CMPU 100 · Programming with Data

Grouping and Aggregation

Class 12

Where are we?

Cleaning and analyzing tabular data might involve

`select`ing / **`drop`**ping columns,

making a table with **`relabeled`** columns,

filtering to only rows **`where`** a predicate is true,

`apply`ing a function to a column, and

making a new table **`with_columns`** we create,

As well as visualizing the results with

horizontal bar charts (**`hbar`**),

scatter plots (**`scatter`**), and

line plots (**`plot`**).

But we also had an interesting table method show up last class – and again in lab…

Group

NBC News: “President Barack Obama receives a gift from Saudi King Abdullah… The large gold medallion was among several gifts given that day that were valued at $34,500 …”

Notebook: *Load and clean data*

Grouping values in a column

gifts

year	*country*	*value*
2009	Denmark	388
2009	Egypt	630
2009	Denmark	340
2010	Finland	445
2010	Egypt	356
2010	Egypt	380
2011	Denmark	485

```
gifts.group("year")
```

`gifts`

year	*country*	*value*
2009	Denmark	388
2009	Egypt	630
2009	Denmark	340
2010	Finland	445
2010	Egypt	356
2010	Egypt	380
2011	Denmark	485

```
gifts.group("year")
```

`gifts`

year	*country*	*value*
2009	Denmark	388
2009	Egypt	630
2009	Denmark	340
2010	Finland	445
2010	Egypt	356
2010	Egypt	380
2011	Denmark	485

`gifts.group("year")`

year	*count*
2009	3
2010	3
2011	1

gifts

year	*country*	*value*
2009	Denmark	388
2009	Egypt	630
2009	Denmark	340
2010	Finland	445
2010	Egypt	356
2010	Egypt	380
2011	Denmark	485

```
gifts.group("country")
```

`gifts`

year	*country*	*value*
2009	Denmark	388
2009	Egypt	630
2009	Denmark	340
2010	Finland	445
2010	Egypt	356
2010	Egypt	380
2011	Denmark	485

```
gifts.group("country")
```

`gifts`

year	*country*	*value*
2009	Denmark	388
2009	Egypt	630
2009	Denmark	340
2010	Finland	445
2010	Egypt	356
2010	Egypt	380
2011	Denmark	485

`gifts.group("country")`

country	*count*
Denmark	3
Egypt	3
Finland	1

Ways to summarize values in a group

`gifts`

year	*country*	*value*
2009	Denmark	388
2009	Egypt	630
2009	Denmark	340
2010	Finland	445
2010	Egypt	356
2010	Egypt	380
2011	Denmark	485

```
gifts.group("year", sum)
```

`gifts`

year	*country*	*value*
2009	Denmark	388
2009	Egypt	630
2009	Denmark	340
2010	Finland	445
2010	Egypt	356
2010	Egypt	380
2011	Denmark	485

```
gifts.group("year", sum)
```

The aggregation function

Aggregation is a process in which information is gathered and expressed in collective or summary form.

gifts

year	*country*	*value*
2009	Denmark	388
2009	Egypt	630
2009	Denmark	340
2010	Finland	445
2010	Egypt	356
2010	Egypt	380
2011	Denmark	485

```
gifts.group("year", sum)
```

`gifts`

year	*country*	*value*
2009	Denmark	388
2009	Egypt	630
2009	Denmark	340
2010	Finland	445
2010	Egypt	356
2010	Egypt	380
2011	Denmark	485

`gifts.group("year", sum)`

year	*country sum*	*value sum*
2009		`sum([388, 630, 340])`
2010		`sum([445, 356, 380])`
2011		`sum([485])`

`gifts`

year	*country*	*value*
2009	Denmark	388
2009	Egypt	630
2009	Denmark	340
2010	Finland	445
2010	Egypt	356
2010	Egypt	380
2011	Denmark	485

We can't sum strings, so it's empty

`gifts.group("year", sum)`

year	*country sum*	*value sum*
2009		`sum([388, 630, 340])`
2010		`sum([445, 356, 380])`
2011		`sum([485])`

`gifts`

year	*country*	*value*
2009	Denmark	388
2009	Egypt	630
2009	Denmark	340
2010	Finland	445
2010	Egypt	356
2010	Egypt	380
2011	Denmark	485

We can't sum strings, so it's empty

`gifts.group("year", sum)`

year	*country sum*	*value sum*
2009		`sum([388, 630, 340])`
2010		`sum([445, 356, 380])`
2011		`sum([485])`

year	*country sum*	*value sum*
2009		1358
2010		1181
2011		485

`gifts`

year	*country*	*value*
2009	Denmark	388
2009	Egypt	630
2009	Denmark	340
2010	Finland	445
2010	Egypt	356
2010	Egypt	380
2011	Denmark	485

```
gifts.group("country", max)
```

`gifts`

year	*country*	*value*
2009	Denmark	388
2009	Egypt	630
2009	Denmark	340
2010	Finland	445
2010	Egypt	356
2010	Egypt	380
2011	Denmark	485

```
gifts.group("country", max)
```

`gifts`

year	*country*	*value*
2009	Denmark	388
2009	Egypt	630
2009	Denmark	340
2010	Finland	445
2010	Egypt	356
2010	Egypt	380
2011	Denmark	485

`gifts.group("country", max)`

country	*year max*	*value max*
Denmark	`max([2009, 2009, 2011])`	`max([388, 340, 485])`
Egypt	`max([2009, 2010, 2011])`	`max([630, 356, 380])`
Finland	`max([2010])`	`max([445])`

`gifts`

year	*country*	*value*
2009	Denmark	388
2009	Egypt	630
2009	Denmark	340
2010	Finland	445
2010	Egypt	356
2010	Egypt	380
2011	Denmark	485

`gifts.group("country", max)`

country	*year max*	*value max*
Denmark	`max([2009, 2009, 2011])`	`max([388, 340, 485])`
Egypt	`max([2009, 2010, 2011])`	`max([630, 356, 380])`
Finland	`max([2010])`	`max([445])`

country	*year max*	*value max*
Denmark	2011	485
Egypt	2010	630
Finland	2010	445

`gifts`

year	*country*	*value*
2009	Denmark	388
2009	Egypt	630
2009	Denmark	340
2010	Finland	445
2010	Egypt	356
2010	Egypt	380
2011	Denmark	485

`gifts.group("country", max)`

country	*year max*	*value max*
Denmark	`max([2009, 2009, 2011])`	`max([388, 340, 485])`
Egypt	`max([2009, 2010, 2011])`	`max([630, 356, 380])`
Finland	`max([2010])`	`max([445])`

country	*year max*	*value max*
Denmark	2011	485
Egypt	2010	630
Finland	2010	445

Computes max for each column separately.

Results may not always be useful!

Grouping values in multiple columns

gifts

year	*country*	*value*
2009	Denmark	388
2009	Egypt	630
2009	Denmark	340
2010	Finland	445
2010	Egypt	356
2010	Egypt	380
2011	Denmark	485

```
gifts.group(
    ["year", "country"],
    sum
)
```

gifts

year	*country*	*value*
2009	Denmark	388
2009	Egypt	630
2009	Denmark	340
2010	Finland	445
2010	Egypt	356
2010	Egypt	380
2011	Denmark	485

```
gifts.group(
    ["year", "country"],
    sum
)
```

`gifts`

year	*country*	*value*
2009	Denmark	388
2009	Egypt	630
2009	Denmark	340
2010	Finland	445
2010	Egypt	356
2010	Egypt	380
2011	Denmark	485

```
gifts.group(
    ["year", "country"],
    sum
)
```

year	*country*	*value sum*
2009	Denmark	`sum([388, 340])`
2009	Egypt	`sum([630])`
2010	Egypt	`sum([356, 380])`
2010	Finland	`sum([445])`
2011	Denmark	`sum([485])`

gifts

year	*country*	*value*
2009	Denmark	388
2009	Egypt	630
2009	Denmark	340
2010	Finland	445
2010	Egypt	356
2010	Egypt	380
2011	Denmark	485

```
gifts.group(
    ["year", "country"],
    sum
)
```

year	*country*	*value sum*
2009	Denmark	728
2009	Egypt	630
2010	Egypt	736
2010	Finland	445
2011	Denmark	485

Pivot

`gifts`

year	*country*	*value*
2009	Denmark	388
2009	Egypt	630
2009	Denmark	340
2010	Finland	445
2010	Egypt	356
2010	Egypt	380
2011	Denmark	485

```
gifts.pivot("country", "year")
```

gifts

year	*country*	*value*
2009	Denmark	388
2009	Egypt	630
2009	Denmark	340
2010	Finland	445
2010	Egypt	356
2010	Egypt	380
2011	Denmark	485

```
gifts.pivot("country", "year")
```

`gifts`

year	*country*	*value*
2009	Denmark	388
2009	Egypt	630
2009	Denmark	340
2010	Finland	445
2010	Egypt	356
2010	Egypt	380
2011	Denmark	485

`gifts.pivot("country", "year")`

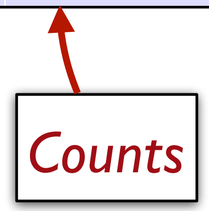
year	*Denmark*	*Egypt*	*Finland*
2009			
2010			
2011			

`gifts`

year	*country*	*value*
2009	Denmark	388
2009	Egypt	630
2009	Denmark	340
2010	Finland	445
2010	Egypt	356
2010	Egypt	380
2011	Denmark	485

```
gifts.pivot("country", "year")
```

year	*Denmark*	*Egypt*	*Finland*
2009	2	1	0
2010	0	2	1
2011	1	0	0



Pivot tables make it easier to quickly summarize in a grid data that has been grouped by two variables.

`gifts`

year	*country*	*value*
2009	Denmark	388
2009	Egypt	630
2009	Denmark	340
2010	Finland	445
2010	Egypt	356
2010	Egypt	380
2011	Denmark	485

```
gifts.pivot("year", "country")
```

`gifts`

year	*country*	*value*
2009	Denmark	388
2009	Egypt	630
2009	Denmark	340
2010	Finland	445
2010	Egypt	356
2010	Egypt	380
2011	Denmark	485

```
gifts.pivot("year", "country")
```

`gifts`

year	*country*	*value*
2009	Denmark	388
2009	Egypt	630
2009	Denmark	340
2010	Finland	445
2010	Egypt	356
2010	Egypt	380
2011	Denmark	485

`gifts.pivot("year", "country")`

country	*2009*	*2010*	*2011*
Denmark			
Egypt			
Finland			

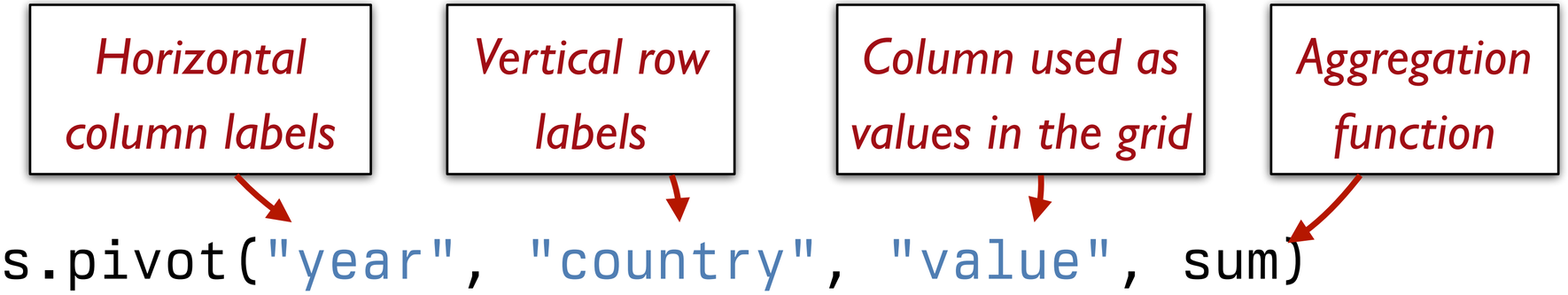
`gifts`

year	*country*	*value*
2009	Denmark	388
2009	Egypt	630
2009	Denmark	340
2010	Finland	445
2010	Egypt	356
2010	Egypt	380
2011	Denmark	485

`gifts.pivot("year", "country")`

country	*2009*	*2010*	*2011*
Denmark	2	0	1
Egypt	1	2	0
Finland	0	1	0

Pivot with four parameters



year	country	value
2009	Denmark	388
2009	Egypt	630
2009	Denmark	340
2010	Finland	445
2010	Egypt	356
2010	Egypt	380
2011	Denmark	485

Horizontal column labels

Vertical row labels

Column used as values in the grid

Aggregation function

```
gifts.pivot("year", "country", "value", sum)
```

year	*country*	*value*
2009	Denmark	388
2009	Egypt	630
2009	Denmark	340
2010	Finland	445
2010	Egypt	356
2010	Egypt	380
2011	Denmark	485

country	*2009*	*2010*	*2011*
Denmark	388 + 340	0	485
Egypt	630	356 + 380	0
Finland	0	445	0

Horizontal column labels

Vertical row labels

Column used as values in the grid

Aggregation function

```
gifts.pivot("year", "country", "value", sum)
```

year	*country*	*value*
2009	Denmark	388
2009	Egypt	630
2009	Denmark	340
2010	Finland	445
2010	Egypt	356
2010	Egypt	380
2011	Denmark	485

country	*2009*	*2010*	*2011*
Denmark	728	0	485
Egypt	630	736	0
Finland	0	445	0

Group vs pivot

Consider: When is it an advantage to use **`pivot`** over **`group`** and vice versa?

```
gifts.group(
    ["year", "country"],
    sum
)
```

year	*country*	*value sum*
2009	Denmark	728
2009	Egypt	630
2010	Egypt	736
2010	Finland	445
2011	Denmark	485

```
gifts.pivot(
   "year", "country", "value",
    sum
)
```

country	*2009*	*2010*	*2011*
Denmark	728	0	485
Egypt	630	736	0
Finland	0	445	0

Notebook: *Back to Greenland*

Acknowledgments

This class incorporates material from:

Stephen Freund and Katie Keith, Williams College

Data 6, University of California, Berkeley