Pandas 5
Pivot Tables

```
dogs_height_by_breed_vs_color = dog_pack.pivot_table(
        "height_cm", index="breed", columns="color")
print(dogs_height_by_breed_vs_color)
```

```
color            Black    Brown       Gray        Tan      White
breed
Beagle       34.500000  36.4500  36.313333  35.740000  38.810000
Boxer        57.203333  62.6400  58.280000  62.310000  56.360000
Chihuahua    18.555000      NaN  21.660000  20.096667  17.933333
Chow Chow    51.262500  50.4800        NaN  53.497500  54.413333
Dachshund    21.186667  19.7250        NaN  19.375000  20.660000
Labrador     57.125000      NaN        NaN  55.190000  55.310000
Poodle       48.036000  57.1300  56.645000        NaN  44.740000
St. Bernard  63.920000  65.8825  67.640000  68.334000  67.495000
```

Here the first column - height_cm is the column to aggregate upon
Index column contains the column to group by and display in rows.

columns argument lists the columns to group by and display in columns

.loc[] + slicing is a power combo

```
dogs_height_by_breed_vs_color.loc["Chow Chow":"Poodle"]
```

```
color          Black   Brown    Gray      Tan      White
breed
Chow Chow  51.262500  50.480     NaN  53.4975  54.413333
Dachshund  21.186667  19.725     NaN  19.3750  20.660000
Labrador   57.125000     NaN     NaN  55.1900  55.310000
Poodle     48.036000  57.130  56.645      NaN  44.740000
```

The axis argument

```
dogs_height_by_breed_vs_color.mean(axis="index")
```

```
color
Black    43.973563
Brown    48.717917
Gray     48.107667
Tan      44.934738
White    44.465208
dtype: float64
```

default value for axis is index

Calculating summary stats across columns

```
dogs_height_by_breed_vs_color.mean(axis="columns")
```

```
breed
Beagle         36.362667
Boxer          59.358667
Chihuahua      19.561250
Chow Chow      52.413333
Dachshund      20.236667
Labrador       55.875000
Poodle         51.637750
St. Bernard    66.654300
dtype: float64
```

- Make a pivot table of the avg_temp_c column, with country and city as rows, and year as columns. Assign to temp_by_country_city_vs_year, and *look at the result*.

Add a year column to temperatures
temperatures['year'] = temperatures["date"].dt.year

# Pivot avg_temp_c by country and city vs year
temp_by_country_city_vs_year = temperatures.pivot_table("avg_temp_c", index = ['country', 'city'], columns = 'year')

```
<script.py> output:
    year                              2000    2001    2002    2003    2004  ...    2009    2010    2011    2012    2013
    country       city                                                      ...
    Afghanistan   Kabul             15.823  15.848  15.715  15.133  16.128  ...  15.093  15.676  15.812  14.510  16.206
    Angola        Luanda            24.410  24.427  24.791  24.867  24.216  ...  24.325  24.440  24.151  24.240  24.554
    Australia     Melbourne         14.320  14.180  14.076  13.986  13.742  ...  14.647  14.232  14.191  14.269  14.742
                  Sydney            17.567  17.854  17.734  17.592  17.870  ...  18.176  17.999  17.713  17.474  18.090
    Bangladesh    Dhaka             25.905  25.931  26.095  25.927  26.136  ...  26.536  26.648  25.803  26.284  26.587
    ...                               ...     ...     ...     ...     ...  ...     ...     ...     ...     ...     ...
    United States Chicago           11.090  11.703  11.532  10.482  10.943  ...  10.298  11.816  11.214  12.821  11.587
                  Los Angeles       16.643  16.466  16.430  16.945  16.553  ...  16.677  15.887  15.875  17.090  18.121
                  New York           9.969  10.931  11.252   9.836  10.389  ...  10.142  11.358  11.272  11.971  12.164
    Vietnam       Ho Chi Minh City  27.589  27.832  28.065  27.828  27.687  ...  27.853  28.282  27.675  28.249  28.455
    Zimbabwe      Harare            20.284  20.861  21.079  20.889  20.308  ...  20.524  21.166  20.782  20.523  19.756

    [100 rows x 14 columns]
```