

Introduction to Data Science

Data Wrangling

Joanna Bieri DATA101

Tidy Data

ASK

What makes a table *tidy*?

Give me one example of a table that is not.

Tidy Data

ANSWER

- Each **variable** is a column. Each **observation** is a row. Each kind of thing you measured gets its own table.
- Not tidy: a spreadsheet with merged cells for the year, a total row at the bottom, or two things crammed into one cell like “5 ft 10 in”.
- Most data you get is not tidy. Cleaning it up is the job, and you keep a record of every step.

Tidy Data

PREDICT

DF

	name	gender
0	Luke	masculine
1	Leia	feminine
2	R2-D2	masculine
3	Han	masculine

```
DF['gender'].value_counts()
```

What comes out? Is the table that comes out tidy?

Tidy Data

ANSWER

```
DF['gender'].value_counts()
```

gender	count
masculine	3
feminine	1

A count for each category. It is a nice clean table, but it is **summarized** data, not tidy raw data. You cannot get Luke back from a 3. Keep the raw data and never overwrite it.

Selecting Columns

REVERSE

DF

	hotel	lead_time	adults
0	Resort Hotel	342	2
1	City Hotel	7	1
2	Resort Hotel	85	2
3	City Hotel	13	2

I only want the hotel and lead_time columns. What is the command?

Selecting Columns

ANSWER

```
my_columns = ['hotel', 'lead_time']  
DF[my_columns]
```

	hotel	lead_time
0	Resort Hotel	342
1	City Hotel	7
2	Resort Hotel	85
3	City Hotel	13

A list of column names inside the square brackets. Two sets of brackets if you write it in one line: `DF[['hotel', 'lead_time']]`. Spelling has to be exact.

Sorting

PREDICT

DF

	hotel	lead_time	adults
0	Resort Hotel	342	2
1	City Hotel	7	1
2	Resort Hotel	85	2
3	City Hotel	13	2

```
DF[my_columns].sort_values('lead_time')
```

```
DF[my_columns].sort_values('lead_time', ascending=False)
```

What order do the rows come out in each case? What happens to the row labels 0, 1, 2, 3?

Sorting

ANSWER

ascending (the default)

	hotel	lead_time
1	City Hotel	7
3	City Hotel	13
2	Resort Hotel	85
0	Resort Hotel	342

ascending=False

	hotel	lead_time
0	Resort Hotel	342
2	Resort Hotel	85
3	City Hotel	13
1	City Hotel	7

Smallest first, or biggest first with the flag. The row labels travel with their rows, so 0 is still the 342 booking. Nothing in DF changed.

Sorting

PREDICT

DF

	hotel	month
0	Resort Hotel	July
1	City Hotel	April
2	Resort Hotel	March
3	City Hotel	August

```
DF.sort_values('month')
```

Words, not numbers. Does it error? If not, what order comes out?

Sorting

ANSWER

```
DF.sort_values('month')
```

	hotel	month
1	City Hotel	April
3	City Hotel	August
0	Resort Hotel	July
2	Resort Hotel	March

No error. It sorts alphabetically: April, August, July, March. Python does not know these are months. If you want calendar order you have to tell it the order yourself.

Sorting

ASK

You run `DF.sort_values('lead_time')` and see a sorted table.

Then you print DF again. Is it sorted?

Sorting

ANSWER

- No. `sort_values` hands you back a **new** sorted data frame and leaves DF alone.
- That is good news: you can experiment on the left without wrecking your raw data.
- If you want to keep the sorted version, give it a name: `DF_sorted = DF.sort_values('lead_time').copy()`

Boolean Masks

PREDICT

DF

	hotel	lead_time	adults
0	Resort Hotel	342	2
1	City Hotel	7	1
2	Resort Hotel	85	2
3	City Hotel	13	2

```
DF['hotel'] == 'Resort Hotel'
```

What comes out? How many rows? What is in each one?

Boolean Masks

ANSWER

```
DF['hotel'] == 'Resort Hotel'
```

	hotel
0	True
1	False
2	True
3	False

Four rows of True or False, one per observation. This is a **boolean mask**. The double equals is a question: is this equal to that? One equals sign would be an assignment.

Boolean Masks

REVERSE

DF

	hotel	lead_time	adults
0	Resort Hotel	342	2
1	City Hotel	7	1
2	Resort Hotel	85	2
3	City Hotel	13	2

I want only the rows for the Resort Hotel. What is the command?

Boolean Masks

ANSWER

```
mask = DF['hotel'] == 'Resort Hotel'  
DF[mask]
```

	hotel	lead_time	adults
0	Resort Hotel	342	2
2	Resort Hotel	85	2

Make the mask, then hand it to the data frame in square brackets. Only the True rows come back. The row labels 0 and 2 stay, the gaps are where the False rows were.

Logical Operators

PREDICT

```
'CAT' == 'cat'
```

```
3 < 10 and 10 < 3
```

```
42 in [3, 6, 18, 42]
```

```
2 <= 2 or 2 < 1
```

True or False for each line?

Logical Operators

ANSWER

<code>'CAT' == 'cat'</code>	<code>False</code>
<code>3 < 10 and 10 < 3</code>	<code>False</code>
<code>42 in [3, 6, 18, 42]</code>	<code>True</code>
<code>2 <= 2 or 2 < 1</code>	<code>True</code>

Python is 100% literal, so CAT and cat are different words. and needs both sides true. or needs one. in checks whether something is inside a list.

Logical Operators

REVERSE

DF

	country	arrival_date_year
0	PRT	2015
1	GBR	2016
2	PRT	2017
3	PRT	2015

I want only the bookings from PRT that arrived after 2015. What are the commands?

Logical Operators

ANSWER

```
mask_country = DF['country'] == 'PRT'  
DF_prt = DF[mask_country]
```

```
mask_year = DF_prt['arrival_date_year'] > 2015  
DF_prt[mask_year]
```

	country	arrival_date_year
2	PRT	2017

Two masks, one after the other. The second mask is built from the smaller frame, not from DF. This is the mini analysis in the homework.

Questions?

Questions?