discovery**: In the graph below, we can see that relationship between ***income*** and ***educational-num***, when ***educational-num*** surpasses 12 years, there's a clear decrease in respondents with income <=50K and an increase in those with income >50K. According to the initial data, "12+ years of education" corresponds to higher education or profession training.

[10]:
```
income_by_education = df.groupby(by='educational-num', dropna=True)['income'].
↪value_counts(normalize=True).unstack()*100
ax = income_by_education.plot(kind='area', title='Income vs Number of␣
↪educational years', rot=0)
ax.set_xlabel('Number of years of education in total')
ax.set_ylabel('Percentage')
```

[10]:
```
Text(0, 0.5, 'Percentage')
```