DATA 100: Vitamin 1 Solutions

February 14, 2019

1 Foreign Key

Which of the following are best represented using a foreign key column in a table of houses in which each row describes a house?

□ The most recent sale price

☑ The nearest school

☑ The house across the street

□ The size in square feet

Explanation: A foreign key is a column in one table that refers to the primary key in another or the same table. We can therefore imagine that the nearest school column is a foreign key relating to another table containing information on schools where each school is the primary key. Similarly, the column containing the house across the street is a foreign key relating to another row in the same table where houses are the primary key. Recent sale price and square footage do not relate to any other tables, they only describe the observations in the table of houses.

2 Value Counts

For the `elections` dataframe from class created by `pd.read_csv('elections.csv')`, which of the following expressions compute the same result as `elections['Party'].value_counts()`? Ignore differences in the index name or value order.

- □ `elections.groupby('Party').agg(sum)`
- □ `elections['Party'].groupby(sum)`
- ☑ `elections['Candidate'].groupby(elections['Party']).size()`
- ☑ `elections.groupby(elections['Party']).size()`
- □ None of these

Explanation: Try running this code for yourself!

3 Data Cleaning

The `to_json` method of a Pandas DataFrame object generates which of the following data formats:

- ☑ A list of lists, where each list contains the values in a row.
- ☑ A dictionary of dictionaries in which the column labels are keys and each corresponding value is the dictionary of entries in the column corresponding to a row key.
- ☑ A list of dictionaries in which each dictionary has column labels as keys and the entries in a row as values.

Explanation: Note that the original question was badly worded, and therefore carries no weight in your vitamin 2 grade. See the `to_json` documentation (specifically the `orient` argument).

4 Multi Index

For the `elections` dataframe from class created by `pd.read_csv('elections.csv')`, what's the number of different entries in the `%` column of the original dataframe that affect the result of the following expression?
`elections.groupby(['Party', 'Year']).mean()['%'][:,2016]`
Reminder: the data include election results from 1980 through 2016 of candidates who won at least 5% of the popular vote.

□ 1

☑ 1 for each political party whose candidate won at least 5% of the vote in 2016

□ 1 for each year

□ 1 for each year and each political party

□ 1 for each year and each political party in which that political party's candidate won at least 5% of the vote

Explanation: Since the observations in the original `elections` dataframe consist of election results for the candidates from each party who won at least 5% of the popular vote, and since each party can have no more than one candidate, taking the mean `%` of the groups based on `Party` and `Year` is identical to simply extracting the `%` of each candidate per year. The final piece of code, `[:, 2016]`, selects the `%` column associated with the 2016 election.

5 Primary Keys

Can a string valued column be a primary key?

☑ Yes

□ False

Explanation: Yes, relational database management systems allow primary keys to be string valued.