The next six columns represent a single checkbox question. The respondent checked off a series of boxes in response to the question, Which of the following Star Wars films have you seen? Please select all that apply.

The columns for this question are:

* Which of the following Star Wars films have you seen? Please select all that apply. - Whether or not the respondent saw Star Wars: Episode I The Phantom Menace.
* Unnamed: 4 - Whether or not the respondent saw Star Wars: Episode II Attack of the Clones.
* Unnamed: 5 - Whether or not the respondent saw Star Wars: Episode III Revenge of the Sith.
* Unnamed: 6 - Whether or not the respondent saw Star Wars: Episode IV A New Hope.
* Unnamed: 7 - Whether or not the respondent saw Star Wars: Episode V The Empire Strikes Back.
* Unnamed: 8 - Whether or not the respondent saw Star Wars: Episode VI Return of the Jedi.

For each of these columns, if the value in a cell is the name of the movie, that means the respondent saw the movie. If the value is NaN, the respondent either didn't answer or didn't see the movie. We'll assume that they didn't see the movie.

We'll need to convert each of these columns to a Boolean, then rename the column something more intuitive. We can convert the values the same way we did earlier, except that we'll need to include the movie title and NaN in the mapping dictionary.

For example, imagine we had this column series:



["Star Wars: Episode I The Phantom Menace", NaN, "Star Wars: Episode I The Phantom Menace"]

We could convert the values using this mapping dictionary:



{

   "Star Wars: Episode I The Phantom Menace": True,

   np.NaN: False

}

After calling the map() method on a series, the column should only contain the values True and False.

Next, we'll need to rename the columns to better reflect what they represent. We can use the [pandas.DataFrame.rename()](http://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.rename.html" \t "_blank)method on dataframes to accomplish this.

The df.rename() method works a lot like map(). We pass it a dictionary that maps the current column names to new ones:



star\_wars = star\_wars.rename(columns={

   "Which of the following Star Wars films have you seen? Please select all that apply.": "seen\_1"

})

The pandas.DataFrame.rename() method will only rename the columns we specify in the dictionary, and won't change the names of other columns. The code above will rename the Which of the following Star Wars films have you seen? Please select all that apply. column to seen\_1.

Instructions

* Convert each column above so that it only contains the values True and False.
  + You can select the column names more quickly by entering star\_wars.columns[3:9], rather than typing them out.
  + Be very careful with spacing when constructing your mapping dictionary! In the cells, Star Wars: Episode I The Phantom Menace has two spaces between the end of Episode I and the start of The Phantom, but this is not the case in Star Wars: Episode VI Return of the Jedi. Check the values in the cells carefully to make sure you use the appropriate spacing.
* Rename each of the columns above so the names are more intuitive. We recommend using seen\_1 to indicate whether the respondent saw Star Wars: Episode I The Phantom Menace, seen\_2 for Star Wars: Episode II Attack of the Clones, and so on.
* When you're finished, the columns should have intuitive names, along with True and False values that indicate whether the respondent saw each of the six *Star Wars* movies.