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

DS 6001: Practice and Application of Data Science

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.

Problem 0

Import the following libraries:

In [1]:
```
!pip install sidetable
```

```
Requirement already satisfied: sidetable in c:\users\valenty\anaconda3\lib\site-packages (0.9.1)
Requirement already satisfied: pandas>=1.0 in c:\users\valenty\anaconda3\lib\site-packages (from sidetable) (2.0.3)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\valenty\anaconda3\lib\site-packages (from pandas>=1.0->sidetable) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\valenty\anaconda3\lib\site-packages (from pandas>=1.0->sidetable) (2024.1)
Requirement already satisfied: tzdata>=2022.1 in c:\users\valenty\anaconda3\lib\site-packages (from pandas>=1.0->sidetable) (2023.3)
Requirement already satisfied: numpy>=1.20.3 in c:\users\valenty\anaconda3\lib\site-packages (from pandas>=1.0->sidetable) (1.24.3)
Requirement already satisfied: six>=1.5 in c:\users\valenty\anaconda3\lib\site-packages (from python-dateutil>=2.8.2->pandas>=1.0->sidetable) (1.16.
0)
```

In [2]:
```
import numpy as np
import pandas as pd
import sidetable
import sqlite3
import warnings
warnings.filterwarnings('ignore')
```

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]

In [3]:
```
df = pd.read_csv('https://www.cdc.gov/workplacehealthpromotion/data-surveillance/docs/whpps_120717.csv', sep='~')
df.iloc[[14,28,102], :] #this format matches that in the textbook section 8.4
```

Out[3]:

	OC1	OC3	HI1	HI2	HI3	HI4	HRA1	HRA1A	HRA1B	HRA1E	...	WL3_05	E1_09	Suppquex	Id	Region	CDC_Region	Industry	Size	Varstrata	Fi
14	7	2.0	2.0	1.0	2.0	1.0	1.0	3.0	2.0	2.0	...	NaN	NaN	2.0	1539.0	2.0	4.0	7.0	5.0	0.0	
28	1	3.0	2.0	3.0	1.0	1.0	2.0	96.0	96.0	96.0	...	NaN	NaN	2.0	2755.0	3.0	5.0	7.0	6.0	0.0	
102	1	3.0	2.0	3.0	1.0	1.0	1.0	1.0	4.0	2.0	...	NaN	NaN	2.0	12686.0	3.0	5.0	7.0	8.0	0.0	

3 rows × 301 columns

In [4]:
```
df
```

Out[4]:

	OC1	OC3	HI1	HI2	HI3	HI4	HRA1	HRA1A	HRA1B	HRA1E	...	WL3_05	E1_09	Suppquex	Id	Region	CDC_Region	Industry	Size	Varstrata
0	7	3.0	2.0	1.0	2.0	1.0	1.0	3.0	4.0	2.0	...	PTO	NaN	2.0	217.0	1.0	2.0	7.0	7.0	0.0
1	2	3.0	2.0	3.0	1.0	1.0	1.0	3.0	3.0	1.0	...	NaN	NaN	1.0	326.0	3.0	7.0	7.0	6.0	0.0
2	7	3.0	1.0	3.0	1.0	1.0	1.0	3.0	97.0	2.0	...	NaN	NaN	1.0	399.0	4.0	8.0	7.0	8.0	0.0
3	1	2.0	1.0	2.0	1.0	1.0	97.0	96.0	96.0	96.0	...	NaN	NaN	1.0	475.0	5.0	9.0	7.0	4.0	0.0
4	2	3.0	1.0	3.0	1.0	1.0	1.0	3.0	3.0	2.0	...	NaN	NaN	1.0	489.0	2.0	4.0	7.0	4.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2838	1	4.0	1.0	3.0	1.0	1.0	2.0	96.0	96.0	96.0	...	NaN	NaN	2.0	29419.0	5.0	10.0	6.0	5.0	549.0
2839	7	4.0	2.0	3.0	1.0	1.0	1.0	1.0	1.0	2.0	...	NaN	NaN	1.0	53041.0	5.0	10.0	6.0	5.0	549.0
2840	2	4.0	2.0	3.0	1.0	1.0	1.0	3.0	2.0	2.0	...	NaN	NaN	1.0	7496.0	5.0	10.0	6.0	8.0	552.0
2841	1	4.0	2.0	3.0	1.0	1.0	1.0	2.0	97.0	2.0	...	NaN	NaN	2.0	18994.0	5.0	10.0	6.0	8.0	552.0
2842	2	4.0	2.0	3.0	1.0	1.0	1.0	1.0	3.0	2.0	...	NaN	Nothing is needed	2.0	27704.0	5.0	10.0	6.0	8.0	552.0

2843 rows × 301 columns

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]

In [5]:
```
WD_start = [col for col in df.columns if col.startswith('WD')]
keep_col = ['Industry','Size','OC3','HI1','HI2','HI3','CP1','WL6'] + WD_start
```

In [6]:
```
df2 = df[keep_col]
df2.head()
```

Out[6]:

	Industry	Size	OC3	HI1	HI2	HI3	CP1	WL6	WD1_1	WD1_2	WD2	WD3	WD4	WD5	WD6	WD7
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	40.0	15.0	0.0	22.0
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	997.0	997.0	0.0	997.0
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	40.0	15.0	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	75.0	0.0	0.0	997.0
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	40.0	30.0	0.0	28.0

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]

In [7]:
```
col_map = {'OC3': 'organization_type',
           'HI1': 'insurance_coverage',
           'HI2': 'premium_proportion',
           'HI3': 'pt_insurace',
           'CP1': 'health_education',
           'WL6': 'WFH',
           'WD1_1': 'under_30',
           'WD1_2': 'over_60',
           'WD2': 'female',
           'WD3': 'hourly',
           'WD4': 'atypical_shift',
           'WD5': 'remote',
           'WD6': 'unionized',
           'WD7': 'turnover'}
df2 = df2.rename(col_map, axis=1)
```

In [8]:
```
df2.head()
```

Out[8]:

	Industry	Size	organization_type	insurance_coverage	premium_proportion	pt_insurace	health_education	WFH	under_30	over_60	female	hourly	atypical_
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	9
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	

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]

In [9]:
```
industry_map = {1 : 'natural_resources',
                2 : 'trade_transport',
                3 : 'entertainment_food',
                4 : 'business_finance',
                5 : 'health_education',
                6 : 'public_admin',
                7 : 'hospital'}
df2 = df2.replace({'Industry': industry_map})
```

In [10]:
```
df2['Industry'].value_counts()
```

Out[10]:
```
Industry
health_education      551
natural_resources     525
entertainment_food    433
business_finance      429
hospital              338
trade_transport       311
public_admin          255
Name: count, dtype: int64
```

In [11]:
```
df2.head()
```

Out[11]:

	Industry	Size	organization_type	insurance_coverage	premium_proportion	pt_insurace	health_education	WFH	under_30	over_60	female	hourly	atypical_
0	hospital	7.0	3.0	2.0	1.0	2.0	1.0	1.0	25.0	20.0	85.0	60.0	
1	hospital	6.0	3.0	2.0	3.0	1.0	1.0	1.0	997.0	997.0	90.0	90.0	9
2	hospital	8.0	3.0	1.0	3.0	1.0	1.0	1.0	35.0	4.0	997.0	997.0	
3	hospital	4.0	2.0	1.0	2.0	1.0	2.0	2.0	50.0	15.0	50.0	85.0	
4	hospital	4.0	3.0	1.0	3.0	1.0	1.0	1.0	50.0	40.0	60.0	60.0	

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]

In [12]:
```
df2['Size'] = np.where(df2['Size'] <= 3, 'Small', np.where(
                       df2['Size'] < 6, 'Medium', 'Large'))
```

In [13]:
```
df2['Size'].value_counts()
```

Out[13]:
```
Size
Small     2195
Medium     393
Large      255
Name: count, dtype: int64
```

In [14]:
```
df2.head()
```

Out[14]:

	Industry	Size	organization_type	insurance_coverage	premium_proportion	pt_insurace	health_education	WFH	under_30	over_60	female	hourly	atypi
0	hospital	Large	3.0	2.0	1.0	2.0	1.0	1.0	25.0	20.0	85.0	60.0	
1	hospital	Large	3.0	2.0	3.0	1.0	1.0	1.0	997.0	997.0	90.0	90.0	
2	hospital	Large	3.0	1.0	3.0	1.0	1.0	1.0	35.0	4.0	997.0	997.0	
3	hospital	Medium	2.0	1.0	2.0	1.0	2.0	2.0	50.0	15.0	50.0	85.0	
4	hospital	Medium	3.0	1.0	3.0	1.0	1.0	1.0	50.0	40.0	60.0	60.0	

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]

In [15]:
```
org_type_map = {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}
```

In [16]:
```
ins_cov_map = {
1 :"Full insurance coverage offered",
2 :"Partial insurance coverage offered",
3 : "No insurance coverage offered",
97 : np.nan,
98 : np.nan,
99 : np.nan}
```

In [17]:
```
prem_prop_map = {
1 : "Larger",
2 : "Smaller",
3 : "About the same",
96 : np.nan,
97 : np.nan,
98 : np.nan,
99 : np.nan}
```

In [18]:
```
pt_ins_map = {
1 : "Yes",
2 : "No",
97 : np.nan,
98 : np.nan,
99 : np.nan}
```

In [19]:
```
health_ed_map = {
    1: "Yes",
    2: "No",
    97: np.nan,
    98: np.nan}
```

In [20]:
```
wfh_map = {
    1: "Yes",
    2: "No",
    97: np.nan,
    98: np.nan,
    99: np.nan}
```

In [21]:
```
df2 = df2.replace({'organization_type': org_type_map, 'insurance_coverage': ins_cov_map,
                   'premium_proportion': prem_prop_map, 'pt_insurace': pt_ins_map, 'health_education': health_ed_map,
                   'WFH': wfh_map})
```

In [22]:
```
df2.head()
```

Out[22]:

	Industry	Size	organization_type	insurance_coverage	premium_proportion	pt_insurace	health_education	WFH	under_30	over_60	female	hourly	atypi
0	hospital	Large	Non-profit	Partial insurance coverage offered	Larger	No	Yes	Yes	25.0	20.0	85.0	60.0	
1	hospital	Large	Non-profit	Partial insurance coverage offered	About the same	Yes	Yes	Yes	997.0	997.0	90.0	90.0	
2	hospital	Large	Non-profit	Full insurance coverage offered	About the same	Yes	Yes	Yes	35.0	4.0	997.0	997.0	
3	hospital	Medium	For profit, private	Full insurance coverage offered	Smaller	Yes	No	No	50.0	15.0	50.0	85.0	
4	hospital	Medium	Non-profit	Full insurance coverage offered	About the same	Yes	Yes	Yes	50.0	40.0	60.0	60.0	

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]

In [23]:
```
rename_cols = ['under_30','over_60','female','hourly','atypical_shift','remote','unionized','turnover']
df2[rename_cols] = df2[rename_cols].replace([997, 998, 999], np.nan)
df2.head()
```

Out[23]:

	Industry	Size	organization_type	insurance_coverage	premium_proportion	pt_insurace	health_education	WFH	under_30	over_60	female	hourly	atypi
0	hospital	Large	Non-profit	Partial insurance coverage offered	Larger	No	Yes	Yes	25.0	20.0	85.0	60.0	
1	hospital	Large	Non-profit	Partial insurance coverage offered	About the same	Yes	Yes	Yes	NaN	NaN	90.0	90.0	
2	hospital	Large	Non-profit	Full insurance coverage offered	About the same	Yes	Yes	Yes	35.0	4.0	NaN	NaN	
3	hospital	Medium	For profit, private	Full insurance coverage offered	Smaller	Yes	No	No	50.0	15.0	50.0	85.0	
4	hospital	Medium	Non-profit	Full insurance coverage offered	About the same	Yes	Yes	Yes	50.0	40.0	60.0	60.0	

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]

In [24]:
```
df2 = df2.sort_values(by = ['Industry', 'Size','under_30'], ascending = [True, True, False])
```

In [25]:
```
df2
```

Out[25]:

	Industry	Size	organization_type	insurance_coverage	premium_proportion	pt_insurace	health_education	WFH	under_30	over_60	female	hourl
465	business_finance	Large	For profit, public	Full insurance coverage offered	About the same	No	Yes	Yes	75.0	1.0	49.0	25.
2471	business_finance	Large	For profit, private	Partial insurance coverage offered	Larger	No	Yes	Yes	70.0	30.0	NaN	NaI
1352	business_finance	Large	For profit, private	Full insurance coverage offered	About the same	No	Yes	Yes	40.0	5.0	30.0	30.
821	business_finance	Large	For profit, private	Partial insurance coverage offered	About the same	No	Yes	Yes	35.0	15.0	45.0	75.
822	business_finance	Large	Non-profit	Full insurance coverage offered	About the same	No	Yes	No	34.0	10.0	86.0	88.
...	...	...	...	...	...	...	...	...	...	...	...	
2604	trade_transport	Small	Non-profit	Full insurance coverage offered	About the same	No	Yes	No	NaN	NaN	NaN	NaI
2626	trade_transport	Small	For profit, private	Partial insurance coverage offered	Larger	Yes	No	No	NaN	NaN	NaN	NaI
2629	trade_transport	Small	For profit, public	Full insurance coverage offered	Larger	No	Yes	Yes	NaN	2.0	15.0	NaI
2631	trade_transport	Small	For profit, private	Partial insurance coverage offered	Larger	Yes	No	No	NaN	NaN	NaN	95.
1662	NaN	Large	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaI

2843 rows × 16 columns

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]

In [26]:
```
df2 = df2.drop(df2[df2['Industry'].isnull() == True].index, axis=0)
df2
```

Out[26]:

	Industry	Size	organization_type	insurance_coverage	premium_proportion	pt_insurace	health_education	WFH	under_30	over_60	female	hourl
465	business_finance	Large	For profit, public	Full insurance coverage offered	About the same	No	Yes	Yes	75.0	1.0	49.0	25.
2471	business_finance	Large	For profit, private	Partial insurance coverage offered	Larger	No	Yes	Yes	70.0	30.0	NaN	Nal
1352	business_finance	Large	For profit, private	Full insurance coverage offered	About the same	No	Yes	Yes	40.0	5.0	30.0	30.
821	business_finance	Large	For profit, private	Partial insurance coverage offered	About the same	No	Yes	Yes	35.0	15.0	45.0	75.
822	business_finance	Large	Non-profit	Full insurance coverage offered	About the same	No	Yes	No	34.0	10.0	86.0	88.
...	...	...	...	...	...	...	...	...	...	...	...	
2595	trade_transport	Small	For profit, private	Full insurance coverage offered	About the same	Yes	No	No	NaN	2.0	2.0	90.
2604	trade_transport	Small	Non-profit	Full insurance coverage offered	About the same	No	Yes	No	NaN	NaN	NaN	Nal
2626	trade_transport	Small	For profit, private	Partial insurance coverage offered	Larger	Yes	No	No	NaN	NaN	NaN	Nal
2629	trade_transport	Small	For profit, public	Full insurance coverage offered	Larger	No	Yes	Yes	NaN	2.0	15.0	Nal
2631	trade_transport	Small	For profit, private	Partial insurance coverage offered	Larger	Yes	No	No	NaN	NaN	NaN	95.

2842 rows × 16 columns

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]

In [27]:
```
df2['gender_balance'] = pd.cut(df2.female, bins=[-0.1,35,65,100], labels=("Mostly Men", "Balanced", "Mostly Women"))
df2.head()
```

Out[27]:

	Industry	Size	organization_type	insurance_coverage	premium_proportion	pt_insurace	health_education	WFH	under_30	over_60	female	hourl
465	business_finance	Large	For profit, public	Full insurance coverage offered	About the same	No	Yes	Yes	75.0	1.0	49.0	25.
2471	business_finance	Large	For profit, private	Partial insurance coverage offered	Larger	No	Yes	Yes	70.0	30.0	NaN	Nal
1352	business_finance	Large	For profit, private	Full insurance coverage offered	About the same	No	Yes	Yes	40.0	5.0	30.0	30.
821	business_finance	Large	For profit, private	Partial insurance coverage offered	About the same	No	Yes	Yes	35.0	15.0	45.0	75.
822	business_finance	Large	Non-profit	Full insurance coverage offered	About the same	No	Yes	No	34.0	10.0	86.0	88.

In [28]:
```
df2['gender_balance'].value_counts()
```

Out[28]:
```
gender_balance
Mostly Men      733
Mostly Women    727
Balanced        587
Name: count, dtype: int64
```

Problem 11

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

In [29]:
```
df2.columns
```

Out[29]:
```
Index(['Industry', 'Size', 'organization_type', 'insurance_coverage',
       'premium_proportion', 'pt_insurace', 'health_education', 'WFH',
       'under_30', 'over_60', 'female', 'hourly', 'atypical_shift', 'remote',
       'unionized', 'turnover', 'gender_balance'],
      dtype='object')
```

In [30]:
```
catcols = ['Industry', 'Size', 'organization_type', 'insurance_coverage',
       'premium_proportion', 'pt_insurace', 'health_education', 'WFH', 'gender_balance']
df2[catcols] = df2[catcols].astype('category')
df2.dtypes
```

Out[30]:
```
Industry              category
Size                  category
organization_type     category
insurance_coverage    category
premium_proportion    category
pt_insurace           category
health_education      category
WFH                   category
under_30               float64
over_60                float64
female                 float64
hourly                 float64
atypical_shift         float64
remote                 float64
unionized              float64
turnover               float64
gender_balance        category
dtype: object
```

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]

In [31]:
```
df2.query("Size == 'Small' & WFH == 'Yes'").stb.freq(['insurance_coverage'])
```

Out[31]:

	insurance_coverage	count	percent	cumulative_count	cumulative_percent
0	Full insurance coverage offered	324	46.285714	324	46.285714
1	Partial insurance coverage offered	310	44.285714	634	90.571429
2	No insurance coverage offered	66	9.428571	700	100.000000

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]

In [32]:
```
wha_db = sqlite3.connect("wha.db")
```

In [33]:
```
df2.to_sql('wha', wha_db, index=False, chunksize=1000, if_exists='replace')
```

Out[33]:
```
2842
```

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]

SQL Query

In [34]:
```
pd.read_sql_query('''
SELECT size,
organization_type,
insurance_coverage AS insurance,
female
    FROM wha
WHERE industry = 'hospital' AND premium_proportion='Smaller'
ORDER BY female DESC;''', wha_db)
```

Out[34]:

	Size	organization_type	insurance	female
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	Large	Non-profit	Partial insurance coverage offered	NaN
7	Medium	Non-profit	Full insurance coverage offered	NaN
8	Medium	None	Partial insurance coverage offered	NaN
9	Medium	Non-profit	Partial insurance coverage offered	NaN
10	Medium	Non-profit	Full insurance coverage offered	NaN

Pandas Selection

In [35]:
```
df2[['Industry', 'premium_proportion','Size','organization_type',
     'insurance_coverage','female']].query("Industry == 'hospital' & premium_proportion == 'Smaller'").sort_values(by = 'female', ascending = False
    .drop(['Industry','premium_proportion'], axis = 1).reset_index(drop=True)
```

Out[35]:

	Size	organization_type	insurance_coverage	female
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	Large	Non-profit	Partial insurance coverage offered	NaN
7	Medium	Non-profit	Full insurance coverage offered	NaN
8	Medium	NaN	Partial insurance coverage offered	NaN
9	Medium	Non-profit	Partial insurance coverage offered	NaN
10	Medium	Non-profit	Full insurance coverage offered	NaN

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]

SQL Query

In [36]:
```
pd.read_sql_query('''SELECT industry,
    AVG(female) as percent_female,
    AVG(under_30) as percent_under30,
    AVG(over_60) as percent_over60
FROM wha
GROUP BY industry
ORDER BY percent_female DESC;''', wha_db)
```

Out[36]:

	Industry	percent_female	percent_under30	percent_over60
0	health_education	80.657143	25.745665	11.349570
1	hospital	76.427027	27.213793	16.489655
2	entertainment_food	53.804416	38.566343	11.544872
3	business_finance	50.632184	23.821752	12.465465
4	public_admin	39.056738	21.015625	15.015385
5	trade_transport	32.657258	29.108696	12.584034
6	natural_resources	20.328605	22.257143	10.690355

Pandas Selection

In [37]:
```
df2[['Industry','female','under_30','over_60']].groupby('Industry').mean().sort_values(by = 'female', ascending = False).reset_index()
```

Out[37]:

	Industry	female	under_30	over_60
0	health_education	80.657143	25.745665	11.349570
1	hospital	76.427027	27.213793	16.489655
2	entertainment_food	53.804416	38.566343	11.544872
3	business_finance	50.632184	23.821752	12.465465
4	public_admin	39.056738	21.015625	15.015385
5	trade_transport	32.657258	29.108696	12.584034
6	natural_resources	20.328605	22.257143	10.690355

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]

SQL Query

In [38]:
```
pd.read_sql_query('''SELECT
gender_balance,
insurance_coverage,
COUNT(*)
    FROM wha
GROUP BY gender_balance, insurance_coverage
HAVING gender_balance is NOT NULL and insurance_coverage is NOT NULL;''', wha_db)
```

Out[38]:

	gender_balance	insurance_coverage	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

Pandas Selection

In [39]:
```
df2.groupby(['gender_balance','insurance_coverage']).size().reset_index()\
.rename({0 : 'COUNT(*)'}, axis=1)
```

Out[39]:

	gender_balance	insurance_coverage	COUNT(*)
0	Mostly Men	Full insurance coverage offered	301
1	Mostly Men	No insurance coverage offered	91
2	Mostly Men	Partial insurance coverage offered	332
3	Balanced	Full insurance coverage offered	226
4	Balanced	No insurance coverage offered	77
5	Balanced	Partial insurance coverage offered	271
6	Mostly Women	Full insurance coverage offered	267
7	Mostly Women	No insurance coverage offered	107
8	Mostly Women	Partial insurance coverage offered	333

In [ ]:
```

```