

DVA_lab-4_mahalakshmi18

March 20, 2021

0.0.1 Lab4. Pandas Grouping and Aggregation

0.0.2 Import necessary modules

```
[3]: import pandas as pd
import csv
```

```
[4]: data = pd.read_csv("thanksgiving-2015-poll-data.csv", encoding="Latin-1")
data
```

```
[4]: RespondentID Do you celebrate Thanksgiving? \
0      4337954960      Yes
1      4337951949      Yes
2      4337935621      Yes
3      4337933040      Yes
4      4337931983      Yes
...      ...      ...
1053    4335944082      Yes
1054    4335943173      Yes
1055    4335943060      Yes
1056    4335934708      Yes
1057    4335894916      Yes

      What is typically the main dish at your Thanksgiving dinner? \
0      Turkey
1      Turkey
2      Turkey
3      Turkey
4      Tofurkey
...      ...
1053    Turkey
1054    Turkey
1055    Other (please specify)
1056    Turkey
1057    Turkey
```

```
      What is typically the main dish at your Thanksgiving dinner? - Other
(please specify) \
```

0	NaN
1	NaN
2	NaN
3	NaN
4	NaN
...	...
1053	NaN
1054	NaN
1055	Duck
1056	NaN
1057	NaN

How is the main dish typically cooked? \	
0	Baked
1	Baked
2	Roasted
3	Baked
4	Baked
...	...
1053	Roasted
1054	Baked
1055	Baked
1056	Baked
1057	Baked

How is the main dish typically cooked? - Other (please specify) \	
0	NaN
1	NaN
2	NaN
3	NaN
4	NaN
...	...
1053	NaN
1054	NaN
1055	NaN
1056	NaN
1057	NaN

What kind of stuffing/dressing do you typically have? \	
0	Bread-based
1	Bread-based
2	Rice-based
3	Bread-based
4	Bread-based
...	...
1053	Bread-based
1054	Bread-based

1055	Rice-based
1056	None
1057	Bread-based

What kind of stuffing/dressing do you typically have? - Other (please specify) \

0	NaN
1	NaN
2	NaN
3	NaN
4	NaN
...	...
1053	NaN
1054	NaN
1055	NaN
1056	NaN
1057	NaN

What type of cranberry saucedo you typically have? \

0	None
1	Other (please specify)
2	Homemade
3	Homemade
4	Canned
...	...
1053	Homemade
1054	Canned
1055	None
1056	Homemade
1057	Canned

What type of cranberry saucedo you typically have? - Other (please specify)

\	
0	NaN
1	Homemade cranberry gelatin ring
2	NaN
3	NaN
4	NaN
...	...
1053	NaN
1054	NaN
1055	NaN
1056	NaN
1057	NaN

...	\
0	...

1 ...
 2 ...
 3 ...
 4 ...

 1053 ...
 1054 ...
 1055 ...
 1056 ...
 1057 ...

Have you ever tried to meet up with hometown friends on Thanksgiving night?

\
 0 Yes
 1 No
 2 Yes
 3 Yes
 4 Yes

 1053 Yes
 1054 No
 1055 Yes
 1056 Yes
 1057 Yes

Have you ever attended a "Friendsgiving?" \

0 No
 1 No
 2 Yes
 3 No
 4 No

 1053 Yes
 1054 No
 1055 Yes
 1056 No
 1057 Yes

Will you shop any Black Friday sales on Thanksgiving Day? \

0 No
 1 Yes
 2 Yes
 3 No
 4 No

 1053 No
 1054 No

1055	Yes
1056	Yes
1057	Yes

Do you work in retail? Will your employer make you work on Black Friday? \		
0	No	NaN
1	No	NaN
2	No	NaN
3	No	NaN
4	No	NaN
...
1053	No	NaN
1054	No	NaN
1055	No	NaN
1056	Yes	Yes
1057	No	NaN

How would you describe where you live? Age What is your gender? \			
0	Suburban	18 - 29	Male
1	Rural	18 - 29	Female
2	Suburban	18 - 29	Male
3	Urban	30 - 44	Male
4	Urban	30 - 44	Male
...
1053	Rural	30 - 44	Female
1054	Suburban	60+	Female
1055	Urban	60+	Male
1056	NaN	NaN	NaN
1057	NaN	NaN	NaN

How much total combined money did all members of your HOUSEHOLD earn last year? \	
0	\$75,000 to \$99,999
1	\$50,000 to \$74,999
2	\$0 to \$9,999
3	\$200,000 and up
4	\$100,000 to \$124,999
...	...
1053	\$100,000 to \$124,999
1054	\$50,000 to \$74,999
1055	\$100,000 to \$124,999
1056	NaN
1057	NaN

US Region	
0	Middle Atlantic
1	East South Central

```

2           Mountain
3           Pacific
4           Pacific
...
1053        Mountain
1054        Pacific
1055        Pacific
1056        NaN
1057        NaN

```

```
[1058 rows x 65 columns]
```

```
[5]: data.head() # Print top 5 rows from data
```

```

[5]: RespondentID Do you celebrate Thanksgiving? \
0      4337954960      Yes
1      4337951949      Yes
2      4337935621      Yes
3      4337933040      Yes
4      4337931983      Yes

```

```

What is typically the main dish at your Thanksgiving dinner? \
0      Turkey
1      Turkey
2      Turkey
3      Turkey
4      Tofurkey

```

```

What is typically the main dish at your Thanksgiving dinner? - Other (please
specify) \
0      NaN
1      NaN
2      NaN
3      NaN
4      NaN

```

```

How is the main dish typically cooked? \
0      Baked
1      Baked
2      Roasted
3      Baked
4      Baked

```

```

How is the main dish typically cooked? - Other (please specify) \
0      NaN
1      NaN
2      NaN

```

3	NaN
4	NaN

	What kind of stuffing/dressing do you typically have? \
0	Bread-based
1	Bread-based
2	Rice-based
3	Bread-based
4	Bread-based

	What kind of stuffing/dressing do you typically have? - Other (please specify)
\	
0	NaN
1	NaN
2	NaN
3	NaN
4	NaN

	What type of cranberry saucedo you typically have? \
0	None
1	Other (please specify)
2	Homemade
3	Homemade
4	Canned

	What type of cranberry saucedo you typically have? - Other (please specify) \
0	NaN
1	Homemade cranberry gelatin ring
2	NaN
3	NaN
4	NaN

	...
\	
0	...
1	...
2	...
3	...
4	...

	Have you ever tried to meet up with hometown friends on Thanksgiving night? \
0	Yes
1	No
2	Yes
3	Yes
4	Yes

	Have you ever attended a "Friendsgiving?" \
--	---

0	No
1	No
2	Yes
3	No
4	No

	Will you shop any Black Friday sales on Thanksgiving Day? \
0	No
1	Yes
2	Yes
3	No
4	No

	Do you work in retail? Will you employer make you work on Black Friday? \	
0	No	NaN
1	No	NaN
2	No	NaN
3	No	NaN
4	No	NaN

	How would you describe where you live?	Age	What is your gender? \
0	Suburban	18 - 29	Male
1	Rural	18 - 29	Female
2	Suburban	18 - 29	Male
3	Urban	30 - 44	Male
4	Urban	30 - 44	Male

	How much total combined money did all members of your HOUSEHOLD earn last year? \
0	\$75,000 to \$99,999
1	\$50,000 to \$74,999
2	\$0 to \$9,999
3	\$200,000 and up
4	\$100,000 to \$124,999

	US Region
0	Middle Atlantic
1	East South Central
2	Mountain
3	Pacific
4	Pacific

[5 rows x 65 columns]

```
[6]: data.shape      # what is the size?
```

```
[6]: (1058, 65)
```


As you can see above, the data has 65 columns of mostly categorical data. For example, the first column appears to allow for Yes and No responses only. Let's verify by using the `pandas.Series.unique` method to see what unique values are in the Do you celebrate Thanksgiving? column of data.

0.0.3 What are unique values of “Do you celebrate Thanksgiving?” column?

```
[7]: uniqueValues = data['Do you celebrate Thanksgiving?'].unique()
uniqueValues
```

```
[7]: array(['Yes', 'No'], dtype=object)
```

0.0.4 View all column names (top 5)

```
[8]: data.columns[:6]
```

```
[8]: Index(['RespondentID', 'Do you celebrate Thanksgiving?',
        'What is typically the main dish at your Thanksgiving dinner?',
        'What is typically the main dish at your Thanksgiving dinner? - Other
        (please specify)',
        'How is the main dish typically cooked?',
        'How is the main dish typically cooked? - Other (please specify)'],
        dtype='object')
```

0.0.5 Apply function to Series

0.0.6 DATA CLEANING - Now, let us transform gender to numeric value.

We'll assign 0 to Male, and 1 to Female. Before we dive into transforming the values, let's confirm that the values in the column are either Male or Female. We can use the `pandas.Series.value_counts` method to help us with this. We'll pass the `dropna=False` keyword argument to also count missing values.

0.0.7 How many male, female and NaN in “What is your gender?” column

```
[9]: data["What is your gender?"].value_counts(dropna=False)
```

```
[9]: Female    544
     Male     481
     NaN       33
     Name: What is your gender?, dtype: int64
```

Yes, they are female, male or nan

0.0.8 Let apply a user defined function to each value in the What is your gender? column to transform Male to 0 and female to 1

```
[10]: import math
def gender_code(gender_string):
    if isinstance(gender_string, float) and math.isnan(gender_string):
        return gender_string
    return int(gender_string == "Female")
```

0.0.9 Apply gender_code() to What is your gender? column

Let us apply this function to every row of What is your gender? column. It is something like automatic looping. Create a new column 'gender' and put it there

```
[11]: data["gender"] = data["What is your gender?"].apply(gender_code)
data["gender"]
```

```
[11]: 0      0.0
      1      1.0
      2      0.0
      3      0.0
      4      0.0
      ...
     1053     1.0
     1054     1.0
     1055     0.0
     1056     NaN
     1057     NaN
      Name: gender, Length: 1058, dtype: float64
```

0.0.10 Now, count male and females as 0s and 1s. How many in "gender" column?

```
[12]: data["gender"] = data["What is your gender?"].apply(gender_code)
data["gender"].value_counts(dropna=False)
```

```
[12]: 1.0      544
      0.0      481
      NaN       33
      Name: gender, dtype: int64
```

0.0.11 Applying functions to DataFrames

The apply method will work across each column in the DataFrame. If we pass the axis=1 keyword argument, it will work across each row.

0.0.12 Check the data type of each column in data using a lambda function. Just visualize data types of first 5 columns

```
[13]: data.apply(lambda x: x.dtype).head()
```

```
[13]: RespondentID
      int64
      Do you celebrate Thanksgiving?
      object
      What is typically the main dish at your Thanksgiving dinner?
      object
      What is typically the main dish at your Thanksgiving dinner? - Other (please
      specify)    object
      How is the main dish typically cooked?
      object
      dtype: object
```

0.0.13 DATA CLEANING - Let us clean up Income column

We need to convert string values representing income in “How much total combined money did all members of your HOUSEHOLD earn last year” column into numeric values. Check the unique values first

```
[14]: data["How much total combined money did all members of your HOUSEHOLD earn_
      ↳last year?"].value_counts(dropna=False)
```

```
[14]: $25,000 to $49,999      180
      Prefer not to answer   136
      $50,000 to $74,999     135
      $75,000 to $99,999     133
      $100,000 to $124,999   111
      $200,000 and up        80
      $10,000 to $24,999     68
      $0 to $9,999          66
      $125,000 to $149,999   49
      $150,000 to $174,999   40
      NaN                   33
      $175,000 to $199,999   27
      Name: How much total combined money did all members of your HOUSEHOLD earn last
      year?, dtype: int64
```

Looking at this, there are 4 different patterns for the values in the column: X to Y — an example is 25, 000 to 49,999. We can convert this to a numeric value by extracting the numbers and averaging them. NaN We’ll preserve NaN values, and not convert them at all. X and up — an example is \$200,000 and up. We can convert this to a numeric value by extracting the number. Prefer not to answer We’ll turn this into an NaN value.

```
[15]: import numpy as np
      def clean_income(value):
```

```

if value == "$200,000 and up":
    return 200000
elif value == "Prefer not to answer":
    return np.nan
elif isinstance(value, float) and math.isnan(value):
    return np.nan

value = value.replace("$", "").replace(",", "")
income_high, income_low = value.split(" to ")

return (int(income_high) + int(income_low)) / 2

```

0.0.14 Now apply this function to the “How much total combined money did all members of your HOUSEHOLD earn last year?” column and put it in new column “income”

```

[16]: data["income"] = data["How much total combined money did all members of your_
    ↳HOUSEHOLD earn last year?"].apply(clean_income)
data["income"].head()

```

```

[16]: 0      87499.5
      1      62499.5
      2       4999.5
      3    200000.0
      4    112499.5
      Name: income, dtype: float64

```

0.0.15 Grouping Data with Pandas

0.0.16 Who earn more income?

Suppose, we want to find who earn more income?. Is it People eating homemade sauce or people eating canned sauce during the Thanksgiving Day?

0.0.17 Check unique values in column, “What type of cranberry saucedo you typically have?” first.

```

[17]: data["What type of cranberry saucedo you typically have?"].value_counts()

```

```

[17]: Canned          502
      Homemade       301
      None           146
      Other (please specify)  25
      Name: What type of cranberry saucedo you typically have?, dtype: int64

```

We can now filter data to get two DataFrames, namely, `homemade_df` & `canned_df`, that only contain rows where the What type of cranberry saucedo you typically have?

is Canned or Homemade, respectively

0.0.18 Create a dataframe by filtering values “Homemade”

```
[18]: homemade_df = data[data["What type of cranberry saucedo you typically have?"]_
    ↪ == "Homemade"]
```

0.0.19 Create another dataframe by filtering values “Canned”

```
[19]: canned_df = data[data["What type of cranberry saucedo you typically have?"] ==_
    ↪ "Canned"]
```

0.0.20 Now print mean income of homemade_df and canned_df for these two groups of people

```
[21]: print(homemade_df["income"].mean())
    print(canned_df["income"].mean())
```

```
94878.1072874494
83823.40340909091
```

Conclusion: Wow, great. We can understand from these values that people who eat home made cranberry sauce earn more income that the other group.

0.0.21 Use groupby() and aggregate() to find out “Who earn more income?”

Split dataset based on “What type of cranberry saucedo you typically have?” column automatically into groups based on unique values

```
[22]: grouped = data.groupby("What type of cranberry saucedo you typically have?")
    grouped
```

```
[22]: <pandas.core.groupby.generic.DataFrameGroupBy object at 0x0000018A314234F0>
```

0.0.22 List out all groups that are created by groupby()

```
[29]: grouped.groups
```

```
[29]: {'Canned': [4, 6, 8, 11, 12, 15, 18, 19, 26, 27, 38, 43, 48, 53, 58, 59, 60, 68,
    69, 71, 74, 76, 79, 80, 86, 87, 89, 90, 91, 97, 103, 106, 107, 109, 115, 116,
    118, 119, 123, 127, 129, 130, 132, 135, 136, 137, 140, 141, 143, 144, 145, 150,
    153, 155, 156, 157, 158, 159, 161, 162, 163, 166, 167, 168, 169, 173, 179, 180,
    181, 182, 184, 186, 190, 192, 193, 195, 198, 199, 200, 204, 205, 207, 209, 210,
    211, 212, 213, 215, 217, 218, 220, 222, 224, 226, 229, 230, 231, 239, 243, 245,
    ...], 'Homemade': [2, 3, 5, 7, 13, 14, 16, 20, 21, 23, 25, 28, 30, 32, 33, 37,
    39, 42, 44, 46, 52, 54, 56, 57, 62, 64, 66, 70, 82, 83, 85, 88, 93, 94, 96, 98,
    101, 102, 108, 110, 111, 112, 114, 120, 122, 128, 134, 138, 139, 152, 165, 171,
    172, 174, 175, 176, 177, 178, 183, 188, 189, 194, 201, 202, 203, 208, 219, 223,
    225, 232, 234, 235, 236, 238, 241, 242, 244, 246, 248, 254, 255, 256, 259, 261,
```

```
262, 263, 264, 268, 281, 285, 286, 287, 290, 291, 292, 295, 298, 300, 302, 303,
...], 'None': [0, 17, 24, 29, 34, 36, 40, 47, 49, 51, 55, 61, 67, 72, 73, 77,
78, 81, 92, 99, 100, 104, 105, 117, 121, 124, 126, 131, 133, 142, 146, 148, 149,
160, 164, 185, 187, 191, 197, 227, 228, 237, 240, 274, 275, 319, 321, 329, 337,
362, 370, 377, 391, 395, 406, 409, 414, 417, 421, 437, 439, 466, 480, 491, 492,
495, 505, 514, 526, 529, 532, 537, 540, 553, 560, 564, 571, 573, 580, 584, 591,
594, 598, 602, 605, 606, 609, 610, 618, 626, 631, 639, 647, 658, 672, 673, 684,
700, 701, 716, ...], 'Other (please specify)': [1, 9, 154, 216, 221, 233, 249,
265, 301, 336, 380, 435, 444, 447, 513, 550, 749, 750, 784, 807, 860, 872, 905,
1000, 1007]}
```

```
[30]: grouped.size()
```

```
[30]: What type of cranberry saucedo you typically have?
```

```
Canned          502
Homemade         301
None             146
Other (please specify)  25
dtype: int64
```

```
[31]: for name, group in grouped:
        print(name)
        print(group.shape)
        print(type(group))
```

```
Canned
(502, 67)
<class 'pandas.core.frame.DataFrame'>
Homemade
(301, 67)
<class 'pandas.core.frame.DataFrame'>
None
(146, 67)
<class 'pandas.core.frame.DataFrame'>
Other (please specify)
(25, 67)
<class 'pandas.core.frame.DataFrame'>
```

Here each group is a `DataFrame`, and you can use any normal `DataFrame` methods on it. We can also extract a single column from a group. This will allow us to perform further computations just on that specific column:

```
[32]: grouped["income"]
```

```
[32]: <pandas.core.groupby.generic.SeriesGroupBy object at 0x0000018A314697C0>
```

```
[33]: grouped["income"].size()
```

[33]: What type of cranberry saucedo you typically have?

```
Canned          502
Homemade        301
None            146
Other (please specify)  25
Name: income, dtype: int64
```

0.0.23 Aggregating values in groups

Splitting data into groups will not be sufficient. Real power comes when we can apply computation on each group.

0.0.24 Now, find out average income

We could find the average income for people who served each type of cranberry sauce. Extract income column from grouped DF and find mean value for each group

```
[34]: grouped["income"].agg(np.mean)
```

[34]: What type of cranberry saucedo you typically have?

```
Canned          83823.403409
Homemade        94878.107287
None            78886.084034
Other (please specify)  86629.978261
Name: income, dtype: float64
```

0.0.25 If you want to consider all numeric attributes and find the mean for each group for every column in data, you can do as below.

```
[35]: grouped.agg(np.mean)
```

```
[35]:
```

	RespondentID	gender \
What type of cranberry saucedo you typically have?		
Canned	4.336699e+09	0.552846
Homemade	4.336792e+09	0.533101
None	4.336765e+09	0.517483
Other (please specify)	4.336763e+09	0.640000

	income
What type of cranberry saucedo you typically have?	
Canned	83823.403409
Homemade	94878.107287
None	78886.084034
Other (please specify)	86629.978261

0.0.26 Plotting the results of aggregation

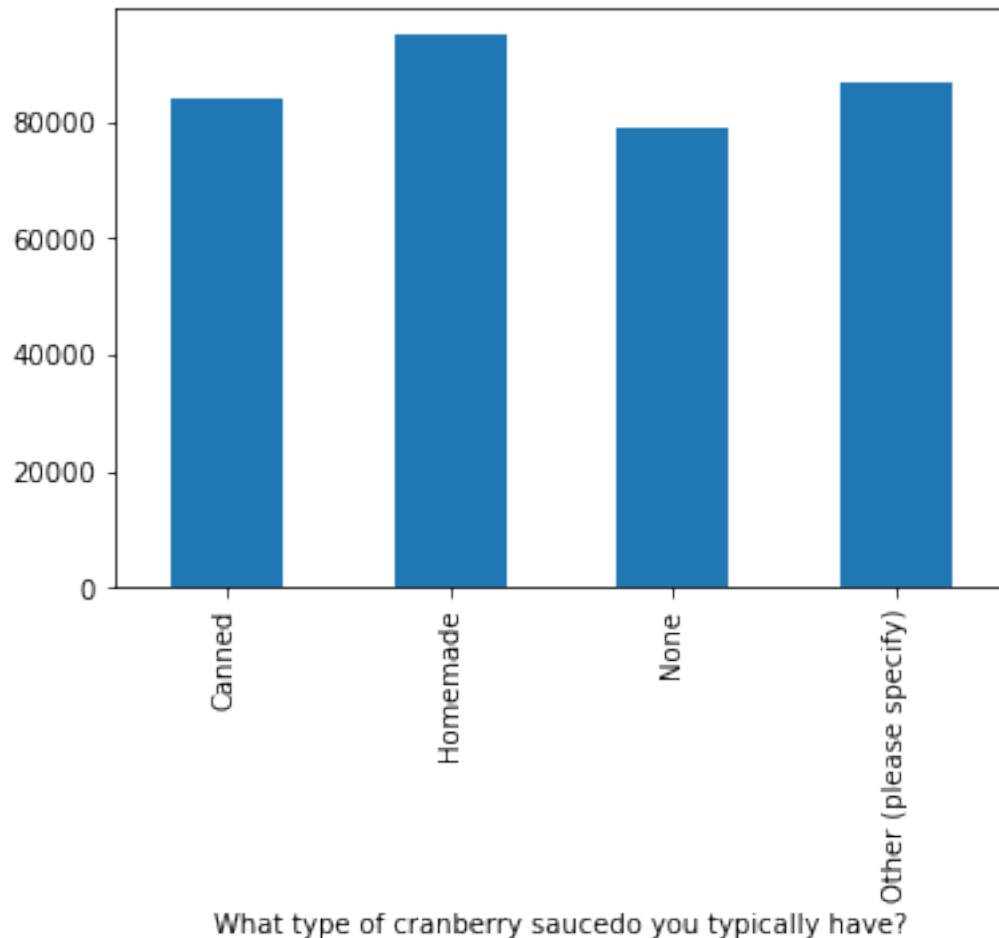
What is the average income of each category?

```
[36]: %matplotlib inline

sauce = grouped.agg(np.mean)
sauce["income"].plot(kind="bar")
```

```
[36]: <AxesSubplot:xlabel='What type of cranberry saucedo you typically have?'

```



0.0.27 Find the average income of people who eat Homemade cranberry sauce and Tofurkey

We need to apply groupby on two columns “What type of cranberry saucedo you typically have?” and “What is typically the main dish at your Thanksgiving dinner?”

```
[37]: grouped = data.groupby(["What type of cranberry saucedo you typically have?",
    ↪ "What is typically the main dish at your Thanksgiving dinner?"])
grouped.agg(np.mean)
```


[37]:

RespondentID \

What type of cranberry saucedo you typically have? What is typically the main dish at your Thanksg..

Canned	Chicken
4.336354e+09	
	Ham/Pork
4.336757e+09	
	I don't know
4.335987e+09	
	Other (please specify)
4.336682e+09	
	Roast beef
4.336254e+09	
	Tofurkey
4.337157e+09	
	Turkey
4.336705e+09	
Homemade	Chicken
4.336540e+09	
	Ham/Pork
4.337253e+09	
	I don't know
4.336084e+09	
	Other (please specify)
4.336863e+09	
	Roast beef
4.336174e+09	
	Tofurkey
4.336790e+09	
	Turducken
4.337475e+09	
	Turkey
4.336791e+09	
None	Chicken
4.336151e+09	
	Ham/Pork
4.336680e+09	
	I don't know
4.336412e+09	
	Other (please specify)
4.336688e+09	
	Roast beef
4.337424e+09	
	Tofurkey
4.336950e+09	
	Turducken
4.336739e+09	

4.336784e+09	Turkey
Other (please specify)	Ham/Pork
4.336465e+09	Other (please specify)
4.337335e+09	Tofurkey
4.336122e+09	Turkey
4.336724e+09	
gender \	
What type of cranberry saucedo you typically have? What is typically the main dish at your Thanksg..	
Canned	Chicken
0.333333	Ham/Pork
0.642857	I don't know
0.000000	Other (please specify)
1.000000	Roast beef
0.571429	Tofurkey
0.714286	Turkey
0.544444	Chicken
Homemade	Ham/Pork
0.750000	I don't know
0.250000	Other (please specify)
1.000000	Roast beef
0.600000	Tofurkey
0.000000	Turducken
0.666667	Turkey
0.500000	Chicken
0.531008	Ham/Pork
None	
0.500000	

0.444444	I don't know
0.500000	Other (please specify)
0.600000	Roast beef
0.000000	Tofurkey
0.500000	Turducken
0.000000	Turkey
0.523364	Ham/Pork
Other (please specify)	Other (please specify)
1.000000	Tofurkey
0.000000	Turkey
1.000000	
0.700000	

income

What type of cranberry saucedo you typically have? What is typically the main dish at your Thanksg...

Canned	Chicken
80999.600000	Ham/Pork
77499.535714	I don't know
4999.500000	Other (please specify)
53213.785714	Roast beef
25499.500000	Tofurkey
100713.857143	Turkey
85242.682045	Chicken
Homemade	Ham/Pork
19999.500000	I don't know
96874.625000	Other (please specify)
NaN	
55356.642857	

33749.500000	Roast beef
57916.166667	Tofurkey
200000.000000	Turducken
97690.147982	Turkey
None	Chicken
11249.500000	Ham/Pork
61249.500000	I don't know
33749.500000	Other (please specify)
119106.678571	Roast beef
162499.500000	Tofurkey
112499.500000	Turducken
NaN	Turkey
74606.275281	Ham/Pork
Other (please specify)	Other (please specify)
87499.500000	Tofurkey
124999.666667	Turkey
37499.500000	
82916.194444	

As you can see above, we get a nice table that shows us the mean of each column for each group. This enables us to find some interesting patterns, such as:

- People who have Turducken and Homemade cranberry sauce seem to have high household incomes.
- People who eat Canned cranberry sauce tend to have lower incomes, but those who also have Roast Beef have the lowest incomes.
- It looks like there's one person who has Canned cranberry sauce and doesn't know what type of main dish he's having.

0.0.28 Aggregating with multiple functions

Find sum, mean and standard deviation of each group in the income column of grouped dataframe

```
[38]: grouped["income"].agg([np.mean, np.sum, np.std]).head(10)
```

```
[38]:
```

	mean \
What type of cranberry saucedo you typically have? What is typically the main dish at your Thanksg...	
Canned	Chicken
80999.600000	
	Ham/Pork
77499.535714	
	I don't know
4999.500000	
	Other (please specify)
53213.785714	
	Roast beef
25499.500000	
	Tofurkey
100713.857143	
	Turkey
85242.682045	
Homemade	Chicken
19999.500000	
	Ham/Pork
96874.625000	
	I don't know
NaN	

	sum \
What type of cranberry saucedo you typically have? What is typically the main dish at your Thanksg...	
Canned	Chicken
404998.0	
	Ham/Pork
1084993.5	
	I don't know
4999.5	
	Other (please specify)
372496.5	
	Roast beef
127497.5	
	Tofurkey
704997.0	
	Turkey
34182315.5	
Homemade	Chicken

59998.5	Ham/Pork
387498.5	I don't know
0.0	
std	
What type of cranberry saucedo you typically have? What is typically the main dish at your Thanksg...	
Canned	Chicken
75779.481062	Ham/Pork
56645.063944	I don't know
NaN	Other (please specify)
29780.946290	Roast beef
24584.039538	Tofurkey
61351.484439	Turkey
55687.436102	Chicken
Homemade	Ham/Pork
16393.596311	I don't know
77308.452805	
NaN	

One of the limitations of aggregation is that each function has to return a single number. While we can perform computations like finding the mean, we can't for example, call `value_counts` to get the exact count of a category. We can do this using the `pandas.GroupBy.apply` method. This method will apply a function to each group, then combine the results.

0.0.29 Find the number of people who live in each area type (Rural, Suburban, etc) who eat different kinds of main dishes for Thanksgiving

```
[39]: grouped = data.groupby("How would you describe where you live?")["What is_
      ↳typically the main dish at your Thanksgiving dinner?"]
      grouped.apply(lambda x:x.value_counts())
```

[39]: How would you describe where you live?

Rural	Turkey	189
	Other (please specify)	9
	Ham/Pork	7

	Tofurkey	3
	I don't know	3
	Chicken	2
	Turducken	2
	Roast beef	1
Suburban	Turkey	449
	Ham/Pork	17
	Other (please specify)	13
	Tofurkey	9
	Roast beef	3
	Chicken	3
	Turducken	1
	I don't know	1
Urban	Turkey	198
	Other (please specify)	13
	Tofurkey	8
	Chicken	7
	Roast beef	6
	Ham/Pork	4

Name: What is typically the main dish at your Thanksgiving dinner?, dtype: int64