lab-assignment8

July 30, 2022

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]:
```
data = pd.read_csv('https://www.cdc.gov/workplacehealthpromotion/
↪data-surveillance/docs/whpps_120717.csv',sep='~')
data.head()
```

[2]:

	OC1	OC3	HI1	HI2	HI3	HI4	HRA1	HRA1A	HRA1B	HRA1E	…	WL3_05	E1_09	\
0	7	3.0	2.0	1.0	2.0	1.0	1.0	3.0	4.0	2.0	…	PTO	NaN	
1	2	3.0	2.0	3.0	1.0	1.0	1.0	3.0	3.0	1.0	…	NaN	NaN	
2	7	3.0	1.0	3.0	1.0	1.0	1.0	3.0	97.0	2.0	…	NaN	NaN	
3	1	2.0	1.0	2.0	1.0	1.0	97.0	96.0	96.0	96.0	…	NaN	NaN	
4	2	3.0	1.0	3.0	1.0	1.0	1.0	3.0	3.0	2.0	…	NaN	NaN	

	Suppquex	Id	Region	CDC_Region	Industry	Size	Varstrata	\
0	2.0	217.0	1.0	2.0	7.0	7.0	0.0	
1	1.0	326.0	3.0	7.0	7.0	6.0	0.0	
2	1.0	399.0	4.0	8.0	7.0	8.0	0.0	
3	1.0	475.0	5.0	9.0	7.0	4.0	0.0	
4	1.0	489.0	2.0	4.0	7.0	4.0	0.0	

	Finalwt_worksite,,,,
0	47.793940929,,,,
1	47.793940929,,,,
2	47.793940929,,,,
3	47.793940929,,,,
4	47.793940929,,,,

[5 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]:
```
filter_col = [col for col in data if col.startswith('WD')]
wdata = data[['Industry',␣
 ↪'Size','OC3','HI1','HI2','HI3','CP1','WL6']+filter_col]
wdata.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]:
```
# OC3 Is your organization for profit, non-profit, government
# HI1  offer full, partial or no payment of premiums
# HI2 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 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?
```

[5]:
```
wdata.rename(columns = {'Industry':'Industry', 'Size':'NumEmploy','OC3':
 ↪'OrgType',
                        'HI1':'CompPrem','HI2':'EmployPrem','HI3':
 ↪'PartInsur','CP1':'HealthEdProg',
```

```
                        'WL6':'WfromH','WD1_1':'EmployPerUnder30',
                        'WD1_2':'EmployPerOver60',
                        'WD2':'EmployFPer',
                        'WD3':'EmployHourPer',
                        'WD4':'OddHourPer',
                        'WD5':'WorkRemotePer',
                        'WD6':'UnionPer',
                        'WD7':'TurnoverPer',
                           }, inplace = True)
wdata.head()
```

[5]:

	Industry	NumEmploy	OrgType	CompPrem	EmployPrem	PartInsur	\
0	7.0	7.0	3.0	2.0	1.0	2.0	
1	7.0	6.0	3.0	2.0	3.0	1.0	
2	7.0	8.0	3.0	1.0	3.0	1.0	
3	7.0	4.0	2.0	1.0	2.0	1.0	
4	7.0	4.0	3.0	1.0	3.0	1.0	

	HealthEdProg	WfromH	EmployPerUnder30	EmployPerOver60	EmployFPer	\
0	1.0	1.0	25.0	20.0	85.0	
1	1.0	1.0	997.0	997.0	90.0	
2	1.0	1.0	35.0	4.0	997.0	
3	2.0	2.0	50.0	15.0	50.0	
4	1.0	1.0	50.0	40.0	60.0	

	EmployHourPer	OddHourPer	WorkRemotePer	UnionPer	TurnoverPer
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]

[6]:
```
# 1 - 11:Agriculture, Forestry, Fishing and Hunting, 21:Mining, 22:Utilities,␣
 ↪23:Construction, 31-33:Manufacturing
# 2 - 42:Wholesale Trade, 44-45:Retail Trade, 48-49:Transportation and␣
 ↪Warehousing
# 3 - 71:Arts, Entertainment, and Recreation, 72:Accommodation and Food␣
 ↪Services, 81:Other Services (except Public Administration)
# 4 - 51:Information, 52:Finance and Insurance, 53:Real Estate Rental and␣
 ↪Leasing, 54:Professional, Scientific, and Technical Services, 55:Management␣
 ↪of Companies and Enterprises, 56:Administrative and Support and Waste␣
 ↪Management and Remediation Services
```

```
# 5 - 61:Educational Services, 62 (excluding hospital worksites): Health Care␣
 ↪and Social Assistance
# 6 - 92:Public Administration
# 7 - Hospital worksites (NAICS6 = 622110, 622210, 622310)
replace_map = {1:'Agriculture and Manfacturing',
               2:'Retail, Wholesale and Transportation',
               3:'Entertainment and Services',
               4:'IT, Finance, Real Estate, Tech Services, Waste Management',
                5: 'Education,Health Care and Social Assistance',
                  6: 'Public Admin',
              7: 'Hospial Worksites'}
wdata.Industry = wdata.Industry.map(replace_map)
wdata.Industry
```

[6]:
```
0       Hospial Worksites
1       Hospial Worksites
2       Hospial Worksites
3       Hospial Worksites
4       Hospial Worksites
              …
2838         Public Admin
2839         Public Admin
2840         Public Admin
2841         Public Admin
2842         Public Admin
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]

[7]:
```
# 1 = Size Category 1: 10-24
# 2 = Size Category 2: 25-49
# 3 = Size Category 3: 50-99
# 4 = Size Category 4: 100-249
# 5 = Size Category 5: 250-499
# 6 = Size Category 6: 500-749
# 7 = Size Category 7: 750-999
# 8 = Size Category 8: 1,000+
```

[8]:
```
wdata['NumEmploy'] = pd.cut(wdata['NumEmploy'], bins=[0, 3, 5, 8],␣
 ↪labels=['Small', 'Medium', 'Large'])
wdata.head()
```

[8]:
```
             Industry NumEmploy  OrgType  CompPrem  EmployPrem  PartInsur  \
0  Hospial Worksites      Large      3.0       2.0         1.0        2.0
1  Hospial Worksites      Large      3.0       2.0         3.0        1.0
2  Hospial Worksites      Large      3.0       1.0         3.0        1.0
3  Hospial Worksites     Medium      2.0       1.0         2.0        1.0
4  Hospial Worksites     Medium      3.0       1.0         3.0        1.0

   HealthEdProg  WfromH  EmployPerUnder30  EmployPerOver60  EmployFPer  \
0           1.0     1.0              25.0             20.0        85.0
1           1.0     1.0             997.0            997.0        90.0
2           1.0     1.0              35.0              4.0       997.0
3           2.0     2.0              50.0             15.0        50.0
4           1.0     1.0              50.0             40.0        60.0

   EmployHourPer  OddHourPer  WorkRemotePer  UnionPer  TurnoverPer
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
```

1.8 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]

[9]:
```
replace_map ={ 'OrgType': {1:'For profit, public',
               2:'For profit, private',
               3:'Non-profit',
               4:'State or local government',
                5: 'Federal government ',
                  6: 'Other',
              97: np.nan,
              98: np.nan,
              99: np.nan},
              "CompPrem":
              {1:'Full insurance coverage offered',
               2:'Partial insurance coverage offered',
               3:'No insurance coverage offered ',
              97: np.nan,
              98: np.nan,
              99: np.nan},
              'EmployPrem':
             {1:'Larger',
                2:'Smaller',
                3:'About the Same ',
```

```
            96: np.nan,
              97: np.nan,
              98: np.nan,
              99: np.nan},
              'PartInsur':
              {1:'Yes',
               2:'No',
              97: np.nan,
              98: np.nan,
              99: np.nan},
              'HealthEdProg':
              {1:'Yes',
               2:'No',
              97: np.nan,
              98: np.nan},
              'WfromH':
              {1:"Yes",
               2:"No",
               97:np.nan,
               98:np.nan,
               99:np.nan}
             }
wdata = wdata.replace(replace_map)
wdata.head()
```

[9]:

	Industry	NumEmploy	OrgType	\
0	Hospial Worksites	Large	Non-profit	
1	Hospial Worksites	Large	Non-profit	
2	Hospial Worksites	Large	Non-profit	
3	Hospial Worksites	Medium	For profit, private	
4	Hospial Worksites	Medium	Non-profit	

	CompPrem	EmployPrem	PartInsur	HealthEdProg	\
0	Partial insurance coverage offered	Larger	No	Yes	
1	Partial insurance coverage offered	About the Same	Yes	Yes	
2	Full insurance coverage offered	About the Same	Yes	Yes	
3	Full insurance coverage offered	Smaller	Yes	No	
4	Full insurance coverage offered	About the Same	Yes	Yes	

	WfromH	EmployPerUnder30	EmployPerOver60	EmployFPer	EmployHourPer	\
0	Yes	25.0	20.0	85.0	60.0	
1	Yes	997.0	997.0	90.0	90.0	
2	Yes	35.0	4.0	997.0	997.0	
3	No	50.0	15.0	50.0	85.0	
4	Yes	50.0	40.0	60.0	60.0	

	OddHourPer	WorkRemotePer	UnionPer	TurnoverPer

```
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
```

1.9 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]

[10]:
```
replace_map = {997:np.nan,998:np.nan,999:np.nan}
wdata = wdata.replace(replace_map)
wdata.head()
```

[10]:
```
             Industry NumEmploy              OrgType  \
0  Hospial Worksites     Large           Non-profit
1  Hospial Worksites     Large           Non-profit
2  Hospial Worksites     Large           Non-profit
3  Hospial Worksites    Medium  For profit, private
4  Hospial Worksites    Medium           Non-profit

                              CompPrem       EmployPrem PartInsur HealthEdProg  \
0  Partial insurance coverage offered           Larger        No          Yes
1  Partial insurance coverage offered  About the Same        Yes          Yes
2     Full insurance coverage offered  About the Same        Yes          Yes
3     Full insurance coverage offered         Smaller       Yes           No
4     Full insurance coverage offered  About the Same        Yes          Yes

  WfromH  EmployPerUnder30  EmployPerOver60  EmployFPer  EmployHourPer  \
0    Yes              25.0             20.0        85.0           60.0
1    Yes               NaN              NaN        90.0           90.0
2    Yes              35.0              4.0         NaN            NaN
3     No              50.0             15.0        50.0           85.0
4    Yes              50.0             40.0        60.0           60.0

   OddHourPer  WorkRemotePer  UnionPer  TurnoverPer
0        40.0           15.0       0.0         22.0
1         NaN            NaN       0.0          NaN
2        40.0           15.0       NaN          NaN
3        75.0            0.0       0.0          NaN
4        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]

[11]:
```
wdata = wdata.
 ↪sort_values(['Industry','NumEmploy','EmployPerUnder30'],ascending=[True,True,False])
wdata.head()
```

[11]:

	Industry	NumEmploy	OrgType	CompPrem	EmployPrem	PartInsur	HealthEdProg	WfromH	EmployPerUnder30	EmployPerOver60	EmployFPer	EmployHourPer	OddHourPer	WorkRemotePer	UnionPer	TurnoverPer
900	Agriculture and Manfacturing	Small	NaN	No insurance coverage offered	NaN	No	No	Yes	100.0	0.0	90.0	90.0	1.0	1.0	0.0	80.0
2034	Agriculture and Manfacturing	Small	For profit, private	NaN	NaN	No	No	No	100.0	0.0	40.0	95.0	100.0	0.0	0.0	NaN
2051	Agriculture and Manfacturing	Small	For profit, private	Full insurance coverage offered	About the Same	No	No	Yes	95.0	1.0	1.0	50.0	50.0	5.0	0.0	30.0
542	Agriculture and Manfacturing	Small	For profit, private	Partial insurance coverage offered	About the Same	No	No	Yes	90.0	20.0	NaN	NaN	NaN	NaN	NaN	NaN
1188	Agriculture and Manfacturing	Small	For profit, private	Partial insurance coverage offered	About the Same	No	No	No	80.0	15.0	2.0	80.0	0.0	0.0	0.0	2.0

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]

[12]:
```
wdata = wdata[wdata.Industry.notnull()]
wdata.head()
```

[12]:

	Industry	NumEmploy	OrgType
900	Agriculture and Manfacturing	Small	NaN

```
2034  Agriculture and Manfacturing     Small  For profit, private
2051  Agriculture and Manfacturing     Small  For profit, private
542   Agriculture and Manfacturing     Small  For profit, private
1188  Agriculture and Manfacturing     Small  For profit, private

                                CompPrem       EmployPrem PartInsur  \
900       No insurance coverage offered             NaN        No
2034                                 NaN             NaN        No
2051     Full insurance coverage offered  About the Same         No
542   Partial insurance coverage offered  About the Same         No
1188  Partial insurance coverage offered  About the Same         No

     HealthEdProg WfromH  EmployPerUnder30  EmployPerOver60  EmployFPer  \
900            No    Yes             100.0              0.0        90.0
2034           No     No             100.0              0.0        40.0
2051           No    Yes              95.0              1.0         1.0
542            No    Yes              90.0             20.0         NaN
1188           No     No              80.0             15.0         2.0

      EmployHourPer  OddHourPer  WorkRemotePer  UnionPer  TurnoverPer
900            90.0         1.0            1.0       0.0         80.0
2034           95.0       100.0            0.0       0.0          NaN
2051           50.0        50.0            5.0       0.0         30.0
542             NaN         NaN            NaN       NaN          NaN
1188           80.0         0.0            0.0       0.0          2.0
```

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]

[13]:
```
wdata['gender_balance'] = pd.cut(wdata['EmployFPer'], bins=[0, 35.0, 65.0, 100.
↪0], labels=['Mostly Men', 'Balanced', 'Mostly Women'])
wdata.head()
```

[13]:
```
                        Industry NumEmploy              OrgType  \
900   Agriculture and Manfacturing     Small                  NaN
2034  Agriculture and Manfacturing     Small  For profit, private
2051  Agriculture and Manfacturing     Small  For profit, private
542   Agriculture and Manfacturing     Small  For profit, private
1188  Agriculture and Manfacturing     Small  For profit, private

                                CompPrem       EmployPrem PartInsur  \
900       No insurance coverage offered             NaN        No
2034                                 NaN             NaN        No
```

```
2051     Full insurance coverage offered  About the Same         No
542   Partial insurance coverage offered  About the Same         No
1188  Partial insurance coverage offered  About the Same         No

     HealthEdProg WfromH  EmployPerUnder30  EmployPerOver60  EmployFPer  \
900            No    Yes             100.0              0.0        90.0
2034           No     No             100.0              0.0        40.0
2051           No    Yes              95.0              1.0         1.0
542            No    Yes              90.0             20.0         NaN
1188           No     No              80.0             15.0         2.0

      EmployHourPer  OddHourPer  WorkRemotePer  UnionPer  TurnoverPer  \
900            90.0         1.0            1.0       0.0         80.0
2034           95.0       100.0            0.0       0.0          NaN
2051           50.0        50.0            5.0       0.0         30.0
542             NaN         NaN            NaN       NaN          NaN
1188           80.0         0.0            0.0       0.0          2.0

     gender_balance
900    Mostly Women
2034       Balanced
2051     Mostly Men
542             NaN
1188     Mostly Men
```

1.13 Problem 11

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

[14]:
```
cols = ['Industry', 'OrgType', 'CompPrem', 'EmployPrem',
       'PartInsur', 'HealthEdProg', 'WfromH']
wdata[cols] = wdata[cols].astype('category')
wdata.dtypes
```

[14]:
```
Industry            category
NumEmploy           category
OrgType             category
CompPrem            category
EmployPrem          category
PartInsur           category
HealthEdProg        category
WfromH              category
EmployPerUnder30     float64
EmployPerOver60      float64
EmployFPer           float64
EmployHourPer        float64
```

```
OddHourPer           float64
WorkRemotePer        float64
UnionPer             float64
TurnoverPer          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]

[15]:
```
filter_data = wdata.query("NumEmploy=='Small' & WfromH == 'Yes'").stb.
↪freq(['CompPrem'])
filter_data
```

[15]:
```
                            CompPrem  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 using 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]

[16]:
```
WHA_db = sqlite3.connect("WHA.db")
wdata.to_sql('WHA',WHA_db,if_exists='replace',index=False,chunksize=1000)
```

[16]:
```
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]

[17]:
```
myquery = """ SELECT NumEmploy, OrgType, CompPrem AS insurance, EmployFPer
 ↪FROM WHA
WHERE Industry = 'Hospial Worksites' AND EmployPrem='Smaller'
ORDER BY EmployFPer DESC
"""
pd.read_sql_query(myquery,WHA_db)
```

[17]:

	NumEmploy	OrgType	insurance	EmployFPer
0	Medium	Non-profit	Full insurance coverage offered	89.0
1	Large	Non-profit	Partial insurance coverage offered	80.0
2	Large	Non-profit	Partial insurance coverage offered	80.0
3	Small	Non-profit	Full insurance coverage offered	75.0
4	Medium	Non-profit	Partial insurance coverage offered	65.0
5	Medium	For profit, private	Full insurance coverage offered	50.0
6	Medium	Non-profit	Full insurance coverage offered	NaN
7	Medium	None	Partial insurance coverage offered	NaN
8	Medium	Non-profit	Partial insurance coverage offered	NaN
9	Medium	Non-profit	Full insurance coverage offered	NaN
10	Large	Non-profit	Partial insurance coverage offered	NaN

[18]:
```
table = wdata.query("Industry=='Hospial Worksites' & EmployPrem=='Smaller'").
 ↪reset_index()
table[['NumEmploy', 'OrgType', 'CompPrem', 'EmployFPer']].rename(columns =
 ↪{'CompPrem':'insurance'}).sort_values('EmployFPer',ascending=False).
 ↪reset_index(drop=True)
```

[18]:

	NumEmploy	OrgType	insurance	EmployFPer
0	Medium	Non-profit	Full insurance coverage offered	89.0
1	Large	Non-profit	Partial insurance coverage offered	80.0
2	Large	Non-profit	Partial insurance coverage offered	80.0
3	Small	Non-profit	Full insurance coverage offered	75.0

```
4     Medium           Non-profit  Partial insurance coverage offered
5     Medium  For profit, private     Full insurance coverage offered
6     Medium           Non-profit     Full insurance coverage offered
7     Medium                  NaN  Partial insurance coverage offered
8     Medium           Non-profit  Partial insurance coverage offered
9     Medium           Non-profit     Full insurance coverage offered
10     Large           Non-profit  Partial insurance coverage offered

    EmployFPer
0         89.0
1         80.0
2         80.0
3         75.0
4         65.0
5         50.0
6          NaN
7          NaN
8          NaN
9          NaN
10         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]

[19]:
```
myquery = """
SELECT Industry,
    AVG(EmployFPer) as percent_female,
    AVG(EmployPerunder30) as percent_under30,
    AVG(EmployPerOver60) as percent_over60
FROM WHA
GROUP BY Industry
ORDER BY AVG(EmployFPer) DESC
"""

pd.read_sql_query(myquery,WHA_db)
```

[19]:

	Industry	percent_female	percent_under30	percent_over60
0	Education,Health Care and Social Assistance	80.657143	25.745665	11.349570
1	Hospial Worksites	76.427027	27.213793	16.489655
2	Entertainment and Services	53.804416	38.566343	11.544872
3	IT, Finance, Real Estate, Tech Services, Waste…	50.632184	23.821752	12.465465
4	Public Admin	39.056738	21.015625	15.015385
5	Retail, Wholesale and Transportation	32.657258	29.108696	12.584034
6	Agriculture and Manfacturing	20.328605	22.257143	10.690355

[20]:
```
table = wdata.groupby('Industry').agg({'EmployFPer':'mean','EmployPerUnder30':
↪'mean','EmployPerOver60':'mean'}).reset_index().
↪sort_values('EmployFPer',ascending=False,ignore_index=True)
table.rename(columns = {'EmployFPer':'percent_female','EmployPerUnder30':
↪'percent_under30','EmployPerOver60':'percent_over60'})
```

[20]:

	Industry	percent_female	percent_under30	percent_over60
0	Education,Health Care and Social Assistance	80.657143	25.745665	11.349570
1	Hospial Worksites	76.427027	27.213793	16.489655
2	Entertainment and Services	53.804416	38.566343	11.544872
3	IT, Finance, Real Estate, Tech Services, Waste…	50.632184	23.821752	12.465465
4	Public Admin	39.056738	21.015625	15.015385
5	Retail, Wholesale and Transportation	32.657258	29.108696	12.584034
6	Agriculture and Manfacturing	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]

[21]:
```
myquery = """
SELECT gender_balance, CompPrem, COUNT(*)
FROM WHA
GROUP BY gender_balance, CompPrem
HAVING gender_balance is NOT NULL and CompPrem is NOT NULL
"""

pd.read_sql_query(myquery,WHA_db)
```

[21]:

	gender_balance	CompPrem	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	293
4	Mostly Men	No insurance coverage offered	87
5	Mostly Men	Partial insurance coverage offered	321
6	Mostly Women	Full insurance coverage offered	267
7	Mostly Women	No insurance coverage offered	107
8	Mostly Women	Partial insurance coverage offered	333

[22]:
```
table = wdata.query('gender_balance.notnull()&CompPrem.notnull()').
↪groupby(['gender_balance','CompPrem']).size().reset_index().
↪sort_values(['gender_balance'],key=lambda col: col.str.
↪lower(),ignore_index=True).rename(columns={0:'COUNT(*)'})
table
```

[22]:

	gender_balance	CompPrem	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	293
4	Mostly Men	No insurance coverage offered	87
5	Mostly Men	Partial insurance coverage offered	321
6	Mostly Women	Full insurance coverage offered	267
7	Mostly Women	No insurance coverage offered	107
8	Mostly Women	Partial insurance coverage offered	333