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]:
```
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 [2]:
```
wha_data = pd.read_csv("https://www.cdc.gov/workplacehealthpromotion/data-su
wha_data.iloc[[13,27,101],]
```

Out[2]:

	OC1	OC3	HI1	HI2	HI3	HI4	HRA1	HRA1A	HRA1B	HRA1E	...	WL3_05	E1_C
13	3	1.0	2.0	3.0	2.0	1.0	1.0	3.0	3.0	1.0	...	NaN	Na
27	1	3.0	1.0	3.0	1.0	1.0	1.0	2.0	4.0	2.0	...	NaN	Na
101	2	1.0	1.0	3.0	2.0	1.0	1.0	2.0	4.0	2.0	...	NaN	Na

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

Out[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
...	...	...	...	...	...	...	...	...	...	...	...	...
2838	6.0	5.0	4.0	1.0	3.0	1.0	1.0	99.0	999.0	999.0	999.0	999.0
2839	6.0	5.0	4.0	2.0	3.0	1.0	1.0	2.0	997.0	997.0	997.0	997.0
2840	6.0	8.0	4.0	2.0	3.0	1.0	1.0	1.0	27.0	997.0	61.0	997.0
2841	6.0	8.0	4.0	2.0	3.0	1.0	2.0	99.0	999.0	999.0	999.0	999.0
2842	6.0	8.0	4.0	2.0	3.0	1.0	1.0	1.0	25.0	10.0	35.0	90.0

2843 rows × 16 columns

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 [5]:
```
work_wha_data = work_wha_data.rename({
       'Industry': 'industry',
       'Size': 'size',
       'OC3': "for_profit_status",
       'HI1': "insurance_coverage",
       'HI2': "insurance_premium_proportion",
       'HI3': "part_time_insurance_coverage",
```

```
        'CP1': "health_education_programs",
        'WL6': "remote_work_offered",
        'WD1_1': "under30_percentage",
        'WD1_2': "over60_percentage",
        'WD2': "female_percentage",
        'WD3': "hourly_non_exempt_percentage",
        'WD4': "non_typical_shift_percentage",
        'WD5': "remote_work_percentage",
        'WD6': "unionized_percentage",
        'WD7': "turnover_percentage"}, axis = 1)
work_wha_data
```

Out[5]:

	industry	size	for_profit_status	insurance_coverage	insurance_premium_propc
0	7.0	7.0	3.0	2.0	
1	7.0	6.0	3.0	2.0	
2	7.0	8.0	3.0	1.0	
3	7.0	4.0	2.0	1.0	
4	7.0	4.0	3.0	1.0	
...	...	...	...	...	
2838	6.0	5.0	4.0	1.0	
2839	6.0	5.0	4.0	2.0	
2840	6.0	8.0	4.0	2.0	
2841	6.0	8.0	4.0	2.0	
2842	6.0	8.0	4.0	2.0	

2843 rows × 16 columns

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 [6]:

```
replace_map = {1.0: "Agriculture, Machinery, & Manufacturing",
               2.0: "Trade & Shipping",
               3.0: "Recreation & Services",
               4.0: "Finance & Technical Services",
               5.0: "Educational & Health Services",
               6.0: "Public Administration",
               7.0: "Hospital Worksites"}
work_wha_data["industry"] = work_wha_data["industry"].replace(replace_map)
work_wha_data["industry"].value_counts()
```

Out[6]:
```
industry
Educational & Health Services              551
Agriculture, Machinery, & Manufacturing    525
Recreation & Services                      433
Finance & Technical Services               429
Hospital Worksites                         338
Trade & Shipping                           311
Public Administration                      255
Name: count, dtype: int64
```

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 [7]:
```
replace_map = {1.0: "Small",
               2.0: "Small",
               3.0: "Small",
               4.0: "Medium",
               5.0: "Medium",
               6.0: "Large",
               7.0: "Large",
               8.0: "Large"}
work_wha_data["size"] = work_wha_data["size"].replace(replace_map)
work_wha_data["size"].value_counts()
```

Out[7]:
```
size
Small     2195
Medium     393
Large      254
Name: count, dtype: int64
```

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 [8]:
```
replace_map = {'for_profit_status': {1.0: "For profit, public",
                                     2.0: "For profit, private",
                                     3.0: "Non-profit",
                                     4.0: "State or local government",
                                     5.0: "Federal government",
                                     6.0: "Other",
                                     97.0: np.nan,
```

```
                                    98.0: np.nan,
                                    99.0: np.nan},
                "insurance_coverage": {1.0: "Full insurance coverage offered"
                                    2.0: "Partial insurance coverage offered
                                    3.0: "No insurance coverage offered",
                                    97.0: np.nan,
                                    98.0: np.nan,
                                    99.0: np.nan},
                "insurance_premium_proportion": {1.0: "Larger",
                                                 2.0: "Smaller",
                                                 3.0: "About the same",
                                                 96.0: np.nan,
                                                 97.0: np.nan,
                                                 98.0: np.nan,
                                                 99.0: np.nan},
                "part_time_insurance_coverage": {1.0: "Yes",
                                                 2.0: "No",
                                                 97.0: np.nan,
                                                 98.0: np.nan,
                                                 99.0: np.nan},
                "health_education_programs": {1.0: "Yes",
                                              2.0: "No",
                                              97.0: np.nan,
                                              98.0: np.nan},
                "remote_work_offered": {1.0: "Yes",
                                    2.0: "No",
                                    97.0: np.nan,
                                    98.0: np.nan,
                                    99.0: np.nan}}

work_wha_data = work_wha_data.replace(replace_map)
work_wha_data
```

Out[8]:

	industry	size	for_profit_status	insurance_coverage	insurance_premiu
0	Hospital Worksites	Large	Non-profit	Partial insurance coverage offered	
1	Hospital Worksites	Large	Non-profit	Partial insurance coverage offered	A
2	Hospital Worksites	Large	Non-profit	Full insurance coverage offered	A
3	Hospital Worksites	Medium	For profit, private	Full insurance coverage offered	
4	Hospital Worksites	Medium	Non-profit	Full insurance coverage offered	A
...	...	...	...	...	
2838	Public Administration	Medium	State or local government	Full insurance coverage offered	A
2839	Public Administration	Medium	State or local government	Partial insurance coverage offered	A
2840	Public Administration	Large	State or local government	Partial insurance coverage offered	A
2841	Public Administration	Large	State or local government	Partial insurance coverage offered	A
2842	Public Administration	Large	State or local government	Partial insurance coverage offered	A

2843 rows × 16 columns

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 [9]:
```
replace_map = {"under30_percentage": {997.0: np.nan,
                                      998.0: np.nan,
                                      999.0: np.nan},
               "over60_percentage": {997.0: np.nan,
                                     998.0: np.nan,
                                     999.0: np.nan},
               "female_percentage": {997.0: np.nan,
                                     998.0: np.nan,
                                     999.0: np.nan},
               "hourly_non_exempt_percentage": {997.0: np.nan,
                                                998.0: np.nan,
                                                999.0: np.nan},
               "non_typical_shift_percentage": {997.0: np.nan,
```

```
                                                998.0: np.nan,
                                                999.0: np.nan},
                        "remote_work_percentage": {997.0: np.nan,
                                                   998.0: np.nan,
                                                   999.0: np.nan},
                        "unionized_percentage": {997.0: np.nan,
                                                 998.0: np.nan,
                                                 999.0: np.nan},
                        "turnover_percentage": {997.0: np.nan,
                                                998.0: np.nan,
                                                999.0: np.nan}}

work_wha_data = work_wha_data.replace(replace_map)
work_wha_data
```

Out[9]:

	industry	size	for_profit_status	insurance_coverage	insurance_premiu
0	Hospital Worksites	Large	Non-profit	Partial insurance coverage offered	
1	Hospital Worksites	Large	Non-profit	Partial insurance coverage offered	A
2	Hospital Worksites	Large	Non-profit	Full insurance coverage offered	A
3	Hospital Worksites	Medium	For profit, private	Full insurance coverage offered	
4	Hospital Worksites	Medium	Non-profit	Full insurance coverage offered	A
...	...	...	...	...	
2838	Public Administration	Medium	State or local government	Full insurance coverage offered	A
2839	Public Administration	Medium	State or local government	Partial insurance coverage offered	A
2840	Public Administration	Large	State or local government	Partial insurance coverage offered	A
2841	Public Administration	Large	State or local government	Partial insurance coverage offered	A
2842	Public Administration	Large	State or local government	Partial insurance coverage offered	A

2843 rows × 16 columns

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 [10]:
```
work_wha_data = work_wha_data.sort_values(["industry","size","under30_percen
work_wha_data
```

Out[10]:

	industry	size	for_profit_status	insurance_coverage	insurance_premium
0	Agriculture, Machinery, & Manufacturing	Large	For profit, private	Partial insurance coverage offered	Abo
1	Agriculture, Machinery, & Manufacturing	Large	For profit, private	Partial insurance coverage offered	Abo
2	Agriculture, Machinery, & Manufacturing	Large	For profit, private	Partial insurance coverage offered	
3	Agriculture, Machinery, & Manufacturing	Large	For profit, private	Full insurance coverage offered	Abo
4	Agriculture, Machinery, & Manufacturing	Large	For profit, private	Full insurance coverage offered	Abo
...	...	...	...	...	
2838	Trade & Shipping	Small	Non-profit	Full insurance coverage offered	Abo
2839	Trade & Shipping	Small	For profit, private	Partial insurance coverage offered	
2840	Trade & Shipping	Small	For profit, public	Full insurance coverage offered	
2841	Trade & Shipping	Small	For profit, private	Partial insurance coverage offered	
2842	NaN	NaN	NaN	NaN	

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 [11]:
```
work_wha_data = work_wha_data[work_wha_data['industry'].notna()]
work_wha_data
```

Out[11]:

	industry	size	for_profit_status	insurance_coverage	insurance_premium
0	Agriculture, Machinery, & Manufacturing	Large	For profit, private	Partial insurance coverage offered	Abo
1	Agriculture, Machinery, & Manufacturing	Large	For profit, private	Partial insurance coverage offered	Abo
2	Agriculture, Machinery, & Manufacturing	Large	For profit, private	Partial insurance coverage offered	
3	Agriculture, Machinery, & Manufacturing	Large	For profit, private	Full insurance coverage offered	Abo
4	Agriculture, Machinery, & Manufacturing	Large	For profit, private	Full insurance coverage offered	Abo
...	...	...	...	...	
2837	Trade & Shipping	Small	For profit, private	Full insurance coverage offered	Abo
2838	Trade & Shipping	Small	Non-profit	Full insurance coverage offered	Abo
2839	Trade & Shipping	Small	For profit, private	Partial insurance coverage offered	
2840	Trade & Shipping	Small	For profit, public	Full insurance coverage offered	
2841	Trade & Shipping	Small	For profit, private	Partial insurance coverage offered	

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 [12]:
```
work_wha_data["gender_balance"] = pd.cut(work_wha_data["female_percentage"],
work_wha_data
```

Out[12]:

	industry	size	for_profit_status	insurance_coverage	insurance_premium
0	Agriculture, Machinery, & Manufacturing	Large	For profit, private	Partial insurance coverage offered	Abo
1	Agriculture, Machinery, & Manufacturing	Large	For profit, private	Partial insurance coverage offered	Abo
2	Agriculture, Machinery, & Manufacturing	Large	For profit, private	Partial insurance coverage offered	
3	Agriculture, Machinery, & Manufacturing	Large	For profit, private	Full insurance coverage offered	Abo
4	Agriculture, Machinery, & Manufacturing	Large	For profit, private	Full insurance coverage offered	Abo
...	...	...	...	...	
2837	Trade & Shipping	Small	For profit, private	Full insurance coverage offered	Abo
2838	Trade & Shipping	Small	Non-profit	Full insurance coverage offered	Abo
2839	Trade & Shipping	Small	For profit, private	Partial insurance coverage offered	
2840	Trade & Shipping	Small	For profit, public	Full insurance coverage offered	
2841	Trade & Shipping	Small	For profit, private	Partial insurance coverage offered	

2842 rows × 17 columns

Problem 11

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

In [13]:
```
cols = ["industry",
        "size",
        "for_profit_status",
        "insurance_coverage",
        "insurance_premium_proportion",
        "part_time_insurance_coverage",
        "health_education_programs",
        "remote_work_offered"]
work_wha_data[cols] = work_wha_data[cols].astype('category')
work_wha_data.dtypes
```

Out[13]:
```
industry                          category
size                              category
for_profit_status                 category
insurance_coverage                category
insurance_premium_proportion      category
part_time_insurance_coverage      category
health_education_programs         category
remote_work_offered               category
under30_percentage                 float64
over60_percentage                  float64
female_percentage                  float64
hourly_non_exempt_percentage       float64
non_typical_shift_percentage       float64
remote_work_percentage             float64
unionized_percentage               float64
turnover_percentage                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 [14]:
```
work_wha_data.query("size == 'Small' & remote_work_offered == 'Yes'").stb.fr
```

Out[14]:

	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 [15]:
```
engine = sqlite3.connect("lab8.db")
work_wha_data.to_sql('work_wha_data', con=engine, chunksize=1000, if_exists=
```

Out[15]: 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]

In [16]:
```
work_wha_data.query(
        "industry == 'Hospital Worksites' & insurance_premium_proportion =='
    )[["size", "for_profit_status", "insurance_coverage", "female_percentage
    ].sort_values(by = "female_percentage", ascending=False
    ).rename({"insurance_coverage": "insurance"}, axis=1).reset_index(drop=T
```

Out[16]:

	size	for_profit_status	insurance	female_percentage
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

In [20]:
```
myquery = '''
    SELECT size, for_profit_status, insurance_coverage AS insurance, female_
WHERE industry = 'Hospital Worksites' AND insurance_premium_proportion ='Sma
ORDER BY female_percentage DESC;
'''
pd.read_sql_query(myquery, con=engine)
```

Out[20]:

	size	for_profit_status	insurance	female_percentage
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

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]

In [21]:
```
work_wha_data.groupby('industry').agg({'female_percentage':'mean',
                                       'under30_percentage':'mean',
                                       'over60_percentage':'mean'}
                                      ).sort_values(by = "female_percentage", asce
                                      ).rename({'female_percentage':'percent_femal
                                       'under30_percentage':'percent_under30',
                                       'over60_percentage':'percent_over60'}, ax
```

Out[21]:

	industry	percent_female	percent_under30	percent_over60
0	Educational & Health Services	80.657143	25.745665	11.349570
1	Hospital Worksites	76.427027	27.213793	16.489655
2	Recreation & Services	53.804416	38.566343	11.544872
3	Finance & Technical Services	50.632184	23.821752	12.465465
4	Public Administration	39.056738	21.015625	15.015385
5	Trade & Shipping	32.657258	29.108696	12.584034
6	Agriculture, Machinery, & Manufacturing	20.328605	22.257143	10.690355

In [22]:
```
myquery = '''
SELECT Industry,
    AVG(female_percentage) as percent_female,
    AVG(under30_percentage) as percent_under30,
    AVG(over60_percentage) as percent_over60
FROM work_wha_data
GROUP BY industry
ORDER BY percent_female DESC;
'''
pd.read_sql_query(myquery, con=engine)
```

Out[22]:

	industry	percent_female	percent_under30	percent_over60
0	Educational & Health Services	80.657143	25.745665	11.349570
1	Hospital Worksites	76.427027	27.213793	16.489655
2	Recreation & Services	53.804416	38.566343	11.544872
3	Finance & Technical Services	50.632184	23.821752	12.465465
4	Public Administration	39.056738	21.015625	15.015385
5	Trade & Shipping	32.657258	29.108696	12.584034
6	Agriculture, Machinery, & Manufacturing	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]

In [23]:
```
pd.DataFrame(work_wha_data.groupby(['gender_balance','insurance_coverage']).
            ).rename({0: "COUNT(*)"}, axis=1).reset_index()
```

Out[23]:

	gender_balance	insurance_coverage	COUNT(*)
0	Mostly men	Full insurance coverage offered	293
1	Mostly men	No insurance coverage offered	87
2	Mostly men	Partial insurance coverage offered	321
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 [24]:
```
myquery = '''
SELECT gender_balance, insurance_coverage, COUNT(*)
FROM work_wha_data
GROUP BY gender_balance, insurance_coverage
HAVING gender_balance is NOT NULL and insurance_coverage is NOT NULL;
'''
pd.read_sql_query(myquery, con=engine)
```

Out[24]:

	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	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

In [25]:
```
engine.commit()
engine.close()
```

In [ ]: