10

SOLUTIONS. Solutions vary; this was a great example from a student.

Dr. Ian Frommer

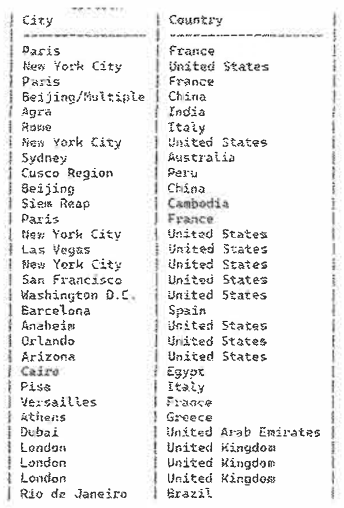
Data Science Techniques 11 February 2026

Data Science Techniques – Homework 4

Question #1

✓

- Example 1: csvcut -c City,Country world_famous_places_2024.csv | csvlook

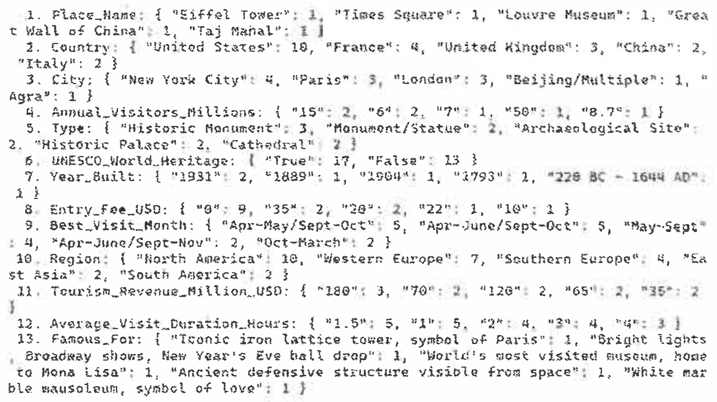


```
| City               | Country              |
| ------------------ | -------------------- |
| Paris              | France               |
| New York City      | United States        |
| Paris              | France               |
| Beijing/Multiple   | China                |
| Agra               | India                |
| Rome               | Italy                |
| New York City      | United States        |
| Sydney             | Australia            |
| Cusco Region       | Peru                 |
| Beijing            | China                |
| Siem Reap          | Cambodia             |
| Paris              | France               |
| New York City      | United States        |
| Las Vegas          | United States        |
| New York City      | United States        |
| San Francisco      | United States        |
| Washington D.C.    | United States        |
| Barcelona          | Spain                |
| Anaheim            | United States        |
| Orlando            | United States        |
| Arizona            | United States        |
| Cairo              | Egypt                |
| Pisa               | Italy                |
| Versailles         | France               |
| Athens             | Greece               |
| Dubai              | United Arab Emirates |
| London             | United Kingdom       |
| London             | United Kingdom       |
| London             | United Kingdom       |
| Rio de Janeiro     | Brazil               |
```

In Example 1, I select the columns "City" and "Country" from the world_famous_places_2024.csv using csvcut, which filters .csv files. From there, I piped it to the csvlook command, which displays the .csv file in tabular form. This complete command allows us to view specific portions of data in an organized fashion.

✓

- Example 2: csvstat –freq world_famous_places_2024.csv | sort



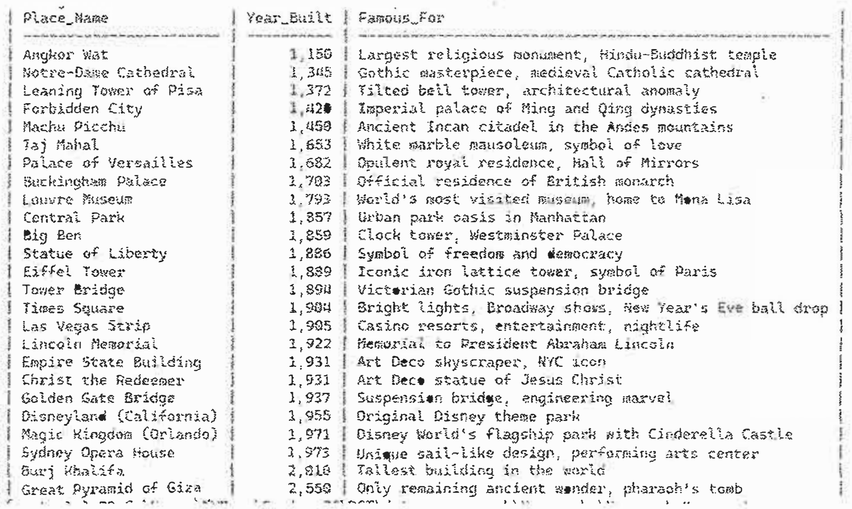
```
  1. Place_Name: { "Eiffel Tower": 1, "Times Square": 1, "Louvre Museum": 1, "Grea
t Wall of China": 1, "Taj Mahal": 1 }
  2. Country: { "United States": 10, "France": 4, "United Kingdom": 3, "China": 2,
 "Italy": 2 }
  3. City: { "New York City": 4, "Paris": 3, "London": 3, "Beijing/Multiple": 1, "
Agra": 1 }
  4. Annual_Visitors_Millions: { "15": 2, "6": 2, "7": 1, "50": 1, "8.7": 1 }
  5. Type: { "Historic Monument": 3, "Monument/Statue": 2, "Archaeological Site":
2, "Historic Palace": 2, "Cathedral": 2 }
  6. UNESCO_World_Heritage: { "True": 17, "False": 13 }
  7. Year_Built: { "1931": 2, "1889": 1, "1904": 1, "1793": 1, "220 BC - 1644 AD":
 1 }
  8. Entry_Fee_USD: { "0": 9, "35": 2, "20": 2, "22": 1, "10": 1 }
  9. Best_Visit_Month: { "Apr-May/Sept-Oct": 5, "Apr-June/Sept-Oct": 5, "May-Sept"
: 4, "Apr-June/Sept-Nov": 2, "Oct-March": 2 }
 10. Region: { "North America": 10, "Western Europe": 7, "Southern Europe": 4, "Ea
st Asia": 2, "South America": 2 }
 11. Tourism_Revenue_Million_USD: { "180": 3, "70": 2, "120": 2, "65": 2, "35": 2
}
 12. Average_Visit_Duration_Hours: { "1.5": 5, "1": 5, "2": 4, "3": 4, "4": 3 }
 13. Famous_For: { "Iconic iron lattice tower, symbol of Paris": 1, "Bright lights
, Broadway shows, New Year's Eve ball drop": 1, "World's most visited museum, home
 to Mona Lisa": 1, "Ancient defensive structure visible from space": 1, "White mar
ble mausoleum, symbol of love": 1 }
```

Nice

The command used here, csvstat, provides a descripted summary of columns, but by adding "—freq" afterwards, the most common values for each column are outputted. I piped them to the sort command to order them neatly.

- Example 3: csvcut -c Place_Name,Year_Built,Famous_For world_famous_places_2024.csv | csvgrep -c Year_Built -r '^\d{4}$' | csvsort -c Year_Built | csvlook

✓

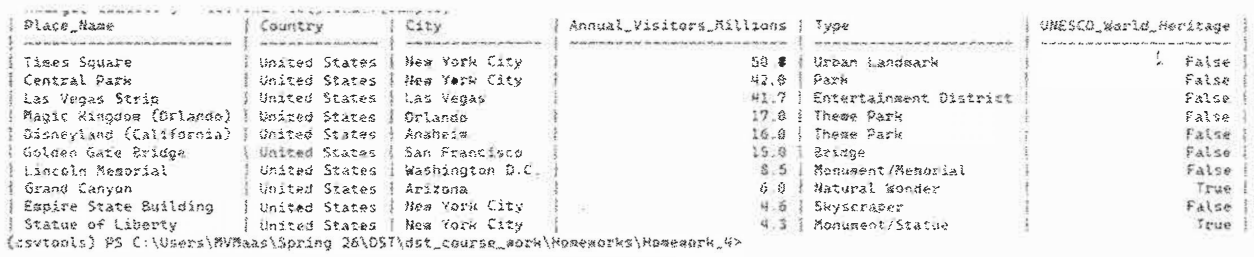


```
| Place_Name                | Year_Built | Famous_For                                         |
| ------------------------- | ---------- | -------------------------------------------------- |
| Angkor Wat                |      1,150 | Largest religious monument, Hindu-Buddhist temple  |
| Notre-Dame Cathedral      |      1,345 | Gothic masterpiece, medieval Catholic cathedral    |
| Leaning Tower of Pisa     |      1,372 | Tilted bell tower, architectural anomaly           |
| Forbidden City            |      1,420 | Imperial palace of Ming and Qing dynasties         |
| Machu Picchu              |      1,450 | Ancient Incan citadel in the Andes mountains       |
| Taj Mahal                 |      1,653 | White marble mausoleum, symbol of love             |
| Palace of Versailles      |      1,682 | Opulent royal residence, Hall of Mirrors           |
| Buckingham Palace         |      1,703 | Official residence of British monarch              |
| Louvre Museum             |      1,793 | World's most visited museum, home to Mona Lisa     |
| Central Park              |      1,857 | Urban park oasis in Manhattan                      |
| Big Ben                   |      1,859 | Clock tower, Westminster Palace                    |
| Statue of Liberty         |      1,886 | Symbol of freedom and democracy                    |
| Eiffel Tower              |      1,889 | Iconic iron lattice tower, symbol of Paris         |
| Tower Bridge              |      1,894 | Victorian Gothic suspension bridge                 |
| Times Square              |      1,904 | Bright lights, Broadway shows, New Year's Eve ball drop |
| Las Vegas Strip           |      1,905 | Casino resorts, entertainment, nightlife           |
| Lincoln Memorial          |      1,922 | Memorial to President Abraham Lincoln              |
| Empire State Building     |      1,931 | Art Deco skyscraper, NYC icon                      |
| Christ the Redeemer       |      1,931 | Art Deco statue of Jesus Christ                    |
| Golden Gate Bridge        |      1,937 | Suspension bridge, engineering marvel              |
| Disneyland (California)   |      1,955 | Original Disney theme park                         |
| Magic Kingdom (Orlando)   |      1,971 | Disney World's flagship park with Cinderella Castle |
| Sydney Opera House        |      1,973 | Unique sail-like design, performing arts center    |
| Burj Khalifa              |      2,010 | Tallest building in the world                      |
| Great Pyramid of Giza     |      2,550 | Only remaining ancient wonder, pharaoh's tomb      |
```

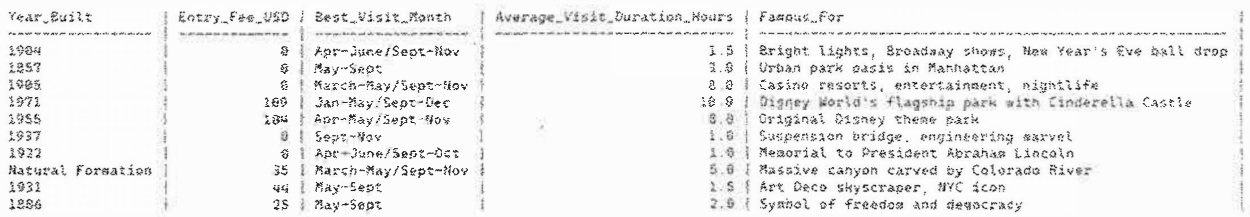
As displayed in the image above, I used csvcut to splice the .csv file and extract the three columns listed. From there, I used the csvgrep command to filter out any years that did not match the four-digit format as specified in the regular expression. Then, I sorted all entries by "Year_Built" from oldest to newest. csvlook outputs this data in a more readable way.

nice

- Example 4: csvgrep -c Region -m 'North America' world_famous_places_2024.csv | csvcut -C Region,Tourism_Revenue_Million_USD | csvsort -r -c Annual_Visitors_Millions | csvlook



```
| Place_Name              | Country       | City             | Annual_Visitors_Millions | Type                   | UNESCO_World_Heritage |
| ----------------------- | ------------- | ---------------- | ------------------------ | ---------------------- | --------------------- |
| Times Square            | United States | New York City    |                     50.0 | Urban Landmark         |                 False |
| Central Park            | United States | New York City    |                     42.0 | Park                   |                 False |
| Las Vegas Strip         | United States | Las Vegas        |                     41.7 | Entertainment District |                 False |
| Magic Kingdom (Orlando) | United States | Orlando          |                     17.0 | Theme Park             |                 False |
| Disneyland (California) | United States | Anaheim          |                     16.0 | Theme Park             |                 False |
| Golden Gate Bridge      | United States | San Francisco    |                     15.0 | Bridge                 |                 False |
| Lincoln Memorial        | United States | Washington D.C.  |                      8.5 | Monument/Memorial      |                 False |
| Grand Canyon            | United States | Arizona          |                      6.0 | Natural Wonder         |                  True |
| Empire State Building   | United States | New York City    |                      4.6 | Skyscraper             |                 False |
| Statue of Liberty       | United States | New York City    |                      4.3 | Monument/Statue        |                  True |
(csvtools) PS C:\Users\MVMaas\Spring 26\DST\dst_course_work\Homeworks\Homework_4>
```



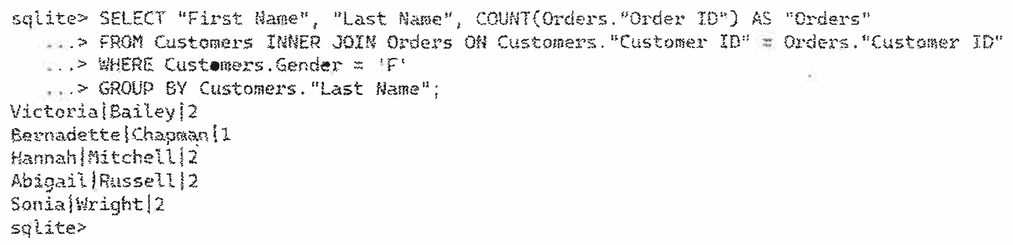
```
Year_Built        | Entry_Fee_USD | Best_Visit_Month      | Average_Visit_Duration_Hours | Famous_For                                              |
----------------- | ------------- | --------------------- | ---------------------------- | ------------------------------------------------------- |
1904              |             0 | Apr-June/Sept-Nov     |                          1.5 | Bright lights, Broadway shows, New Year's Eve ball drop |
1857              |             0 | May-Sept              |                          3.0 | Urban park oasis in Manhattan                           |
1905              |             0 | March-May/Sept-Nov    |                          8.0 | Casino resorts, entertainment, nightlife                |
1971              |           109 | Jan-May/Sept-Dec      |                         10.0 | Disney World's flagship park with Cinderella Castle     |
1955              |           104 | Apr-May/Sept-Nov      |                          8.0 | Original Disney theme park                              |
1937              |             0 | Sept-Nov              |                          1.0 | Suspension bridge, engineering marvel                   |
1922              |             0 | Apr-June/Sept-Oct     |                          1.0 | Memorial to President Abraham Lincoln                   |
Natural Formation |            35 | March-May/Sept-Nov    |                          5.0 | Massive canyon carved by Colorado River                 |
1931              |            44 | May-Sept              |                          1.5 | Art Deco skyscraper, NYC icon                           |
1886              |            25 | May-Sept              |                          2.5 | Symbol of freedom and democracy                         |
```

great

In Example 4, I filtered for all sites in the North American region using csvgrep before cutting the two columns listed from the final output. Then, I sorted these remaining entries by "Annual_Visitors_Millions" to order them from most to least popular. Finally, I ran the csvlook command to improve the output appearance.

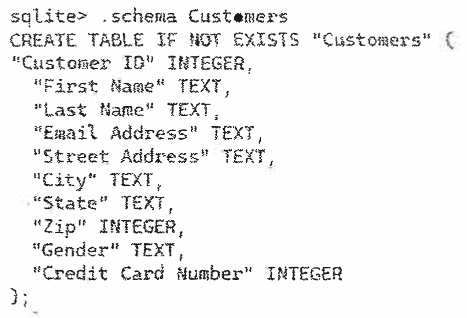
Question #2

A. I used the feline.db for this portion of the homework. I used the following query to gather the first and last names, as well as the total number of orders, of each female customer.



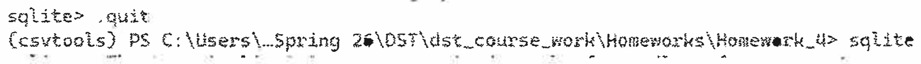
```
sqlite> SELECT "First Name", "Last Name", COUNT(Orders."Order ID") AS "Orders"
   ...> FROM Customers INNER JOIN Orders ON Customers."Customer ID" = Orders."Customer ID"
   ...> WHERE Customers.Gender = 'F'
   ...> GROUP BY Customers."Last Name";
Victoria|Bailey|2
Bernadette|Chapman|1
Hannah|Mitchell|2
Abigail|Russell|2
Sonia|Wright|2
sqlite>
```

B. In addition to .tables, I used the .schema command to identify the columns in the Customer and Orders tables.



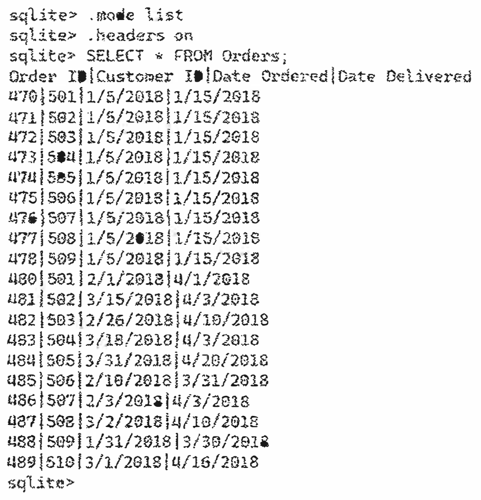
```
sqlite> .schema Customers
CREATE TABLE IF NOT EXISTS "Customers" (
"Customer ID" INTEGER,
  "First Name" TEXT,
  "Last Name" TEXT,
  "Email Address" TEXT,
  "Street Address" TEXT,
  "City" TEXT,
  "State" TEXT,
  "Zip" INTEGER,
  "Gender" TEXT,
  "Credit Card Number" INTEGER
);
```

I used the .quit command to exit out of sqlite3 whenever I needed to either switch databases or make other changes in the command line.



```
sqlite> .quit
(csvtools) PS C:\Users\...Spring 26\DST\dst_course_work\Homeworks\Homework_4> sqlite
```

To change the output appearance, I used the .mode list and .headers on commands prior to running the query shown to get the output below.



```
sqlite> .mode list
sqlite> .headers on
sqlite> SELECT * FROM Orders;
Order ID|Customer ID|Date Ordered|Date Delivered
470|501|1/5/2018|1/15/2018
471|502|1/5/2018|1/15/2018
472|503|1/5/2018|1/15/2018
473|504|1/5/2018|1/15/2018
474|505|1/5/2018|1/15/2018
475|506|1/5/2018|1/15/2018
476|507|1/5/2018|1/15/2018
477|508|1/5/2018|1/15/2018
478|509|1/5/2018|1/15/2018
480|501|2/1/2018|4/1/2018
481|502|3/15/2018|4/3/2018
482|503|2/26/2018|4/10/2018
483|504|3/18/2018|4/3/2018
484|505|3/31/2018|4/20/2018
485|506|2/10/2018|3/31/2018
486|507|2/3/2018|4/3/2018
487|508|3/2/2018|4/10/2018
488|509|1/31/2018|3/30/2018
489|510|3/1/2018|4/16/2018
sqlite>
```