



The Relationship between NHL Players and Different Birth Months

The Beautiful Soups
Noah Chazonoff & Andrea Holber



Project Synopsis

Used NHL API and a hockey stats website to find the top 100 NHL leaders' birth months and birth countries

WHY? To see if Malcolm Gladwell's theory applied to all countries (instead of just Canada): the majority of hockey players were born in the first three months of the year.

Original Goals

- ORIGINAL GOAL

- Create playlists using the billboard 100
 - We realized this was too intricate and data calculation would be difficult
-

- NEW GOAL

- Shifted goal to find if a relationship existed between the top NHL players and when and where they were born.

- NEXT STEPS

- Finding the most common birth months & countries for top NHL players over the past 5 seasons
- Joining the table with player names & points with the table that has names, month, and country
- Creating 3 visualizations from our data

Goals Achieved

- BEAUTIFULSOUP
 - We scraped data from a website that listed the top players who led the league in points over the last 5 years
- API SCRAPING
 - We scraped data from an NHL API to gather other crucial info about all of the players in the league
- JOIN
 - We joined the two tables of 121 players and created three visualizations
- CONCLUSION
 - We were able to find that the relationship that is prominent between **CANADIAN** NHL players and their birth months does not apply to the entirety of the league.

Problems

1. Google Search API stopped working half way through our data collection
 - a. We upgraded our API subscription to permit us 5,000 searches a month and it still did not work :/
2. We switched APIs, but then we were getting every player in the league, instead of the top 100 we wanted
3. Everytime we ran the databases, IT TOOK 4 HOURS TO LOAD
4. When running the code multiple times, our ID tables would input the same information more than once.



Instructions for Running Code



Code Instruction

BETTERCOMBINED.PY

1. Delete database if already on computer in order to create new database and begin running code.
2. Run code 5 times from `bettercombined.py` in order to attain database information. Running the code 5 times will ensure that 121 data points will appear in our largest tables. Note: this takes a relatively long time as the website we scraped takes a little longer to attain data from (this was witnessed and determined with Uche's help).
3. Once this is done, comment out `lines 359-372` in order to stop running the databases to speed up the rest of the code we are running.
4. Uncomment `lines 365-372` to attain 5 different calculations, and 3 different output files.

Code Instruction (cont'd)

VISUALIZATIONS.PY

1. Run `visualizations.py` to attain our 3 different visualizations (2 bar graphs, 1 pie chart)

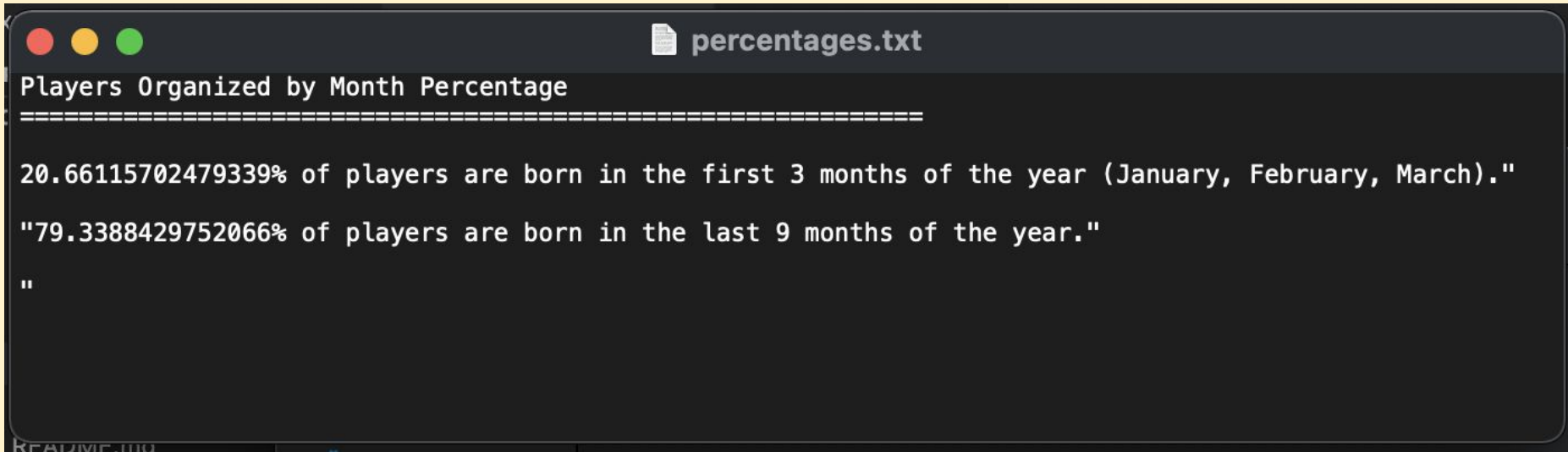
Calculation #1

```
top_ten_player_info.txt
Top 10 Point Leaders in the NHL over the last 5 seasons.
=====
Connor McDavid has 498 points over the last 5 seasons in the NHL.
Leon Draisaitl has 426 points over the last 5 seasons in the NHL.
Patrick Kane has 417 points over the last 5 seasons in the NHL.
Brad Marchand has 414 points over the last 5 seasons in the NHL.
Nathan MacKinnon has 401 points over the last 5 seasons in the NHL.
Artemi Panarin has 398 points over the last 5 seasons in the NHL.
Sidney Crosby has 394 points over the last 5 seasons in the NHL.
David Pastrnak has 381 points over the last 5 seasons in the NHL.
Alexander Ovechkin has 367 points over the last 5 seasons in the NHL.
Blake Wheeler has 354 points over the last 5 seasons in the NHL.
The Average Point Total for the Top 10 Players.
=====
The average number of points over the past 5 seasons by the top 10 players is 405.
```

Calculation #2

```
country_and_month_info.txt
Players Organized by What Country They Were Born In.
=====
28.92% of players were born in USA."
"43.80% of players were born in CAN."
"9.917% of players were born in SWE."
"4.958% of players were born in RUS."
"2.479% of players were born in CZE."
"0.826% of players were born in SVK."
"2.479% of players were born in CHE."
"4.132% of players were born in FIN."
"0.826% of players were born in DEU."
"0.826% of players were born in SVN."
"0.826% of players were born in NOR."
"Players Organized by What Month They Were Born In.
=====
12.39% of players were born in July."
"13.22% of players were born in May."
"6.611% of players were born in Feb."
"11.57% of players were born in Oct."
"7.438% of players were born in Apr."
"4.958% of players were born in Jan."
"9.090% of players were born in Dec."
"9.090% of players were born in Mar."
"5.785% of players were born in Aug."
"5.785% of players were born in Nov."
"5.785% of players were born in Jun."
"8.264% of players were born in Sep."
```

Calculation #3

A terminal window with a dark background and a title bar. The title bar contains three colored window control buttons (red, yellow, green) on the left and a document icon followed by the filename 'percentages.txt' on the right. The terminal content is as follows:

```
Players Organized by Month Percentage
=====
20.66115702479339% of players are born in the first 3 months of the year (January, February, March)."
"79.3388429752066% of players are born in the last 9 months of the year."
"
```

Tables

Players Table

(includes names and points from website)

	id	name	points
	Filt...	Filter	Filter
1	1	Connor McDavid	498
2	2	Leon Draisaitl	426
3	3	Patrick Kane	417
4	4	Brad Marchand	414
5	5	Nathan MacKinnon	401
6	6	Artemi Panarin	398
7	7	Sidney Crosby	394
8	8	David Pastrnak	381
9	9	Mitchell Marner	353
10	10	Auston Matthews	350
11	11	John Tavares	343
12	12	Evgeni Malkin	342
13	13	Johnny Gaudreau	340
14	14	Aleksander Barkov	338
15	15	Claude Giroux	337
16	16	Jonathan Huberdeau	319
17	17	Anze Kopitar	317
18	18	Sebastian Aho	311
19	19	Brayden Point	305
20	20	Steven Stamkos	304
21	21	Mikko Rantanen	304
22	22	Jack Eichel	302
23	23	Ryan O'Reilly	299
24	24	Brent Burns	297
25	25	Evgeny Kuznetsov	297
26	26	Patrice Bergeron	294

	id	name	points
	Filt...	Filter	Filter
96	96	Ryan Strome	205
97	97	Brendan Gallagher	204
98	98	Nino Niederreiter	202
99	99	Alex Killorn	201
100	100	Seth Jones	201
101	101	Brock Boeser	200
102	102	Anthony Mantha	199
103	103	Bryan Rust	199
104	104	Tyler Toffoli	198
105	105	Adam Henrique	196
106	106	Brandon Saad	194
107	107	Tyler Johnson	194
108	108	Zach Parise	194
109	109	Pavel Buchnevich	192
110	110	Jonathan Drouin	191
111	111	Kevin Fiala	191
112	112	Jeff Petry	190
113	113	Jakob Silfverberg	190
114	114	Phillip Danault	189
115	115	Timo Meier	189
116	116	Yanni Gourde	188
117	117	Joe Thornton	187
118	118	Derek Stepan	187
119	119	Zach Hyman	184
120	120	Jordan Staal	182
121	121	Charlie Coyle	180

Country ID Table

	id	country
	Fi...	Filter
1	1	USA
2	2	CAN
3	3	SWE
4	4	RUS
5	5	CZE
6	6	SVK
7	7	CHE
8	8	FIN
9	9	DEU
10	10	SVN
11	11	NOR

API Table 1

Month ID Table

	id	month
	Fi...	Filter
1	1	Jan
2	2	Feb
3	3	Mar
4	4	Apr
5	5	May
6	6	Jun
7	7	July
8	8	Aug
9	9	Sep
10	10	Oct
11	11	Nov
12	12	Dec

API Table 2

Birthdays Table includes names, birth months, and birthplaces

	id	name	birth_month	birth_place
	Filt...	Filter	Filter	Filter
1	1	Anders Lee	7	1
2	2	Jordan Eberle	5	2
3	3	Kyle Palmieri	2	1
4	4	Brock Nelson	10	1
5	5	Mathew Barzal	5	2
6	6	Sebastian Aho	2	3
7	7	Chris Kreider	4	1
8	8	Ryan Strome	7	2
9	9	Mika Zibanejad	4	3
10	10	Pavel Buchnevich	4	4
11	11	Artemi Panarin	10	4
12	12	Claude Giroux	1	2
13	13	James van Riemsdyk	5	1
14	14	Kevin Hayes	5	1
15	15	Sean Couturier	12	1
16	16	Travis Konecny	3	2
17	17	Evgeni Malkin	7	4
18	18	Sidney Crosby	8	2
19	19	Jason Zucker	1	1
20	20	Bryan Rust	5	1
21	21	Jake Guentzel	10	1
22	22	Patrice Bergeron	7	2
23	23	Brad Marchand	5	2
24	24	Charlie Coyle	3	1
25	25	Taylor Hall	11	2
26	26	David Pastrnak	5	5

	id	name	birth_month	birth_place
	Filt...	Filter	Filter	Filter
79	79	Torey Krug	4	1
80	80	Mark Giordano	10	2
81	81	Mikael Backlund	3	3
82	82	Johnny Gaudreau	8	1
83	83	Elias Lindholm	12	3
84	84	Sean Monahan	10	2
85	85	Matthew Tkachuk	12	1
86	86	Nazem Kadri	10	2
87	87	Brandon Saad	10	1
88	88	Gabriel Landeskog	11	3
89	89	Nathan MacKinnon	9	2
90	90	Mikko Rantanen	10	8
91	91	Tyson Barrie	7	2
92	92	Ryan Nugent-Hopkins	4	2
93	93	Leon Draisaitl	10	9
94	94	Connor McDavid	1	2
95	95	J.T. Miller	3	1
96	96	Bo Horvat	4	2
97	97	Brock Boeser	2	1
98	98	Ryan Getzlaf	5	2
99	99	Adam Henrique	2	2
100	100	Jakob Silfverberg	10	3
101	101	Rickard Rakell	5	3
102	102	Alexander Radulov	7	4
103	103	Joe Pavelski	7	1
104	104	Jamie Benn	7	2

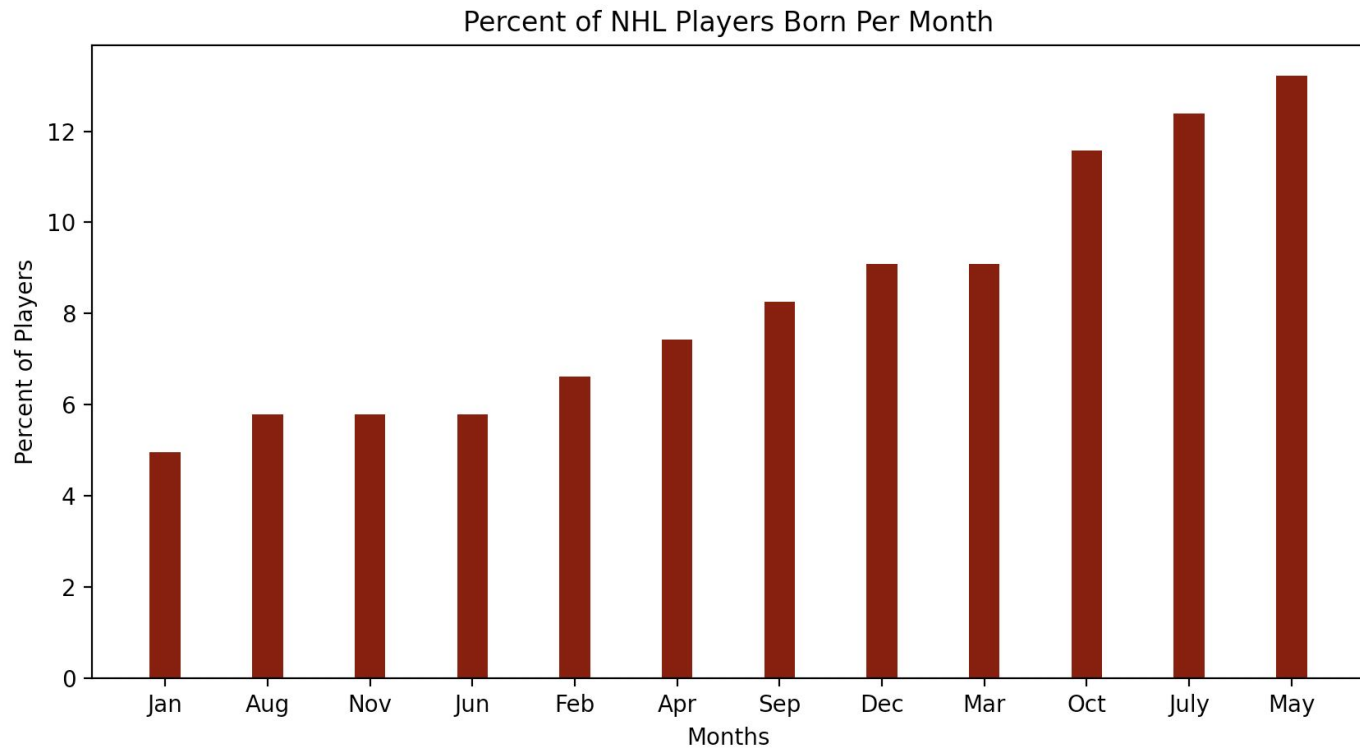
JOIN Table (player name = shared id)

	name	points	birth_month	birth_place
1	Connor McDavid	498	1	2
2	Leon Draisaitl	426	10	9
3	Patrick Kane	417	11	1
4	Brad Marchand	414	5	2
5	Nathan MacKinnon	401	9	2
6	Artemi Panarin	398	10	4
7	Sidney Crosby	394	8	2
8	David Pastrnak	381	5	5
9	Mitchell Marner	353	5	2
10	Auston Matthews	350	9	1
11	John Tavares	343	9	2
12	Evgeni Malkin	342	7	4
13	Johnny Gaudreau	340	8	1
14	Aleksander Barkov	338	9	8
15	Claude Giroux	337	1	2
16	Jonathan Huberdeau	319	6	2
17	Anze Kopitar	317	8	10
18	Sebastian Aho	311	2	3
19	Sebastian Aho	311	7	8
20	Brayden Point	305	3	2
21	Steven Stamkos	304	2	2
22	Mikko Rantanen	304	10	8
23	Jack Eichel	302	10	1

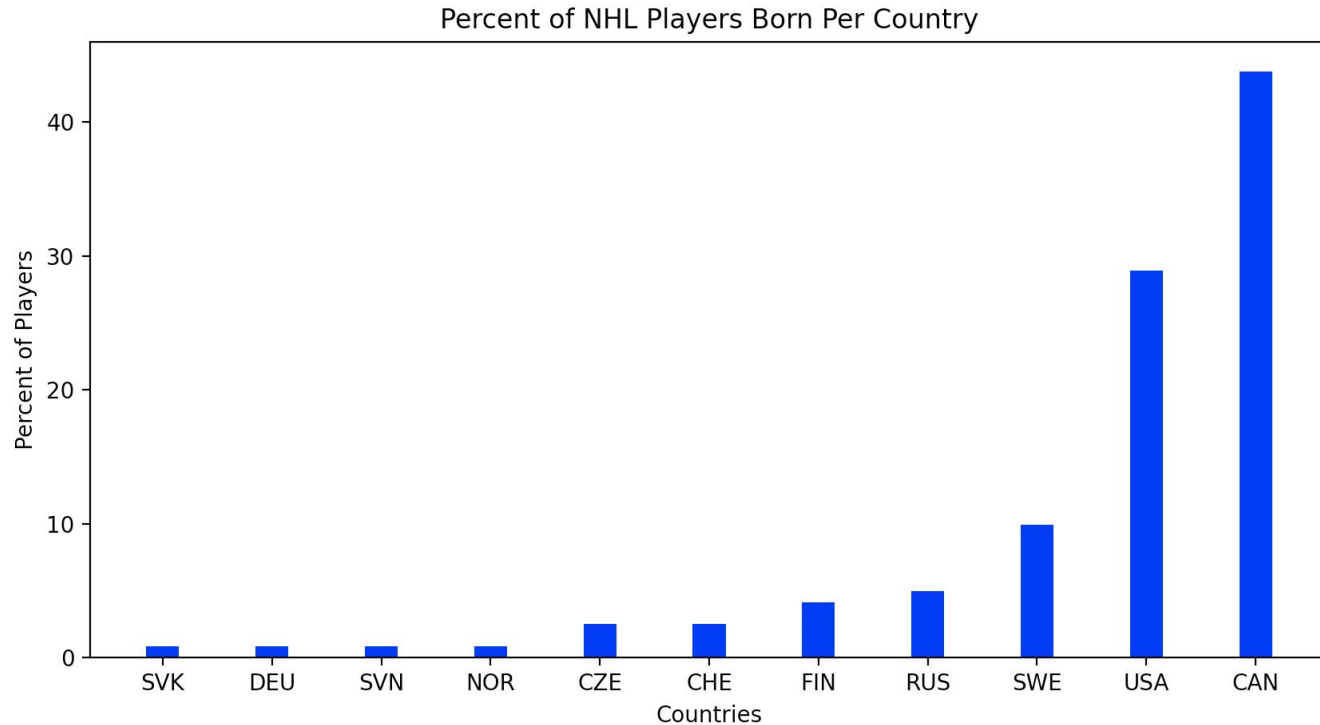
	name	points	birth_month	birth_place
98	Brendan Gallagher	204	9	2
99	Nino Niederreiter	202	9	7
100	Alex Killorn	201	9	2
101	Seth Jones	201	10	1
102	Brock Boeser	200	2	1
103	Anthony Mantha	199	9	2
104	Bryan Rust	199	5	1
105	Tyler Toffoli	198	4	2
106	Adam Henrique	196	2	2
107	Brandon Saad	194	10	1
108	Tyler Johnson	194	7	1
109	Zach Parise	194	7	1
110	Pavel Buchnevich	192	4	4
111	Jonathan Drouin	191	3	2
112	Jeff Petry	190	12	1
113	Jakob Silfverberg	190	10	3
114	Phillip Danault	189	2	2
115	Timo Meier	189	10	7
116	Yanni Gourde	188	12	2
117	Joe Thornton	187	7	2
118	Derek Stepan	187	6	1
119	Zach Hyman	184	6	2
120	Jordan Staal	182	9	2
121	Charlie Coyle	180	3	1

Visualizations

Visualization #1

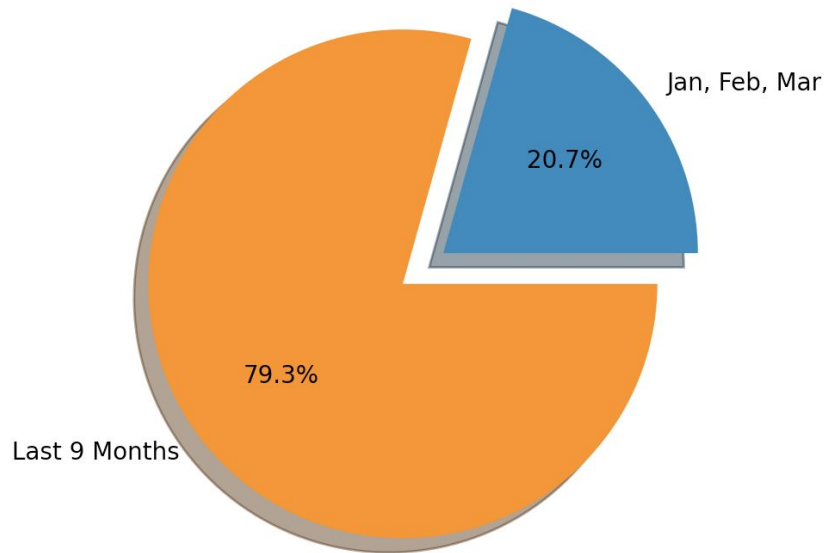


Visualization #2



Visualization #3

Percent of NHL Leaders Born in First Three Months





Code Documentation



Code Documentation

Part 1 → NHL API and NHL Stats Website Scraping and Data Collection

```
def player_info():
```

```
    """No inputs. Returns a list of tuples in which the contents are (player, point total). Does  
    this for 150 players."""
```

```
def search():
```

```
    """No inputs. Returns a list of tuples in which the contents are (player, birth month, country).  
    Does this for every single player in the league."""
```

```
def same_names():
```

```
    """No inputs. Returns a list of tuples in which the contents are (player, point total). Does  
    this only for players that coincide with both lists."""
```

```
def other_same_names():
```

```
    """No inputs. Returns a list of tuples in which the contents are (player, birth month,  
    country). Does this only for players that coincide with both lists."""
```

Code Documentation (cont'd)

Part 2 → Database Set Up and Inputting

```
def setUpDatabase(db_name):  
    """Takes the name of a database, a string, as an input. Returns the cursor and connection to the  
    database."""  
  
def setup_players_table(cur, conn):  
    """Takes the database cursor and connection as inputs. Returns nothing. Inserts into the table  
    all of the player names and their corresponding point totals over the last 5 years."""  
  
def set_up_country_table(cur, conn):  
    """Takes the database cursor and connection as inputs. Returns nothing. Inserts into the table  
    all of the player countries and a corresponding id number."""  
  
def setup_month_id(cur, conn):  
    """Takes the database cursor and connection as inputs. Returns nothing. Inserts into the table  
    all of the birth months and a corresponding id number."""  
  
def birth_info_table(cur, conn):  
    """Takes the database cursor and connection as inputs. Returns nothing. Inserts into the table  
    all of the player names, and both their corresponding birth months and birth countries."""  
  
def join_tables(cur, conn):  
    """Takes the database cursor and connection as inputs. Joining the tables in order to have all  
    data wanted in the same table in database."""
```

Code Documentation (cont'd)

Part 3 → Calculations and File Writing (1/3)

#CALCULATION 1

```
def return_top_ten_players():
```

```
    """Takes nothing as input. Returns the top 10 players and their point totals over the past 5
    years in the NHL."""
```

#CALCULATION 2

```
def return_average_points():
```

```
    """Takes nothing as input. Returns the average number of points of the top 10 players over the
    past 5 years in the NHL in a statement."""
```

#FILE 1

```
def write_data_to_file(filename):
```

```
    """Takes in a filename (string) as an input. Returns nothing. Creates a file and writes return
    value of the function return_top_ten_players() and return_average_points() to the file."""
```


Code Documentation (cont'd)

Part 3 → Calculations and File Writing (2/3)

#CALCULATION 3

```
def return_most_pop_country(cur, conn):
```

```
    """Takes the database cursor and connection as inputs. Returns the country where the majority of
    the top 100 players were born."""
```

#CALCULATION 4

```
def return_most_pop_month(cur, conn):
```

```
    """Takes the database cursor and connection as inputs. Returns the month that the majority of
    the top 100 players were born."""
```

#FILE 2

```
def write_data_to_file_2(filename, cur, conn):
```

```
    """Takes in a filename (string) as an input and the database cursor and connection as inputs.
    Returns nothing. Creates a file and writes return value of the function return_top_ten_players()
    and return_average_points() to the file."""
```

Code Documentation (cont'd)

Part 3 → Calculations and File Writing (3/3)

```
#CALCULATION 5
```

```
def return_first_three_months(cur, conn):
```

```
    """Takes in a filename (string) as an input and the database cursor and connection as inputs.  
    Returns the percentage of players that were born in the months of January, February, and  
    March."""
```

```
#FILE 3
```

```
def write_data_to_file_3(filename, cur, conn):
```

```
    """Takes in a filename (string) as an input. Returns nothing. Creates a file and writes return  
    value of the function return_top_ten_players() and return_average_points() to the file."""
```

```
def main():
```

```
    """Takes no inputs and returns nothing. Calls all of the functions in order to run the  
    project."""
```

Code Documentation (cont'd)

Part 4 → Visualizations Code

```
def main():  
    """Takes no inputs and returns nothing. Selects data from the database to create visualizations  
    that represent our data collected."""  
  
def func(pct, allvalues):  
    """Takes in two parameters pct and allvalues that create the percentages for the pie chart  
    and displays them on each section. Returns %."""
```

Resources

Resources

Date	Issue Description	Location of Resource	Result
April 17	Maxed out use of original API	Serpstack Website	We changed our API to receive the same information but using a different API altogether
April 20	IntegrityError: Unique constraint, was not inputting duplicate data	https://stackoverflow.com/questions/36518628/sqlite-3-integrityerror-unique-constraint-failed-when-inserting-a-value	Used "INSERT OR IGNORE" instead of Insert
April 21	OperationalError: near "index"	https://stackoverflow.com/questions/53919698/sqlite-3-operationalerror-near-index-syntax-error	Index was a keyword, so we did not used "id" instead as column title
April 22	How to update country names in 3rd API table to be their id numbers	https://stackoverflow.com/questions/1293330/how-can-i-do-an-update-statement-with-join-in-sql-server	Helped with UPDATE and SELECT statements
April 23	Our git repositories were out of sink and we wanted to force a git pull	https://stackoverflow.com/questions/1125968/how-do-i-force-git-pull-to-overwrite-local-files	This worked very effortlessly, and we were able to intertwine our files to where they needed to be.

Resources

Date	Issue Description	Location of Resource	Result
April 25	Indexing databases	https://www.w3schools.com/sql/sql_ref_create_index.asp	Learned to index, but Uche helped us with this, so we did not need the website link.
April 25	Force pushing in github when we have written in the same file on accident	https://evilmartians.com/chronicles/git-push---force-and-how-to-deal-with-it	This allowed us to overwrite one of our files, but this was fine because we needed to force a push. This resource gave us the answer we needed.
April 26	How to add percentages on a pie chart	https://www.geeksforgeeks.org/plot-a-pie-chart-in-python-using-matplotlib/	Yes, this gave a function to input percentages on each section of the pie chart.
April 26	How to call a function from a different file for visualizations	https://intellipaat.com/community/9924/call-a-function-from-another-file-in-python#:~:text=If%20you%20want%20to%20call,b)and%20you%20are%20done.	Yes, this allowed us to reference our functions in bettercombined.py, while creating visualizations based on these functions in visualizations.py.