labassignment8bn

March 24, 2025

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:

[1]:

```
import numpy as np
import pandas as pd
import sidetable
import sqlite3
import warnings
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]

[2]:
```
whpps = pd.read_csv('https://www.cdc.gov/workplace-health-promotion/media/files/
 ↪2024/06/whpps_120717.csv', sep='~')
whpps.iloc[[14,28,104],:]
```

[2]:
```
     OC1   OC3   HI1   HI2   HI3  HI4  HRA1  HRA1A  HRA1B  HRA1E  …  WL3_05  \
14     7   2.0   2.0   1.0   2.0  1.0   1.0    3.0    2.0    2.0  …     NaN
28     1   3.0   2.0   3.0   1.0  1.0   2.0   96.0   96.0   96.0  …     NaN
104    7  97.0  97.0  96.0  97.0  1.0  97.0   96.0   96.0   96.0  …     NaN

    E1_09  Suppquex       Id  Region  CDC_Region  Industry  Size  Varstrata  \
14    NaN       2.0   1539.0     2.0         4.0       7.0   5.0        0.0
28    NaN       2.0   2755.0     3.0         5.0       7.0   6.0        0.0
104   NaN       2.0  12982.0     1.0         1.0       7.0   8.0        0.0

    Finalwt_worksite,,,,
14      47.793940929,,,,
28      47.793940929,,,,
104     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]

```
[3]: wd = [x for x in whpps.columns if x.startswith('WD')]
     cols = ['Industry', 'Size', 'OC3', 'HI1', 'HI2', 'HI3', 'CP1', 'WL6'] + wd
     whpps = whpps[cols]
     whpps.head()
```

```
[3]:    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]

```
[4]: whpps = whpps.rename({'OC3': 'org_status',
                           'HI1': 'premium_payment',
                           'HI2': 'premium_prop',
                           'HI3': 'part-time_insurance',
                           'CP1': 'health_education',
                           'WL6': 'telework',
                           'WD1_1': 'under_30',
                           'WD1_2': 'over_60',
                           'WD2': 'female',
                           'WD3': 'hourly_nonexempt',
                           'WD4': 'nondaytime',
                           'WD5': 'remote_offsite',
                           'WD6': 'union',
                           'WD7': 'turnover'}, axis=1)
     whpps.head()
```

```
[4]:    Industry  Size  org_status  premium_payment  premium_prop  \
     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  telework  under_30  over_60  female  \
0                  2.0               1.0       1.0      25.0     20.0    85.0
1                  1.0               1.0       1.0     997.0    997.0    90.0
2                  1.0               1.0       1.0      35.0      4.0   997.0
3                  1.0               2.0       2.0      50.0     15.0    50.0
4                  1.0               1.0       1.0      50.0     40.0    60.0

   hourly_nonexempt  nondaytime  remote_offsite  union  turnover
0              60.0        40.0            15.0    0.0      22.0
1              90.0       997.0           997.0    0.0     997.0
2             997.0        40.0            15.0  997.0     997.0
3              85.0        75.0             0.0    0.0     997.0
4              60.0        40.0            30.0    0.0      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]

[5]:
```
industry_map = {1: 'Manufacturing & Construction',
                2: 'Trade & Transportation',
                3: 'Hospitality & Services',
                4: 'Finance & Technology',
                5: 'Education & Healthcare',
                6: 'Public Administration',
                7: 'Hospital worksites' }
whpps.Industry = whpps.Industry.map(industry_map)
whpps.Industry
```

[5]:
```
0          Hospital worksites
1          Hospital worksites
2          Hospital worksites
3          Hospital worksites
4          Hospital worksites
                ...
2838    Public Administration
2839    Public Administration
2840    Public Administration
2841    Public Administration
2842    Public Administration
Name: Industry, Length: 2843, dtype: object
```

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]

```
[6]: size_map = {1: 'Small',
                 2: 'Small',
                 3: 'Small',
                 4: 'Medium',
                 5: 'Medium',
                 6: "Large",
                 7: "Large",
                 8: "Large"}
     whpps.Size = whpps.Size.map(size_map)
     whpps.Size
```

```
[6]: 0         Large
     1         Large
     2         Large
     3        Medium
     4        Medium
               …
     2838     Medium
     2839     Medium
     2840      Large
     2841      Large
     2842      Large
     Name: Size, Length: 2843, dtype: object
```

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]

```
[7]: replace_map = {'org_status': {1:'For Profit', 2: 'For Profit', 3:'Non-profit',␣
       ↪4:'Government', 5:'Government', 6: np.nan, 7: np.nan, 8: np.nan},
                    'premium_payment': {1:'Full', 2:'Partial', 3:'No', 97: np.nan,␣
       ↪98: np.nan, 99: np.nan},
                    'premium_prop': {1: 'Full', 2: 'Partial', 3: 'No', 97: np.nan,␣
       ↪98: np.nan, 99: np.nan},
                    'part-time_insurance': {1: 'Yes', 2: 'No', 97: np.nan, 98: np.
       ↪nan, 99: np.nan},
                    'health_education': {1: 'Yes', 2: 'No', 97: np.nan, 98: np.nan},
                    'telework': {1: 'Yes', 2: 'No', 97: np.nan, 98: np.nan, 99: np.
       ↪nan}}
     whpps = whpps.replace(replace_map)
     whpps.head()
```

[7]:
```
             Industry    Size   org_status premium_payment premium_prop  \
0  Hospital worksites   Large   Non-profit         Partial         Full
1  Hospital worksites   Large   Non-profit         Partial           No
2  Hospital worksites   Large   Non-profit            Full           No
3  Hospital worksites  Medium   For Profit            Full      Partial
4  Hospital worksites  Medium   Non-profit            Full           No

  part-time_insurance health_education telework  under_30  over_60  female  \
0                  No              Yes      Yes      25.0     20.0    85.0
1                 Yes              Yes      Yes     997.0    997.0    90.0
2                 Yes              Yes      Yes      35.0      4.0   997.0
3                 Yes               No       No      50.0     15.0    50.0
4                 Yes              Yes      Yes      50.0     40.0    60.0

   hourly_nonexempt  nondaytime  remote_offsite  union  turnover
0              60.0        40.0            15.0    0.0      22.0
1              90.0       997.0           997.0    0.0     997.0
2             997.0        40.0            15.0  997.0     997.0
3              85.0        75.0             0.0    0.0     997.0
4              60.0        40.0            30.0    0.0      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]

[8]:
```
whpps = whpps.replace([997, 998, 999], np.nan)
whpps.head()
```

[8]:
```
             Industry    Size   org_status premium_payment premium_prop  \
0  Hospital worksites   Large   Non-profit         Partial         Full
1  Hospital worksites   Large   Non-profit         Partial           No
2  Hospital worksites   Large   Non-profit            Full           No
3  Hospital worksites  Medium   For Profit            Full      Partial
4  Hospital worksites  Medium   Non-profit            Full           No

  part-time_insurance health_education telework  under_30  over_60  female  \
0                  No              Yes      Yes      25.0     20.0    85.0
1                 Yes              Yes      Yes       NaN      NaN    90.0
2                 Yes              Yes      Yes      35.0      4.0     NaN
3                 Yes               No       No      50.0     15.0    50.0
4                 Yes              Yes      Yes      50.0     40.0    60.0

   hourly_nonexempt  nondaytime  remote_offsite  union  turnover
0              60.0        40.0            15.0    0.0      22.0
1              90.0         NaN             NaN    0.0       NaN
```

```
2                 NaN        40.0             15.0   NaN       NaN
3                85.0        75.0              0.0   0.0       NaN
4                60.0        40.0             30.0   0.0      28.0
```

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]

```
[9]: whpps.sort_values(by=['Industry', 'Size', 'under_30'], ascending=[True, True,␣
      ↪False])
     whpps.head()
```

```
[9]:               Industry    Size  org_status premium_payment premium_prop  \
     0  Hospital worksites   Large  Non-profit         Partial         Full
     1  Hospital worksites   Large  Non-profit         Partial           No
     2  Hospital worksites   Large  Non-profit            Full           No
     3  Hospital worksites  Medium  For Profit            Full      Partial
     4  Hospital worksites  Medium  Non-profit            Full           No

       part-time_insurance health_education telework  under_30  over_60  female  \
     0                  No              Yes      Yes      25.0     20.0    85.0
     1                 Yes              Yes      Yes       NaN      NaN    90.0
     2                 Yes              Yes      Yes      35.0      4.0     NaN
     3                 Yes               No       No      50.0     15.0    50.0
     4                 Yes              Yes      Yes      50.0     40.0    60.0

        hourly_nonexempt  nondaytime  remote_offsite  union  turnover
     0              60.0        40.0            15.0    0.0      22.0
     1              90.0         NaN             NaN    0.0       NaN
     2               NaN        40.0            15.0    NaN       NaN
     3              85.0        75.0             0.0    0.0       NaN
     4              60.0        40.0            30.0    0.0      28.0
```

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]

```
[10]: null = np.where(whpps.Industry.isna())
      null
      whpps = whpps.drop(null[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]

```
[11]: whpps = whpps.assign(gender_balance =
                          pd.cut(whpps.female,
                                 bins=[0, 35, 65, 100],
                                 labels=("Mostly men", "Balanced", "Mostly women")))
      whpps.head()
```

```
[11]:              Industry    Size  org_status premium_payment premium_prop  \
      0  Hospital worksites   Large  Non-profit         Partial         Full
      1  Hospital worksites   Large  Non-profit         Partial           No
      2  Hospital worksites   Large  Non-profit            Full           No
      3  Hospital worksites  Medium  For Profit            Full      Partial
      4  Hospital worksites  Medium  Non-profit            Full           No

        part-time_insurance health_education telework  under_30  over_60  female  \
      0                  No              Yes      Yes      25.0     20.0    85.0
      1                 Yes              Yes      Yes       NaN      NaN    90.0
      2                 Yes              Yes      Yes      35.0      4.0     NaN
      3                 Yes               No       No      50.0     15.0    50.0
      4                 Yes              Yes      Yes      50.0     40.0    60.0

         hourly_nonexempt  nondaytime  remote_offsite  union  turnover  \
      0              60.0        40.0            15.0    0.0      22.0
      1              90.0         NaN             NaN    0.0       NaN
      2               NaN        40.0            15.0    NaN       NaN
      3              85.0        75.0             0.0    0.0       NaN
      4              60.0        40.0            30.0    0.0      28.0

        gender_balance
      0   Mostly women
      1   Mostly women
      2            NaN
      3       Balanced
      4       Balanced
```

1.13 Problem 11

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

```
[12]: cats = ['Industry', 'Size', 'org_status', 'premium_payment', 'premium_prop',␣
       ↪'part-time_insurance', 'health_education', 'telework', 'gender_balance']
      whpps[cats] = whpps[cats].astype('category')
      whpps.dtypes
```

```
[12]: Industry                category
      Size                    category
      org_status              category
      premium_payment         category
      premium_prop            category
      part-time_insurance     category
      health_education        category
      telework                category
      under_30                 float64
      over_60                  float64
      female                   float64
      hourly_nonexempt         float64
      nondaytime               float64
      remote_offsite           float64
      union                    float64
      turnover                 float64
      gender_balance          category
      dtype: object
```

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]

```
[13]: whpps.query('Size == "Small" & telework == "Yes"').stb.freq(['premium_payment'])
```

```
[13]:   premium_payment  count    percent  cumulative_count  cumulative_percent
      0            Full    324  46.285714               324           46.285714
      1         Partial    310  44.285714               634           90.571429
      2              No     66   9.428571               700          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]

```
[ ]: whadb = sqlite3.connect('whadb.db')
     whpps.to_sql('whpps', whadb, index=False, chunksize=1000, if_exists = 'replace')
```

```
[ ]: 2842
```

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]

```
[15]: myquery14 = '''
      SELECT Size, org_status, premium_payment AS insurance, female FROM whpps
      WHERE industry = 'Hospital worksites' AND premium_prop='Partial'
      ORDER BY female DESC
      '''
      pd.read_sql_query(myquery14, whadb)
```

```
[15]:       Size  org_status insurance  female
      0   Medium  Non-profit      Full    89.0
      1    Large  Non-profit   Partial    80.0
      2    Large  Non-profit   Partial    80.0
      3    Small  Non-profit      Full    75.0
      4   Medium  Non-profit   Partial    65.0
      5   Medium  For Profit      Full    50.0
      6   Medium        97.0   Partial     NaN
      7   Medium  Non-profit   Partial     NaN
      8   Medium  Non-profit      Full     NaN
      9   Medium  Non-profit      Full     NaN
      10   Large  Non-profit   Partial     NaN
```

```
[20]: whpps14 = whpps.query('Industry=="Hospital worksites" &␣
        ↪premium_prop=="Partial"').sort_values(by='female', ascending=False)
      col14 = ['Size', 'org_status', 'premium_payment', 'female']
      whpps14 = whpps14[col14]
      whpps14 = whpps14.rename({'premium_payment': 'insurance'}, axis=1)
      whpps14
```

```
[20]:        Size  org_status insurance  female
      320  Medium  Non-profit      Full    89.0
      187   Large  Non-profit   Partial    80.0
      214   Large  Non-profit   Partial    80.0
      229   Small  Non-profit      Full    75.0
      191  Medium  Non-profit   Partial    65.0
      3    Medium  For Profit      Full    50.0
      11   Medium        97.0   Partial     NaN
      48   Medium  Non-profit   Partial     NaN
      51   Medium  Non-profit      Full     NaN
      75   Medium  Non-profit      Full     NaN
      97    Large  Non-profit   Partial     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]

[30]:

```
myquery15 = '''
SELECT industry,
    AVG(female) as percent_female,
    AVG(under_30) as percent_under30,
    AVG(over_60) as percent_over60
FROM whpps
GROUP BY industry
ORDER BY percent_female DESC
'''
pd.read_sql_query(myquery15, whadb)
```

[30]:

```
                         Industry  percent_female  percent_under30  \
0          Education & Healthcare       80.657143        25.745665
1              Hospital worksites       76.427027        27.213793
2          Hospitality & Services       53.804416        38.566343
3            Finance & Technology       50.632184        23.821752
4           Public Administration       39.056738        21.015625
5          Trade & Transportation       32.657258        29.108696
6   Manufacturing & Construction       20.328605        22.257143

   percent_over60
0       11.349570
1       16.489655
2       11.544872
3       12.465465
4       15.015385
5       12.584034
6       10.690355
```

[29]:

```
whpps15 = whpps.groupby('Industry').agg({'female': 'mean', 'under_30': 'mean',␣
 ↪'over_60': 'mean'}).sort_values('female', ascending=False)
whpps15 = whpps15.reset_index()
whpps15
```

```
[29]:                        Industry     female   under_30    over_60
      0        Education & Healthcare  80.657143  25.745665  11.349570
      1            Hospital worksites  76.427027  27.213793  16.489655
      2        Hospitality & Services  53.804416  38.566343  11.544872
      3          Finance & Technology  50.632184  23.821752  12.465465
      4         Public Administration  39.056738  21.015625  15.015385
      5        Trade & Transportation  32.657258  29.108696  12.584034
      6  Manufacturing & Construction  20.328605  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]

```
[ ]: myquery16 = '''
     SELECT gender_balance, premium_payment, COUNT(*)
     FROM whpps
     GROUP BY gender_balance, premium_payment
     HAVING gender_balance is NOT NULL and premium_payment is NOT NULL
     '''
     pd.read_sql_query(myquery16, whadb)
```

```
[ ]:   gender_balance premium_payment  COUNT(*)
     0       Balanced            Full       226
     1       Balanced              No        77
     2       Balanced         Partial       271
     3     Mostly men            Full       293
     4     Mostly men              No        87
     5     Mostly men         Partial       321
     6   Mostly women            Full       267
     7   Mostly women              No       107
     8   Mostly women         Partial       333
```

```
[38]: whpps16 = whpps.groupby(['gender_balance', 'premium_payment']).size().
       ↪reset_index(name='count').sort_values('gender_balance', ascending=True)
      whpps16
```

```
[38]:   gender_balance premium_payment  count
      0     Mostly men            Full    293
      1     Mostly men              No     87
      2     Mostly men         Partial    321
      3       Balanced            Full    226
      4       Balanced              No     77
```

```
5       Balanced         Partial    271
6   Mostly women            Full    267
7   Mostly women              No    107
8   Mostly women         Partial    333
```

The sort didn't work for some reason, but it's all there. Balanced is always in the middle no matter how I sort.