1. Filtering Rows

The `tbl.where` method provides a powerful way to **filter tables down to only specific rows**. Previously, you've seen how `tbl.where` can be used to find an 'exact match'. In this section, you'll explore additional ways to use `tbl.where`.

Player	Minute	Opponent
Däbritz	42	Spain
Däbritz	29	South Africa
García	89	South Africa
Gauvin	46	Norway
Graham Hansen	5	South Korea
Gwinn	66	China
Henry	85	South Korea
Herlovsen	51	South Korea
Hermoso	69	South Africa
Hermoso	82	South Africa

The table `goals` above shows information about goals scored at the 2019 Women's World Cup.

a. Filter the `goals` table to contain only rows for goals scored after the 80th minute.

```
goals.where('Minute', are.above(80))
```

b. Filter the `goals` table to find all goals scored against countries with 'South' in their name.

```
goals.where('Opponent', are.containing('South'))
```

c. Narrow down the table to only goals scored in Group B games. Group B contained the countries Germany, Spain, China and South Africa.

```
goals.where('Opponent', are.contained_in(['Germany', 'Spain', 'China', 'South Africa'])
```

d. **Bonus:** Are all countries in Group B guaranteed to be in the table you generated in question (c)? Why or why not?

No. If a country in Group B did not concede any goals across all matches, then that country would not be recorded in the 'Opponents' column. The table generated in (c) contains only the teams in Group B that conceded at least one goal.

2. Grouping and Joining Tables

First Name	Last Name	Country	Position	Age	Caps	Club
Yanara	Aedo	Chile	FW	25	20	Valencia
Jonna	Andersson	Sweden	DF	26	41	Chelsea
Estefanía	Banini	Argentina	FW	28	32	Levante
Karen	Bardsley	England	GK	34	75	Manchester City
Agustina	Barroso	Argentina	DF	26	36	Madrid CFF
Florencia	Bonsegundo	Argentina	FW	25	34	Sporting Huelva
C.J.	Bott	New Zealand	DF	24	17	Vittsjö
Ruth	Bravo	Argentina	MF	27	18	Tacón
Morgan	Brian	USA	MF	26	82	Chicago Red Stars
Lucy	Bronze	England	DF	27	67	Lyon

The table `players` above contains statistics for each player who participated in the World Cup, including their country, club team, position, and their number of national team appearances ("caps"). Using the `players` table, you want to discover some more interesting stats about the World Cup.

a. First, you want to find the club team that sent the most players to the World Cup. Write an expression that will generate a table containing each unique club team and the number of players in the `players` table that play for that club. Sort the table from most number of players to least number of players.

```
players.group('Club').sort('count', descending = True)
```

b. Using grouping, determine the average number of caps for all players by their position.

```
players.group('Position', np.mean).select('Position', 'Caps mean')
```

OR

```
players.select('Position', 'Caps').group('Position', np.mean)
```

Unfortunately, James lost the data containing the scores for each World Cup game. However, as a clever data scientist you know how to reconstruct that data using both the `players` and `goals` tables.

c. First, join the `players` and `goals` table into one large table called `stats`. The `stats` table should contain one row for each goal scored, with each row containing all of the player information for the goalscorer.

```
stats = goals.join('Player', players, 'Last Name')
```

[Note the join order — in order to only have one row per goal, the `goals` table needs to be first in the `tbl.join` call.]

d. Using the `stats` table, determine the total number of goals each country scored at the world cup. *Hint:* use `.group`

```
stats.group('Country')
```

e. For each opponent a country faced, generate a table displaying the number of goals the country scored against that opponent.

```
stats.group(make_array('Country', 'Opponent'))
```

f. **Bonus:** Will this process (questions (a) and (c)) fully reconstruct the score lines for each game? *Hint:* how would own goals be recorded in the `goals` table?

No, this process cannot completely reconstruct the score lines for all games. If an own goal is scored in a game (i.e. a player scored a goal against their own team), then the 'Opponent' is recorded as that player's country, not the country that benefited from the own goal. Therefore, it is impossible to know in which game the own goal was scored.

3. Pivot Tables

Pivot tables take two columns in a table and display statistics about **unique combinations of the two column's values**. The unique values of one column of the original table become the new columns of the pivot table, while the unique values of the second column in the original table become the new rows of the pivot table. `tbl.pivot` also allows you to specify what values or statistics will be populated in the cells of the table.

a. Using the `stats` table from Question 2, create a pivot table with one column for each country and one row for each unique club team represented in the World Cup. The cells of the table should show the total number of caps of players for each combination of country and club team.

```
stats.pivot('Country', 'Club', 'Caps', np.sum)
```

4. Introduction to Writing Functions (Bonus)

Functions are an important tool in any programming language to help **compartmentalize code** and **manage complexity**. In Python, a custom function is defined using the keyword `def`.

a. The stadium announcer needs a script to read out for each player's introduction. Create a function called `intro` that takes in a player's country, first and last name, and club team and returns an introduction in the following format: "From [COUNTRY], please welcome [CLUB] player [FIRST] [LAST]". For example: "From USA, please welcome Seattle Reign FC player Megan Rapinoe".

```
def intro(country, club, first, last):
return "From " + country + ", please welcome " + club + " player " + first + " " +
last
```

[Second line should be indented]

b. Using the `tbl.apply` method, apply the `intro` function to the `players` table to return an array of all player introductions.

```
players.apply(intro, 'Country', 'Club', 'First Name', 'Last Name')
```