lab-assignment8

August 6, 2024

1 Lab Assignment 8: Data Management Using pandas, Part 1

1.1 DS 6001: Practice and Application of Data Science

1.1.1 Instructions

Please answer the following questions as completely as possible using text, code, and the results of code as needed. Format your answers in a Jupyter notebook. To receive full credit, make sure you address every part of the problem, and make sure your document is formatted in a clean and professional way.

In this lab, you will be working with the 2017 Workplace Health in America survey which was conducted by the Centers for Disease Control and Prevention. According to the survey's guidence document:

> The Workplace Health in America (WHA) Survey gathered information from a cross-sectional, nationally representative sample of US worksites. The sample was drawn from the Dun & Bradstreet (D&B) database of all private and public employers in the United States with at least 10 employees. Like previous national surveys, the worksite served as the sampling unit rather than the companies or firms to which the worksites belonged. Worksites were selected using a stratified simple random sample (SRS) design, where the primary strata were ten multi-state regions defined by the Centers for Disease Control and Prevention (CDC), plus an additional stratum containing all hospital worksites.

The data contain over 300 features that report the industry and type of company where the respondents are employed, what kind of health insurance and other health programs are offered, and other characteristics of the workplaces including whether employees are allowed to work from home and the gender and age makeup of the workforce. The data are full of interesting information, but in order to make use of the data a great deal of data manipulation is required first.

1.2 Problem 0

Import the following libraries:

[ ]:

```
import numpy as np
import pandas as pd
import sidetable
import sqlite3
import warnings
import requests
import io
```

```
from io import StringIO
warnings.filterwarnings('ignore')
```

1.3 Problem 1

The raw data are stored in an ASCII file on the 2017 Workplace Health in America survey homepage. Load the raw data directly into Python without downloading the data onto your harddrive and display a dataframe with only the 14th, 28th, and 102nd rows of the data. [1 point]

```
[ ]: url = 'https://www.cdc.gov/workplacehealthpromotion/data-surveillance/docs/
       ↪whpps_120717.csv'
     response = requests.get(url)
     data = response.text

     df_io = io.StringIO(data)
     data_df = pd.read_csv(df_io, delimiter='~')
     selected_rows = data_df.iloc[[13, 27, 101]]
     print(selected_rows)
```

```
    OC1  OC3  HI1  HI2  HI3  HI4  HRA1  HRA1A  HRA1B  HRA1E  …  WL3_05  \
13    3  1.0  2.0  3.0  2.0  1.0   1.0    3.0    3.0    1.0  …     NaN
27    1  3.0  1.0  3.0  1.0  1.0   1.0    2.0    4.0    2.0  …     NaN
101   2  1.0  1.0  3.0  2.0  1.0   1.0    2.0    4.0    2.0  …     NaN

     E1_09  Suppquex       Id  Region  CDC_Region  Industry  Size  Varstrata  \
13     NaN       1.0   1437.0     4.0         6.0       7.0   3.0        0.0
27     NaN       1.0   2501.0     2.0         4.0       7.0   8.0        0.0
101    NaN       2.0  12636.0     4.0         6.0       7.0   4.0        0.0

     Finalwt_worksite,,,,
13       47.793940929,,,,
27       47.793940929,,,,
101      47.793940929,,,,

[3 rows x 301 columns]
```

1.4 Problem 2

The data contain 301 columns. Create a new variable in Python's memory to store a working version of the data. In the working version, delete all of the columns except for the following:

- `Industry`: 7 Industry Categories with NAICS codes
- `Size`: 8 Employee Size Categories
- `OC3` Is your organization for profit, non-profit, government?
- `HI1` In general, do you offer full, partial or no payment of premiums for personal health insurance for full-time employees?

- `HI2` Over the past 12 months, were full-time employees asked to pay a larger proportion, smaller proportion or the same proportion of personal health insurance premiums?
- `HI3`: Does your organization offer personal health insurance for your part-time employees?
- `CP1`: Are there health education programs, which focus on skill development and lifestyle behavior change along with information dissemination and awareness building?
- `WL6`: Allow employees to work from home?
- Every column that begins `WD`, expressing the percentage of employees that have certain characteristics at the firm

[1 point]

[ ]:

```
column_keep = [
    'Industry', 'Size', 'OC3', 'HI1', 'HI2', 'HI3', 'CP1', 'WL6'
]

column_keep.extend([col for col in data_df.columns if col.startswith('WD')])

wk_data = data_df[column_keep]
print(wk_data.head())
```

```
   Industry  Size  OC3  HI1  HI2  HI3  CP1  WL6  WD1_1  WD1_2    WD2    WD3  \
0       7.0   7.0  3.0  2.0  1.0  2.0  1.0  1.0   25.0   20.0   85.0   60.0
1       7.0   6.0  3.0  2.0  3.0  1.0  1.0  1.0  997.0  997.0   90.0   90.0
2       7.0   8.0  3.0  1.0  3.0  1.0  1.0  1.0   35.0    4.0  997.0  997.0
3       7.0   4.0  2.0  1.0  2.0  1.0  2.0  2.0   50.0   15.0   50.0   85.0
4       7.0   4.0  3.0  1.0  3.0  1.0  1.0  1.0   50.0   40.0   60.0   60.0

     WD4    WD5    WD6    WD7
0   40.0   15.0    0.0   22.0
1  997.0  997.0    0.0  997.0
2   40.0   15.0  997.0  997.0
3   75.0    0.0    0.0  997.0
4   40.0   30.0    0.0   28.0
```

1.5 Problem 3

The codebook for the WHA data contain short descriptions of the meaning of each of the columns in the data. Use these descriptions to decide on better and more intuitive names for the columns in the working version of the data, and rename the columns accordingly. [1 point]

[ ]:

```
column_rename = {
    'OC3': 'Organization_Type',
    'HI1': 'Insurance_Type',
    'HI2': 'Insurance_Payment',
    'HI3': 'Part_Time_Insurance',
    'CP1': 'Health_Education_Programs',
    'WL6': 'Work_From_Home',
    'WD1_1': 'Percentage_Under_30',
    'WD1_2': 'Percentage_60_and_Above',
    'WD2': 'Percentage_Female_Workers',
    'WD3': 'Percentage_Hourly_Workers',
    'WD4': 'Percentage_Non_Daytime_Workers',
    'WD5': 'Percentage_Remote_Workers',
    'WD6': 'Percentage_Union_Workers',
    'WD7': 'Annual_Employee_Turnover'
}
wk_data.rename(columns=column_rename, inplace=True)
print(wk_data.head())
```

```
   Industry  Size  Organization_Type  Insurance_Type  Insurance_Payment  \
0       7.0   7.0                3.0             2.0                1.0
1       7.0   6.0                3.0             2.0                3.0
2       7.0   8.0                3.0             1.0                3.0
3       7.0   4.0                2.0             1.0                2.0
4       7.0   4.0                3.0             1.0                3.0

   Part_Time_Insurance  Health_Education_Programs  Work_From_Home  \
0                  2.0                        1.0             1.0
1                  1.0                        1.0             1.0
2                  1.0                        1.0             1.0
3                  1.0                        2.0             2.0
4                  1.0                        1.0             1.0

   Percentage_Under_30  Percentage_60_and_Above  Percentage_Female_Workers  \
0                 25.0                     20.0                       85.0
1                997.0                    997.0                       90.0
2                 35.0                      4.0                      997.0
3                 50.0                     15.0                       50.0
4                 50.0                     40.0                       60.0

   Percentage_Hourly_Workers  Percentage_Non_Daytime_Workers  \
0                       60.0                            40.0
1                       90.0                           997.0
2                      997.0                            40.0
3                       85.0                            75.0
4                       60.0                            40.0

   Percentage_Remote_Workers  Percentage_Union_Workers  \
0                       15.0                       0.0
1                      997.0                       0.0
2                       15.0                     997.0
3                        0.0                       0.0
4                       30.0                       0.0

   Annual_Employee_Turnover
```

```
0                       22.0
1                      997.0
2                      997.0
3                      997.0
4                       28.0
```

1.6 Problem 4

Using the codebook and this dictionary of NAICS industrial codes, place descriptive labels on the categories of the industry column in the working data. [1 point]

[ ]:

```python
naics_map = {
    1: 'Agriculture, Forestry, Fishing and Hunting; Mining; Utilities; Construction; Manufacturing',
    2: 'Wholesale Trade; Retail Trade; Transportation and Warehousing',
    3: 'Arts, Entertainment, and Recreation; Accommodation and Food Services; Other Services',
    4: 'Information; Finance and Insurance; Real Estate Rental and Leasing; Professional, Scientific, and Technical Services; Management of Companies and Enterprises; Administrative and Support and Waste Management Services',
    5: 'Educational Services; Health Care and Social Assistance',
    6: 'Public Administration',
    7: 'General Medical and Surgical Hospitals; Psychiatric and Substance Abuse Hospitals; Specialty Hospitals',
    np.nan: np.nan
}

wk_data['Industry'] = wk_data['Industry'].map(naics_map)
print(wk_data.head())
```

```
                                            Industry  Size  Organization_Type  \
0  General Medical and Surgical Hospitals; Psychi…   7.0                3.0
1  General Medical and Surgical Hospitals; Psychi…   6.0                3.0
2  General Medical and Surgical Hospitals; Psychi…   8.0                3.0
3  General Medical and Surgical Hospitals; Psychi…   4.0                2.0
4  General Medical and Surgical Hospitals; Psychi…   4.0                3.0

   Insurance_Type  Insurance_Payment  Part_Time_Insurance  \
0             2.0                1.0                  2.0
1             2.0                3.0                  1.0
2             1.0                3.0                  1.0
3             1.0                2.0                  1.0
4             1.0                3.0                  1.0

   Health_Education_Programs  Work_From_Home  Percentage_Under_30  \
0                        1.0             1.0                 25.0
1                        1.0             1.0                997.0
2                        1.0             1.0                 35.0
```

```
3                        2.0              2.0                 50.0
4                        1.0              1.0                 50.0

   Percentage_60_and_Above  Percentage_Female_Workers  \
0                     20.0                       85.0
1                    997.0                       90.0
2                      4.0                      997.0
3                     15.0                       50.0
4                     40.0                       60.0

   Percentage_Hourly_Workers  Percentage_Non_Daytime_Workers  \
0                       60.0                            40.0
1                       90.0                           997.0
2                      997.0                            40.0
3                       85.0                            75.0
4                       60.0                            40.0

   Percentage_Remote_Workers  Percentage_Union_Workers  \
0                       15.0                       0.0
1                      997.0                       0.0
2                       15.0                     997.0
3                        0.0                       0.0
4                       30.0                       0.0

   Annual_Employee_Turnover
0                      22.0
1                     997.0
2                     997.0
3                     997.0
4                      28.0
```

1.7 Problem 5

Using the codebook, recode the “size” column to have three categories: “Small” for workplaces with fewer than 100 employees, “Medium” for workplaces with at least 100 but fewer than 500 employees, and “Large” for companies with at least 500 employees. [Note: Python dataframes have an attribute .size that reports the space the dataframe takes up in memory. Don’t confuse this attribute with the column named “Size” in the raw data.] [1 point]

[ ]:
```
def recode_size(size):
    if size in [1, 2, 3]:
        return 'Small'
    elif size in [4, 5]:
        return 'Medium'
    elif size in [6, 7, 8]:
        return 'Large'
    else:
        return np.nan
```

```python
# Apply recoding to the Size column
wk_data['Size'] = wk_data['Size'].apply(recode_size)

print(wk_data.head())
```

```
                                            Industry    Size  \
0  General Medical and Surgical Hospitals; Psychi…   Large   
1  General Medical and Surgical Hospitals; Psychi…   Large   
2  General Medical and Surgical Hospitals; Psychi…   Large   
3  General Medical and Surgical Hospitals; Psychi…  Medium   
4  General Medical and Surgical Hospitals; Psychi…  Medium   

   Organization_Type  Insurance_Type  Insurance_Payment  Part_Time_Insurance  \
0                3.0             2.0                1.0                  2.0   
1                3.0             2.0                3.0                  1.0   
2                3.0             1.0                3.0                  1.0   
3                2.0             1.0                2.0                  1.0   
4                3.0             1.0                3.0                  1.0   

   Health_Education_Programs  Work_From_Home  Percentage_Under_30  \
0                        1.0             1.0                 25.0   
1                        1.0             1.0                997.0   
2                        1.0             1.0                 35.0   
3                        2.0             2.0                 50.0   
4                        1.0             1.0                 50.0   

   Percentage_60_and_Above  Percentage_Female_Workers  \
0                     20.0                       85.0   
1                    997.0                       90.0   
2                      4.0                      997.0   
3                     15.0                       50.0   
4                     40.0                       60.0   

   Percentage_Hourly_Workers  Percentage_Non_Daytime_Workers  \
0                       60.0                            40.0   
1                       90.0                           997.0   
2                      997.0                            40.0   
3                       85.0                            75.0   
4                       60.0                            40.0   

   Percentage_Remote_Workers  Percentage_Union_Workers  \
0                       15.0                       0.0   
1                      997.0                       0.0   
2                       15.0                     997.0   
3                        0.0                       0.0   
4                       30.0                       0.0   
```

```
   Annual_Employee_Turnover
0                      22.0
1                     997.0
2                     997.0
3                     997.0
4                      28.0
```

1.8 Problem 6

Use the codebook to write accurate and descriptive labels for each category for each categorical column in the working data. Then apply all of these labels to the data at once. Code “Legitimate Skip”, “Don’t know”, “Refused”, and “Blank” as missing values. [2 points]

[ ]:

```
column_rename = {
    'Industry': 'Industry_Sectors',
    'OC3': 'Organization_Type',
    'HI1': 'Insurance_Type',
    'HI2': 'Insurance_Payment',
    'HI3': 'Part_Time_Insurance',
    'CP1': 'Health_Education_Programs',
    'WL6': 'Work_From_Home',
    'WD1_1': 'Percentage_Under_30',
    'WD1_2': 'Percentage_60_and_Above',
    'WD2': 'Percentage_Female_Workers',
    'WD3': 'Percentage_Hourly_Workers',
    'WD4': 'Percentage_Non_Daytime_Workers',
    'WD5': 'Percentage_Remote_Workers',
    'WD6': 'Percentage_Union_Workers',
    'WD7': 'Annual_Employee_Turnover'
}
wk_data.rename(columns=column_rename, inplace=True)
```

[ ]:

```
categorical_mapping = {
    'Organization_Type': {
        1: 'For profit, public',
        2: 'For profit, private',
        3: 'Non-profit',
        4: 'State or local government',
        5: 'Federal government',
        6: 'Other'
    },
    'Insurance_Type': {
        1: 'Full insurance coverage offered',
        2: 'Partial insurance coverage offered',
        3: 'No insurance coverage offered'
    },
    'Insurance_Payment': {
        1: 'Larger',
        2: 'Smaller',
        3: 'About the same'
    },
    'Part_Time_Insurance': {
        1: 'Yes',
        2: 'No'
    },
    'Health_Education_Programs': {
        1: 'Yes',
        2: 'No'
    },
    'Work_From_Home': {
        1: 'Yes',
        2: 'No'
    }
}

for column, mapping in categorical_mapping.items():
    wk_data[column] = wk_data[column].map(mapping)

wk_data.replace(['Legitimate Skip', "Don't know", 'Refused', 'Blank'], np.nan,␣
 ↪inplace=True)

print(wk_data.head())
```

```
                                    Industry_Sectors    Size  \
0  General Medical and Surgical Hospitals; Psychi…   Large
1  General Medical and Surgical Hospitals; Psychi…   Large
2  General Medical and Surgical Hospitals; Psychi…   Large
3  General Medical and Surgical Hospitals; Psychi…  Medium
4  General Medical and Surgical Hospitals; Psychi…  Medium

     Organization_Type                      Insurance_Type Insurance_Payment  \
0           Non-profit  Partial insurance coverage offered            Larger
1           Non-profit  Partial insurance coverage offered    About the same
2           Non-profit     Full insurance coverage offered    About the same
3  For profit, private     Full insurance coverage offered           Smaller
4           Non-profit     Full insurance coverage offered    About the same

  Part_Time_Insurance Health_Education_Programs Work_From_Home  \
0                  No                       Yes            Yes
1                 Yes                       Yes            Yes
2                 Yes                       Yes            Yes
3                 Yes                        No             No
4                 Yes                       Yes            Yes

   Percentage_Under_30  Percentage_60_and_Above  Percentage_Female_Workers  \
0                 25.0                     20.0                       85.0
```

```
1                 997.0                      997.0                       90.0
2                  35.0                        4.0                      997.0
3                  50.0                       15.0                       50.0
4                  50.0                       40.0                       60.0

   Percentage_Hourly_Workers  Percentage_Non_Daytime_Workers  \
0                       60.0                            40.0
1                       90.0                           997.0
2                      997.0                            40.0
3                       85.0                            75.0
4                       60.0                            40.0

   Percentage_Remote_Workers  Percentage_Union_Workers  \
0                       15.0                       0.0
1                      997.0                       0.0
2                       15.0                     997.0
3                        0.0                       0.0
4                       30.0                       0.0

   Annual_Employee_Turnover
0                      22.0
1                     997.0
2                     997.0
3                     997.0
4                      28.0
```

1.9 Problem 7

The features that measure the percent of the workforce with a particular characteristic use the codes 997, 998, and 999 to represent "Don't know", "Refusal", and "Blank/Invalid" respectively. Replace these values with missing values for all of the percentage features at the same time. [1 point]

[ ]:

```python
pct_features = [col for col in wk_data.columns if col.startswith('Percentage') or col.startswith('Annual')]


wk_data[pct_features] = wk_data[pct_features].replace([997, 998, 999], np.nan)
```

1.10 Problem 8

Sort the working data by industry in ascending alphabetical order. Within industry categories, sort the rows by size in ascending alphabetical order. Within groups with the same industry and size, sort by percent of the workforce that is under 30 in descending numeric order. [1 point]

[ ]:

```python
wk_data.sort_values(by=['Industry_Sectors', 'Size', 'Percentage_Under_30'], ascending=[True, True, False], inplace=True)
```

1.11 Problem 9

There is one row in the working data that has a NaN value for industry. Delete this row. Use a logical expression, and not the row number. [1 point]

```
[ ]: wk_data = wk_data[wk_data['Industry_Sectors'].notna()]
     print(wk_data.head())
```

```
                                       Industry_Sectors   Size  \
1732  Agriculture, Forestry, Fishing and Hunting; Mi…  Large   
1476  Agriculture, Forestry, Fishing and Hunting; Mi…  Large   
1477  Agriculture, Forestry, Fishing and Hunting; Mi…  Large   
704   Agriculture, Forestry, Fishing and Hunting; Mi…  Large   
1241  Agriculture, Forestry, Fishing and Hunting; Mi…  Large   

        Organization_Type                      Insurance_Type  \
1732  For profit, private  Partial insurance coverage offered   
1476  For profit, private  Partial insurance coverage offered   
1477  For profit, private  Partial insurance coverage offered   
704   For profit, private     Full insurance coverage offered   
1241  For profit, private     Full insurance coverage offered   

     Insurance_Payment Part_Time_Insurance Health_Education_Programs  \
1732    About the same                  No                       Yes   
1476    About the same                  No                       Yes   
1477           Smaller                  No                       Yes   
704     About the same                  No                       Yes   
1241    About the same                  No                       Yes   

     Work_From_Home  Percentage_Under_30  Percentage_60_and_Above  \
1732             No                 50.0                     10.0   
1476             No                 40.0                     10.0   
1477            Yes                 25.0                     15.0   
704             Yes                 20.0                     15.0   
1241            Yes                 20.0                     25.0   

      Percentage_Female_Workers  Percentage_Hourly_Workers  \
1732                       50.0                       75.0   
1476                       30.0                       60.0   
1477                       20.0                       60.0   
704                        17.0                       62.0   
1241                       50.0                       70.0   

      Percentage_Non_Daytime_Workers  Percentage_Remote_Workers  \
1732                            10.0                        0.0   
1476                            30.0                        5.0   
1477                            10.0                        2.0   
704                             10.0                        5.0   
1241                            20.0                        5.0   
```

```
      Percentage_Union_Workers  Annual_Employee_Turnover
1732                       0.0                      75.0
1476                       0.0                      10.0
1477                      60.0                       5.0
704                        0.0                      11.0
1241                       0.0                       3.0
```

1.12 Problem 10

Create a new feature named **gender_balance** that has three categories: “Mostly men” for workplaces with between 0% and 35% female employees, “Balanced” for workplaces with more than 35% and at most 65% female employees, and “Mostly women” for workplaces with more than 65% female employees. [1 point]

[ ]:

```python
def gender_balance(pct_female):
    if pct_female <= 35:
        return 'Mostly men'
    elif 35 < pct_female <= 65:
        return 'Balanced'
    elif pct_female > 65:
        return 'Mostly women'
    else:
        return np.nan

wk_data['Gender_Balance'] = wk_data['Percentage_Female_Workers'].
 ↪apply(gender_balance)
```

1.13 Problem 11

Change the data type of all categorical features in the working data from “object” to “category”. [1 point]

[ ]:

```python
categorical_features = ['Organization_Type', 'Insurance_Type',␣
 ↪'Insurance_Payment', 'Part_Time_Insurance', 'Health_Education_Programs',␣
 ↪'Work_From_Home', 'Gender_Balance']
for col in categorical_features:
    wk_data[col] = wk_data[col].astype('category')
```

1.14 Problem 12

Filter the data to only those rows that represent small workplaces that allow employees to work from home. Then report how many of these workplaces offer full insurance, partial insurance, and no insurance. Use a function that reports the percent, cumulative count, and cumulative percent in addition to the counts. [1 point]

[ ]:

```
filtered_df = wk_data[(wk_data['Size'] == 'Small') & (wk_data['Work_From_Home']␣
 ↪== 'Yes')]
insurance_counts = filtered_df['Insurance_Type'].value_counts().reset_index()
insurance_counts.columns = ['Insurance_Type', 'Count']
insurance_counts['Percent'] = (insurance_counts['Count'] /␣
 ↪insurance_counts['Count'].sum()) * 100
insurance_counts['Cumulative_Count'] = insurance_counts['Count'].cumsum()
insurance_counts['Cumulative_Percent'] = insurance_counts['Percent'].cumsum()

print("Insurance Type Distribution in Small Workplaces that Allow Work From␣
 ↪Home:")
print(insurance_counts)
```

```
Insurance Type Distribution in Small Workplaces that Allow Work From Home:
                        Insurance_Type  Count    Percent  Cumulative_Count  \
0     Full insurance coverage offered    324  46.285714               324
1  Partial insurance coverage offered    310  44.285714               634
2       No insurance coverage offered     66   9.428571               700

   Cumulative_Percent
0           46.285714
1           90.571429
2          100.000000
```

1.15 Problem 13

Anything that can be done in SQL can be done with pandas. The next several questions ask you to write pandas code to match a given SQL query. But to check that the SQL query and pandas code yield the same result, create a new database wsing the sqlite3 package and input the cleaned WHA data as a table in this database. (See module 6 for a discussion of SQlite in Python.) [1 point]

[ ]:

```
db_path = "C:\\Users\\qaism\\OneDrive - University of␣
 ↪Virginia\\Documents\\GitHub\\MSDS\\ds6001databases\\wha_data.db"
conn = sqlite3.connect(db_path)

wk_data.to_sql('wha_cleaned', conn, if_exists='replace', index=False)
conn.close()

conn = sqlite3.connect(db_path)
query = "SELECT * FROM wha_cleaned LIMIT 5;"
result = pd.read_sql(query, conn)
print(result)
conn.close()
```

```
                                    Industry_Sectors   Size  \
0  Agriculture, Forestry, Fishing and Hunting; Mi…  Large
1  Agriculture, Forestry, Fishing and Hunting; Mi…  Large
```

```
2  Agriculture, Forestry, Fishing and Hunting; Mi…  Large
3  Agriculture, Forestry, Fishing and Hunting; Mi…  Large
4  Agriculture, Forestry, Fishing and Hunting; Mi…  Large

     Organization_Type                      Insurance_Type Insurance_Payment  \
0  For profit, private  Partial insurance coverage offered    About the same
1  For profit, private  Partial insurance coverage offered    About the same
2  For profit, private  Partial insurance coverage offered           Smaller
3  For profit, private     Full insurance coverage offered    About the same
4  For profit, private     Full insurance coverage offered    About the same

  Part_Time_Insurance Health_Education_Programs Work_From_Home  \
0                  No                       Yes             No
1                  No                       Yes             No
2                  No                       Yes            Yes
3                  No                       Yes            Yes
4                  No                       Yes            Yes

   Percentage_Under_30  Percentage_60_and_Above  Percentage_Female_Workers  \
0                 50.0                     10.0                       50.0
1                 40.0                     10.0                       30.0
2                 25.0                     15.0                       20.0
3                 20.0                     15.0                       17.0
4                 20.0                     25.0                       50.0

   Percentage_Hourly_Workers  Percentage_Non_Daytime_Workers  \
0                       75.0                            10.0
1                       60.0                            30.0
2                       60.0                            10.0
3                       62.0                            10.0
4                       70.0                            20.0

   Percentage_Remote_Workers  Percentage_Union_Workers  \
0                        0.0                       0.0
1                        5.0                       0.0
2                        2.0                      60.0
3                        5.0                       0.0
4                        5.0                       0.0

   Annual_Employee_Turnover Gender_Balance
0                      75.0       Balanced
1                      10.0     Mostly men
2                       5.0     Mostly men
3                      11.0     Mostly men
4                       3.0       Balanced
```

1.16 Problem 14

Write pandas code that replicates the output of the following SQL code:

```
SELECT size, type, premiums AS insurance, percent_female FROM whpps
WHERE industry = 'Hospitals' AND premium_change='Smaller'
ORDER BY percent_female DESC;
```

For each of these queries, your feature names might be different from the ones listed in the query, depending on the names you chose in problem 3. [2 points]

```
[ ]: result = wk_data[
         (wk_data['Industry_Sectors'] == 'General Medical and Surgical Hospitals;␣
      ↪Psychiatric and Substance Abuse Hospitals; Specialty Hospitals') &
         (wk_data['Insurance_Payment'] == 'Smaller')
     ][['Size', 'Organization_Type', 'Insurance_Type', 'Percentage_Female_Workers']].
      ↪sort_values(by='Percentage_Female_Workers', ascending=False)


     result.rename(columns={
         'Insurance_Type': 'insurance',
         'Percentage_Female_Workers': 'percent_female'
     }, inplace=True)

     print(result)
```

```
       Size    Organization_Type                           insurance  \
320  Medium           Non-profit     Full insurance coverage offered
187   Large           Non-profit  Partial insurance coverage offered
214   Large           Non-profit  Partial insurance coverage offered
229   Small           Non-profit     Full insurance coverage offered
191  Medium           Non-profit  Partial insurance coverage offered
3    Medium  For profit, private     Full insurance coverage offered
97    Large           Non-profit  Partial insurance coverage offered
75   Medium           Non-profit     Full insurance coverage offered
11   Medium                  NaN  Partial insurance coverage offered
48   Medium           Non-profit  Partial insurance coverage offered
51   Medium           Non-profit     Full insurance coverage offered

     percent_female
320            89.0
187            80.0
214            80.0
229            75.0
191            65.0
3              50.0
97              NaN
75              NaN
11              NaN
```

```
48              NaN
51              NaN
```

1.17 Problem 15

Write pandas code that replicates the output of the following SQL code:

```
SELECT industry,
    AVG(percent_female) as percent_female,
    AVG(percent_under30) as percent_under30,
    AVG(percent_over60) as percent_over60
FROM whpps
GROUP BY industry
ORDER BY percent_female DESC;
```

[2 points]

[ ]:
```
result = wk_data.groupby('Industry_Sectors').agg({
    'Percentage_Female_Workers': 'mean',
    'Percentage_Under_30': 'mean',
    'Percentage_60_and_Above': 'mean'
}).reset_index()

result.rename(columns={
    'Percentage_Female_Workers': 'percent_female',
    'Percentage_Under_30': 'percent_under30',
    'Percentage_60_and_Above': 'percent_over60'
}, inplace=True)

result.sort_values(by='percent_female', ascending=False, inplace=True)

print(result)
```

```
                                    Industry_Sectors  percent_female  \
2  Educational Services; Health Care and Social A…       80.657143   
3  General Medical and Surgical Hospitals; Psychi…       76.427027   
1  Arts, Entertainment, and Recreation; Accommoda…       53.804416   
4  Information; Finance and Insurance; Real Estat…       50.632184   
5                              Public Administration       39.056738   
6  Wholesale Trade; Retail Trade; Transportation …       32.657258   
0  Agriculture, Forestry, Fishing and Hunting; Mi…       20.328605   

   percent_under30  percent_over60
2        25.745665       11.349570
3        27.213793       16.489655
1        38.566343       11.544872
4        23.821752       12.465465
5        21.015625       15.015385
6        29.108696       12.584034
```

```
0        22.257143       10.690355
```

1.18 Problem 16

Write pandas code that replicates the output of the following SQL code:

```
SELECT gender_balance, premiums, COUNT(*)
FROM whpps
GROUP BY gender_balance, premiums
HAVING gender_balance is NOT NULL and premiums is NOT NULL;
```

[2 points]

[ ]:

```
result = wk_data.groupby(['Gender_Balance', 'Insurance_Type']).size().
 ↪reset_index(name='count')
result = result.dropna(subset=['Gender_Balance', 'Insurance_Type'])

print(result)
```

```
  Gender_Balance                      Insurance_Type  count
0       Balanced     Full insurance coverage offered    226
1       Balanced       No insurance coverage offered     77
2       Balanced  Partial insurance coverage offered    271
3     Mostly men     Full insurance coverage offered    301
4     Mostly men       No insurance coverage offered     91
5     Mostly men  Partial insurance coverage offered    332
6   Mostly women     Full insurance coverage offered    267
7   Mostly women       No insurance coverage offered    107
8   Mostly women  Partial insurance coverage offered    333
```