SurveyCTO main mistakes

# Variable types

* **select\_one** vs **select\_multiple**
  + Be careful to choose select\_one when only one option should be selected by the enum (e.g. yes/no, rating), and select\_multiple when multiple options should be possible (eg: list of all projects, etc.)
* **time** vs **integer**
  + time variable shows a clock to enter a time (for example, 3pm)
  + when asking for the time it takes to commute, we’re asking for a number (integer)
* **text** vs **note**
  + text variable shows a textbox for enumerators to enter a string answer (open-ended)
  + note variable shows a screen displaying information/text for the enumerator, without requesting any input/answer (for example, welcome or exit notes)

# Multiple questions using the same list of choices

In a survey, multiple questions can use the same list of choices. In this case, you should create only one list of choice, and re-use it for multiple questions. For example, if we ask for the rating of food for multiple days, the rating choices are always the same, and should be just one single choice list in the survey.

Remember the following points:

* The choice list doesn’t have to have the same name as the question – this is why it is possible to share a shame choice list among multiple questions
* You shouldn’t duplicate a choice list if it’s exactly the same across two questions, because if later on that list changes, it will be harder (and/or you will forget) to change it at multiple locations than at just one

# Relevance conditions

Relevance conditions alllows to specify that some questions should be skipped in some cases (we specify the cases for which the question is relevant). When combined with groups they allow to skip an entire group of question (or survey).

Sometimes, it is more tricky to identify questions that should be skipped, and you should always test you survey to check whether you have to add relevance conditions. In the test survey you coded, there were two instances were people forgot to add relevance conditions:

* Forgetting relevance conditions on different food questions
  + There were 5 food questions (on for each day), but without relevance condition, you will ask about all five days to everyone, regardless of how many days people spent in the training (which was an information the survey collected right before the food questions)
  + In this case you had to add a relevance condition like “${training\_days}>=1” for the question on day 1, “${training\_days}>=2” for the question on day 2, “${training\_days}>=3” for the question on day 3, etc. This way, if the person said they spent 2 days (${training\_days}=2), the questions on day 1 and on day 2 will be asked (2 >= 1 so the relevance condition for the day1 question will be true; 2>=2 so same for the relevance condition on day2) but the question on day 3 won’t be asked (2 < 3 so the condition on day3 will be false) and neither will the questions on day 4 and 5.
* Forgetting relevance on area question
  + When a question depends on the answers to a previous one (for example, in this survey, the question area depended on the answer to the question about the country) you need to make sure you think about all possible answers to the first question and their implication for the second question
  + In this case, they could answer “Other” for the first question (country), in which case the area question was NOT relevant, so you had to add something like “${origin\_country} != 99” in the area question to skip the area question if they had said “Other” for the country (then add another text-type (i.e. string) question to ask for “other” area in this case).