data_analysis

December 7, 2024

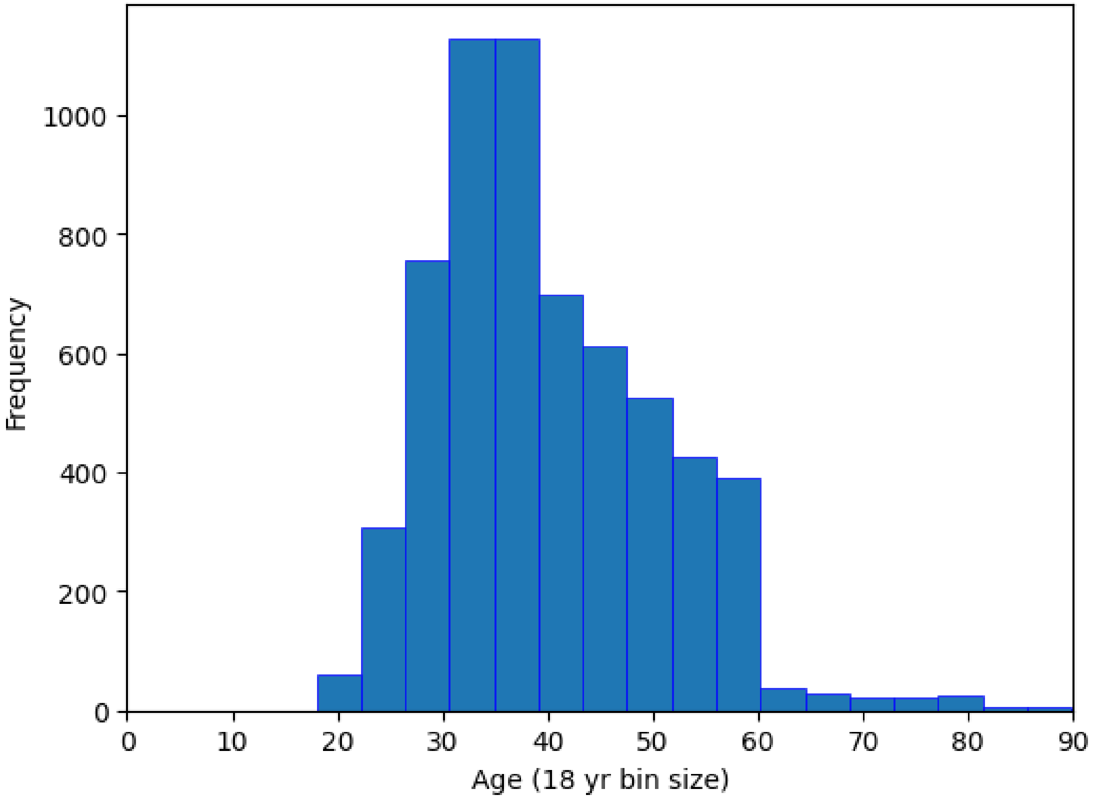
Author: Jase Banta
Data: 2024-12-07
Description: Sample python code written for UChicago M.S. Applied Data Science Program.

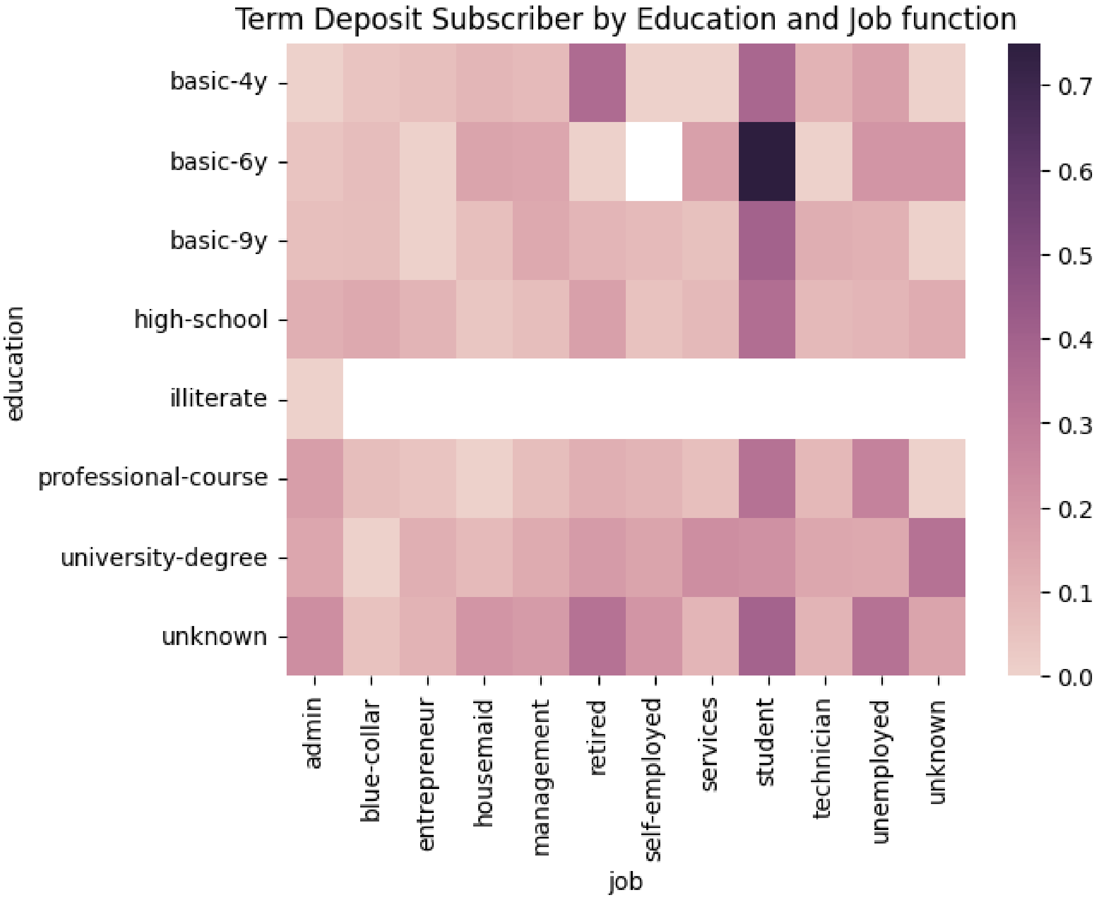
[5]:

```
import csv, random, argparse
import pandas as pd
import numpy as np
import seaborn, matplotlib.pyplot as plt


class Utility:
    def __init__(self):
        pass

    def random_select(self, source: str, target: str, size: float) -> str:
        with open(source, 'r') as file:
            reader = csv.reader(file, delimiter= ';')
            header = next(reader)
            data = list(reader)
            size = round(size * len(data))
            sample_rows = random.sample(data, min(size, len(data)))

        with open(target, 'w', newline= '') as outfile:
            writer = csv.writer(outfile, delimiter= ',')
            writer.writerow(header)
            writer.writerows(sample_rows)

        return target

    def summarize(self, value):
        pass

    def transformBinary(self, df, columns: list):
        mapping = {
            'yes': True,
            'no': False,
            'unknown': np.nan
        }
        for column in columns:
            df[column] = df[column].replace(mapping)

        return df

def fixCols(util, df):
    df['job'] = df['job'].apply(lambda val: val.rstrip(".,!?"))
    df["education"] = df["education"].apply(lambda val: val.replace('.', '-'))
    return df

def showHistogram(col):
    fig, ax = plt.subplots()
    ax.hist(col, bins = 18, linewidth = 0.5, edgecolor = 'blue')
    ax.set(xlim=(0, 90), xlabel= "Age (18 yr bin size)", ylabel= "Frequency")
    plt.show()

def main():
    util = Utility()
    sample_data = util.random_select(source = 'bank-additional-full.csv',␣
  ↪target="bank-sample.csv", size = 0.15)
    sample_data = pd.read_csv(sample_data)
    new_data = util.transformBinary(df = sample_data, columns = ['default',␣
  ↪'housing','loan','y'])
    new_data = fixCols(util, new_data)
    print(new_data[['age', 'duration']].describe())
    showHistogram(new_data['age'])
    pivot_table = new_data.pivot_table(
        values='y',
        index=['education'],
        columns=['job'],
        aggfunc='mean',
        fill_value=np.nan
    )
    # print(pivot_table)
    seaborn.heatmap(pivot_table, cmap=seaborn.cubehelix_palette(as_cmap=True))
    plt.title("Term Deposit Subscriber by Education and Job function")
    plt.show()

main()
```

```
               age     duration
count  6178.000000  6178.000000
mean     40.002266   258.878601
std      10.507797   260.536947
min      18.000000     0.000000
25%      32.000000   102.000000
50%      38.000000   178.000000
```

```
75%       47.000000   319.000000
max       94.000000  3366.000000
```




Term Deposit Subscriber by Education and Job function
education
basic-4y
basic-6y
basic-9y
high-school
illiterate
professional-course
university-degree
unknown
job
admin
blue-collar
entrepreneur
housemaid
management
retired
self-employed
services
student
technician
unemployed
unknown
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0