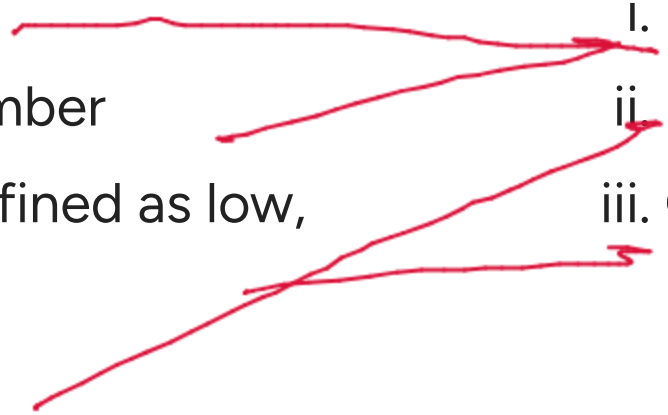
CSCI 1010 Class 4

Profs. Michael Linderman and Phil Chodrow
Department of Computer Science
Middlebury College

Variable description:	Best type:
1. Temperature	i. Non-categorical
2. Customer ID number	ii. Unordered categorical
3. Income level (defined as low, medium, high)	iii. Ordered categorical
4. Country code	



Match the variable description to the best type:

a. 1-i, 2-i, 3-i, 4-i

b. 1-i, 2-ii, 3-iii, 4-ii

c. 1-i, 2-i, 3-iii, 4-ii

d. 1-i, 2-ii, 3-iii, 4-iii

e. 1-iii, 2-ii, 3-iii, 4-ii

Slide 1 Notes

Answer: C

Temperature is a continuous variable (non-categorical), income level has a fixed, but ordered set of values (ordered categorical) and country code is a categorical variable with no inherent ordering. Customer ID is likely (hopefully) not drawn from a fixed set of values, so while it is discrete, it is best treated as non-categorical, e.g., as an integer or string.

	course	students	instructor	room
0	CSCI0145AF25	25	Briggs	75SHS102
1	CSCI0145BF25	30	Briggs	75SHS102
2	CSCI0146AF25	20	Caplan	75SHS102

Which, if any, of the 3 principles of tidy data is violated by this `DataFrame`?

a. No principles are violated; the data is tidy.

b. Each variable is not in its own column.

c. Each observation is not in its own row.

d. Each cell is not a single value.

Slide 2 Notes

Answer: D

Both the "course" and "room" columns contain multiple values per cell, e.g., each course value is really a department, course number, section and semester. To be tidy, and to permit analysis at any of those levels, we would need to split that column into multiple columns.

How would we handle multiple instructors per course? We could create a separate row for each instructor, but that would violate the principle that each observation is in its own row. Instead, we could create a separate `DataFrame` mapping courses to instructors and then use join/merge to combine the data as needed.

	id	year	month	element	d1	d2	d3	...	d25	d26	d27	d28	d29	d30	d31
0	MX17004	2010	1	tmax	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	27.8	NaN
1	MX17004	2010	1	tmin	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	14.5	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
20	MX17004	2010	12	tmax	29.9	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN
21	MX17004	2010	12	tmin	13.8	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Choose the best plan to create a tidy `DataFrame` for this data:

a. No changes needed; the data is already tidy.

b. “Melt” the day columns to lengthen the data, then convert month & day to integer day of year, (12/31 -> 365)

c. “Melt” the day columns to lengthen the data, then convert year, month & day to date type

d. “Melt” the day columns to lengthen the data, “pivot” the element column to widen the data, then convert day to integer, removing “d” prefix

e. “Melt” the day columns to lengthen the data, “pivot” the element column to widen the data, then convert year, month & day to date type

Slide 3 Notes

Answer: E

The data is not tidy because the variables are spread across the rows and columns. We want to have one row per observation (i.e., one row per station and date) and one column per variable (i.e., tmax, tmin). To do so, we first "melt" the day columns to lengthen the data, then "pivot" the element column to widen the data. Finally, we want to convert the year, month and day to the extended date type to enable downstream date operations.