Walkthrough

Baby Names MiniProject: Overview 4 min

Baby Names MiniProject: Data Overview 6 min

Baby Names MiniProject: Total Births 7 min

MiniProject Exercise Guide 10 min

Quiz: Baby Names 14 questions

Extend Your Program 10 min

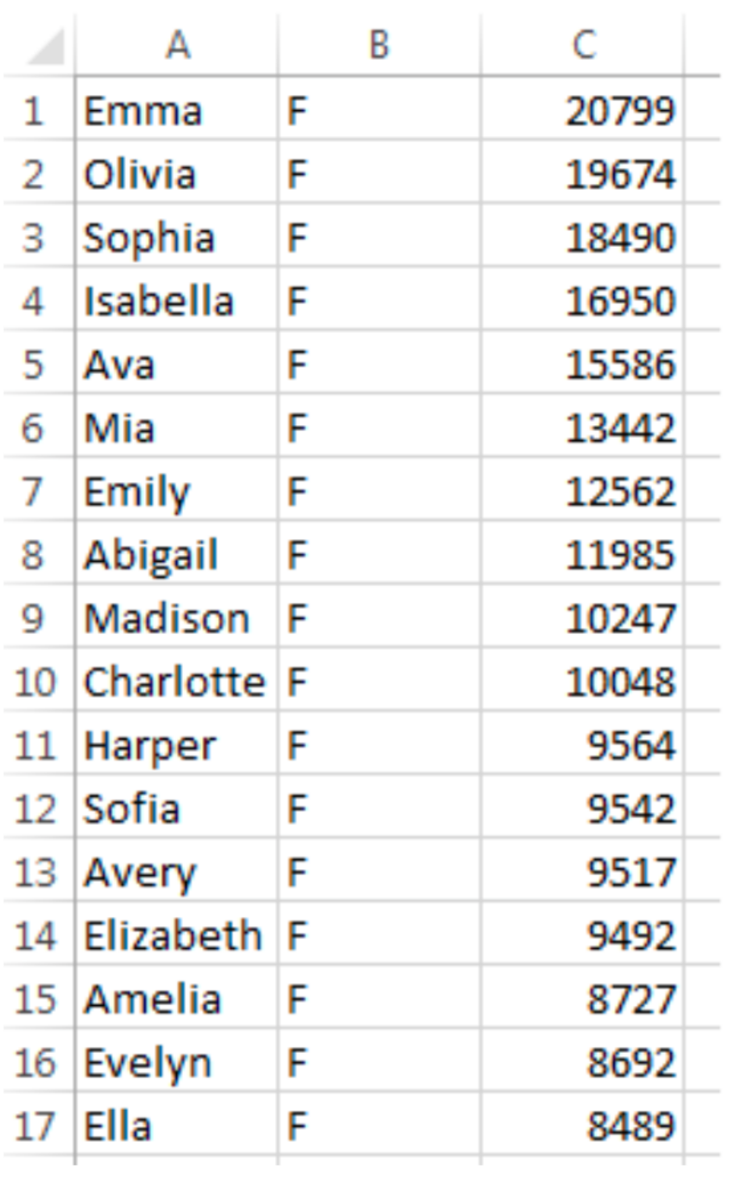
End of Module Survey 10 min

Batch Grayscale Images

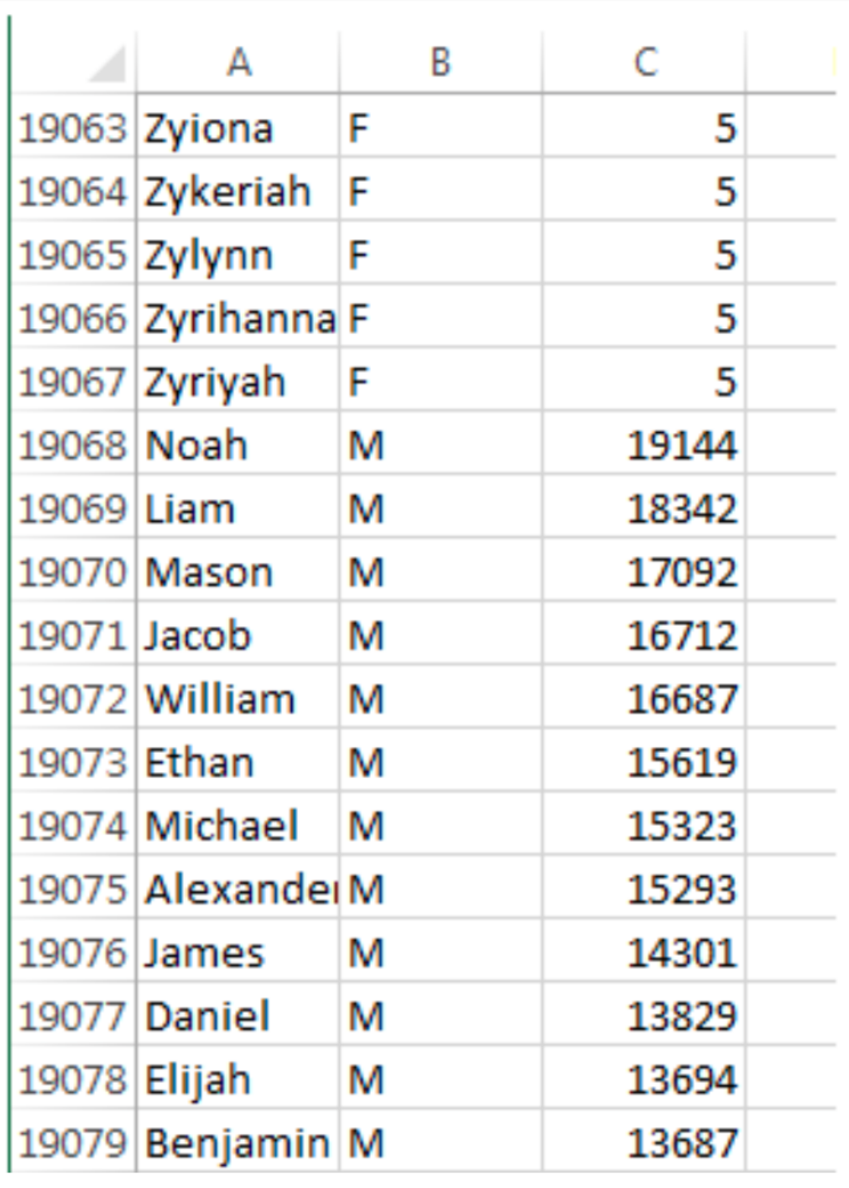
Assignment

For this assignment, we are providing data on baby names from the United States and you will answer questions about this data. The data files give information on the first names of people born in a particular year. We have data from 1880 through 2014 on both boys and girls names. You can download a .zip folder of all baby name data by clicking here.

For example, below are two pictures of parts of the file "yob2014.csv", for names of babies born in 2014. On the left is the top of the file which shows that Emma is the most popular name, the F is for female and the 20799 indicates the number of Emma's born in 2014. The file lists all the girls first, and they are listed in order based on the number of births, from largest numbers to smallest numbers. On the right is another snapshot of the file, showing the end of the girls and the start of the boys. You can see that Noah was the most popular boys name in 2014, the M indicates a male, and 19144 is the number of boys named Noah that year. Also notice there are no headers in this file. The video showed you how to access the fields in a .csv file when there are no headers.



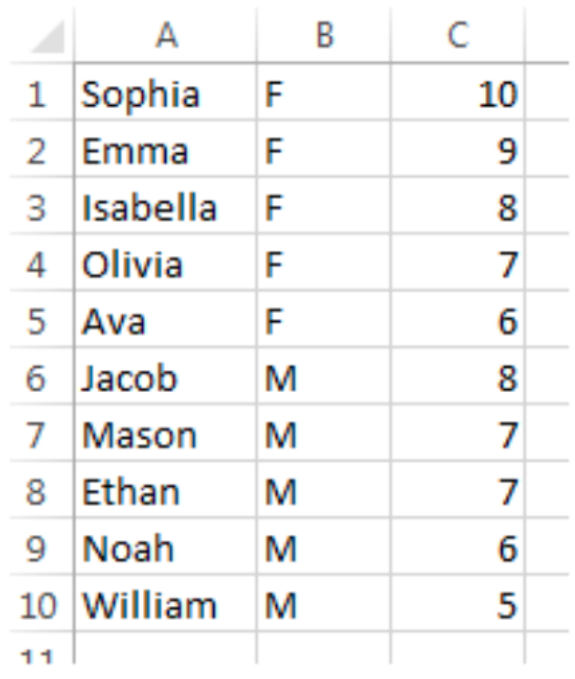
	A	B	C
1	Emma	F	20799
2	Olivia	F	19674
3	Sophia	F	18490
4	Isabella	F	16950
5	Ava	F	15586
6	Mia	F	13442
7	Emily	F	12562
8	Abigail	F	11985
9	Madison	F	10247
10	Charlotte	F	10048
11	Harper	F	9564
12	Sofia	F	9542
13	Avery	F	9517
14	Elizabeth	F	9492
15	Amelia	F	8727
16	Evelyn	F	8692
17	Ella	F	8489



	A	B	C
19063	Zyiona	F	5
19064	Zykeriah	F	5
19065	Zylynn	F	5
19066	Zyrihanna	F	5
19067	Zyriyah	F	5
19068	Noah	M	19144
19069	Liam	M	18342
19070	Mason	M	17092
19071	Jacob	M	16712
19072	William	M	16687
19073	Ethan	M	15619
19074	Michael	M	15323
19075	Alexande	M	15293
19076	James	M	14301
19077	Daniel	M	13829
19078	Elijah	M	13694
19079	Benjamin	M	13687

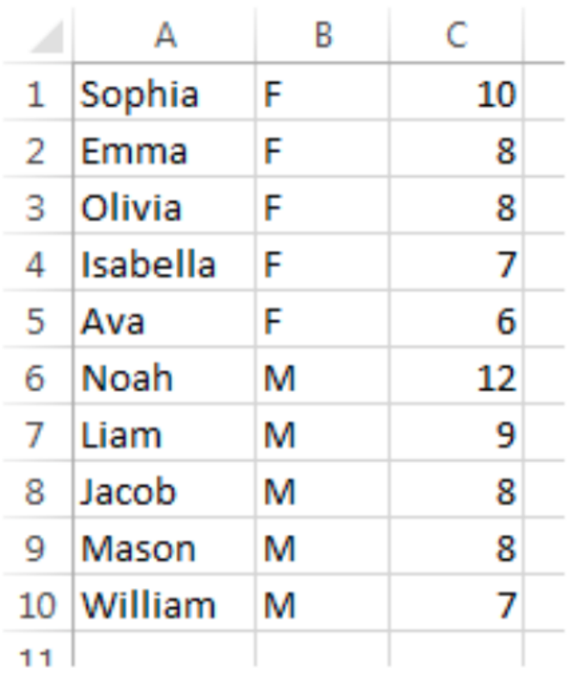
Since these data files are quite large, we will also provide three small files with fake data for testing. We show the three files called "yob2012short.csv", "yob2013short.csv" and "yob2014short.csv" below in that order. The three files have only ten lines each, 5 girls and 5 boys names, and smaller numbers to work with. You can download a .zip folder of the shortened baby name data by clicking here.

yob2012short.csv



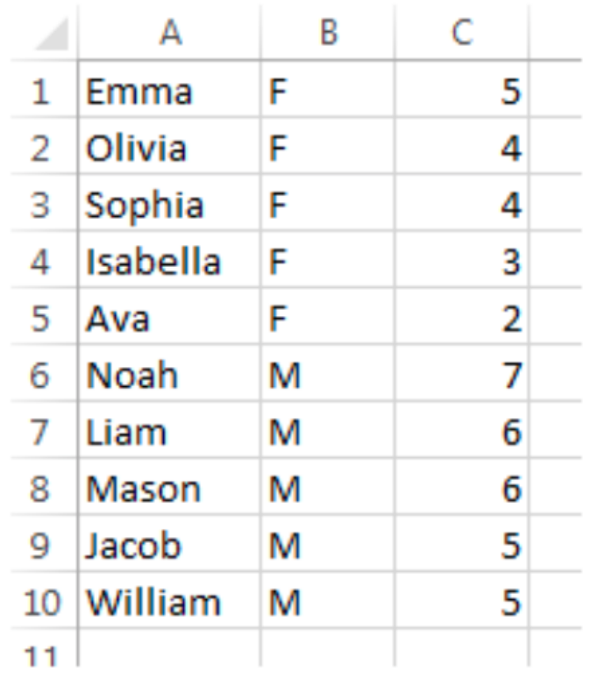
	A	B	C
1	Sophia	F	10
2	Emma	F	9
3	Isabella	F	8
4	Olivia	F	7
5	Ava	F	6
6	Jacob	M	8
7	Mason	M	7
8	Ethan	M	7
9	Noah	M	6
10	William	M	5
11			

yob2013short.csv



	A	B	C
1	Sophia	F	10
2	Emma	F	8
3	Olivia	F	8
4	Isabella	F	7
5	Ava	F	6
6	Noah	M	12
7	Liam	M	9
8	Jacob	M	8
9	Mason	M	8
10	William	M	7
11			

yob2014short.csv



	A	B	C
1	Emma	F	5
2	Olivia	F	4
3	Sophia	F	4
4	Isabella	F	3
5	Ava	F	2
6	Noah	M	7
7	Liam	M	6
8	Mason	M	6
9	Jacob	M	5
10	William	M	5
11			

You will write a program with several methods and tester methods to test each method you write. You should start with understanding the methods shown in the video. Specifically you should write the following methods.

- Modify the method **totalBirths** (shown in the video for this project) to also print the number of girls names , the number of boys names and the total names in the file.
- Write the method named **getRank** that has three parameters: an integer named **year**, a string named **name**, and a string named **gender** (F for female and M for male). This method returns the rank of the name in the file for the given gender, where rank 1 is the name with the largest number of births. If the name is not in the file, then -1 is returned. For example, in the file "yob2012short.csv", given the name Mason, the year 2012 and the gender 'M', the number returned is 2, as Mason is the boys name with the second highest number of births. Given the name Mason, the year 2012 and the gender 'F', the number returned is -1 as Mason does not appear with an F in that file.
- Write the method named **getName** that has three parameters: an integer named **year**, an integer named **rank**, and a string named **gender** (F for female and M for male). This method returns the name of the person in the file at this rank, for the given gender, where rank 1 is the name with the largest number of births. If the rank does not exist in the file, then "NO NAME" is returned.
- What would your name be if you were born in a different year? Write the void method named **whatIsNameInYear** that has four parameters: a string **name**, an integer named **year** representing the year that name was born, an integer named **newYear** and a string named **gender** (F for female and M for male). This method determines what **name** would have been named if they were born in a different year, based on the same popularity. That is, you should determine the rank of name in the year they were born, and then print the name born in **newYear** that is at the same rank and same gender. For example, using the files "yob2012short.csv" and "yob2014short.csv", notice that in 2012 Isabella is the third most popular girl's name. If Isabella was born in 2014 instead, she would have been named Sophia, the third most popular girl's name that year. The output might look like this:

```
Isabella born in 2012 would be Sophia if she was born in 2014.
```

- Write the method **yearOfHighestRank** that has two parameters: a string **name**, and a string named **gender** (F for female and M for male). This method selects a range of files to process and returns an integer, the year with the highest rank for the name and gender. If the name and gender are not in any of the selected files, it should return -1. For example, calling **yearOfHighestRank** with name Mason and gender 'M' and selecting the three test files above results in returning the year 2012. That is because Mason was ranked the 2nd most popular name in 2012, ranked 4th in 2013 and ranked 3rd in 2014. His highest ranking was in 2012.
- Write the method **getAverageRank** that has two parameters: a string **name**, and a string named **gender** (F for female and M for male). This method selects a range of files to process and returns a double representing the average rank of the name and gender over the selected files. It should return -1.0 if the name is not ranked in any of the selected files. For example calling getAverageRank with name Mason and gender 'M' and selecting the three test files above results in returning 3.0, as he is rank 2 in the year 2012, rank 4 in 2013 and rank 3 in 2014. As another example, calling **getAverageRank** with name Jacob and gender 'M' and selecting the three test files above results in returning 2.66.
- Write the method **getTotalBirthsRankedHigher** that has three parameters: an integer named **year**, a string named **name**, and a string named **gender** (F for female and M for male). This method returns an integer, the total number of births of those names with the same gender and same year who are ranked higher than **name**. For example, if **getTotalBirthsRankedHigher** accesses the "yob2012short.csv" file with name set to "Ethan", gender set to "M", and year set to 2012, then this method should return 15, since Jacob has 8 births and Mason has 7 births, and those are the only two ranked higher than Ethan.

Link to FAQ page for this course: http://www.dukelearntoprogram.com/course2/faq.php

Module5-MiniProject-ProgrammingExerci…

Mark as completed