

For the birds

Selecting and sorting data



CS112 Scientific Computation
Department of Computer Science
Wellesley College

A bird database

Name	Family	Habitat	Size	Wingspan	BackColor	UnderColor	HeadColor	Spotted	Comment
GreatEgret	heron	marshes	39	51	white	white	white	no	longLegs
GreatBlueHeron	heron	marshes	48	72	blue/gray	black	black/white	no	longLegs
SnowyEgret	heron	marshes/ponds	24	38	white	white	white	no	longLegs
GreenHeron	heron	marshes/ponds	19	25	green	brown	black	no	darkBill
AmericanBittern	heron	marshes	28	45	brown	brown	brown	no	greenLegs
MuteSwan	waterfowl	lagoons/ponds	60	96	white	white	white	no	orangeBill
CanadaGoose	waterfowl	wetlands/fields	40	72	brown	brown	black/white	no	fliesInV
SnowGoose	waterfowl	marshes/fields	28	58	white	white	white	no	orangeBill
AmericanBlackDuck	waterfowl	wetlands	23	36	brown	brown	brown	no	yellowBill
NorthernPintail	waterfowl	marshes/ponds	26	35	gray	white	brown	no	longTail
Mallard	waterfowl	wetlands	24	36	brown	gray	green	no	purpleChest
BlueWingedTeal	waterfowl	marshes/ponds	15	24	brown	brown	blue/gray	yes	blueShoulders
Osprey	hawk/eagle	wetlands	23	72	brown	white	brown/white	yes	whiteLegs
BaldEagle	hawk/eagle	wetlands	32	80	brown	brown	white	no	yellowBeak
NorthernHarrier	hawk/eagle	marshes/fields	22	54	gray	gray	brown	no	longTail
CoopersHawk	hawk/eagle	woods	17	28	gray	white	gray	no	orangeLegs
RedTailedHawk	hawk/eagle	woods/fields	22	58	brown	white	brown	yes	orangeTail
PeregrineFalcon	hawk/eagle	marshes/cities	18	44	gray	gray	black/white	no	sideburn
...									

Suppose you'd like to **select** all the herons,
select the birds with giant wingspans,
sort by size or wingspan,
sort alphabetically by name...



Let's load it in and see what we have

```
% loadBirds.m
fid = fopen('birds.txt');
birds = textscan(fid, '%s %s %s %u %u %s %s %s %s %s',
                'headerlines', 1);

fclose(fid);

function printBirdInfo (birds)
% prints out all the information stored in the input cell array
for i = 1:length(birds{1})
    disp(sprintf('%22s %10s %17s %3u %3u %12s %12s %4s %17s', ...
                birds{1}{i}, birds{2}{i}, birds{3}{i}, birds{4}{i}, birds{5}{i}, ...
                birds{6}{i}, birds{7}{i}, birds{8}{i}, birds{9}{i}, birds{10}{i}))
end
```

3

Selecting all the herons

```
function herons = getHerons (birds)
% find the indices of all birds from the heron family
indices = find(strcmp(birds{2}, 'heron'));
% create an empty cell array and fill it with all the
% information from the heron family
herons = cell(1,10);
for i = 1:10
    herons{i} = birds{i}(indices);
end
```

Exercise: select birds with large
wingspans (> 48")



4

MATLAB sort function

```
>> nums = [7 2 9 7 8 3 6 1 3 4];
>> sortNums = sort(nums)
sortNums =
    1  2  3  3  4  6  7  7  8  9
>> sortNums = sort(nums, 'descend')
sortNums =
    9  8  7  7  6  4  3  3  2  1
>> [sortNums sortIndices] = sort(nums, 'ascend')
sortNums =
    1  2  3  3  4  6  7  7  8  9
sortIndices =
    8  2  6  9 10  7  1  4  5  3
```

Exercise: What does the expression `nums(sortIndices)` return?

5

Now let's sort a cell array of strings

```
>> words = {'early' 'cloud' 'heights' 'a' 'black' 'great' 'from' 'descended'};
>> sortWords = sort(words)
sortWords =
    'a' 'black' 'cloud' 'descended' 'early' 'from' 'great' 'heights'
>> words = {'early' 'Cloud' 'heights' 'A' 'black' 'Great' 'from' 'Descended'};
>> [sortWords sortIndices] = sort(words)
sortWords =
    'A' 'Cloud' 'Descended' 'Great' 'black' 'early' 'from' 'heights'
sortIndices =
    4  2  8  6  5  1  7  3
```

Hmmm.... What's going on here???



6

Remember the ASCII code?

When comparing the order of two strings MATLAB uses the order of characters in the **ASCII code** in which *all capital letters appear before all lowercase letters*

Exercise: Write a function that sorts a cell array of words alphabetically, independent of capitalization

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00	Null	32	20	Space	64	40	@	96	60	ˆ
1	01	Start of heading	33	21	!	65	41	A	97	61	a
2	02	Start of text	34	22	"	66	42	B	98	62	b
3	03	End of text	35	23	#	67	43	C	99	63	c
4	04	End of transmit	36	24	\$	68	44	D	100	64	d
5	05	Enquiry	37	25	%	69	45	E	101	65	e
6	06	Acknowledge	38	26	&	70	46	F	102	66	f
7	07	Audible bell	39	27	'	71	47	G	103	67	g
8	08	Backspace	40	28	(72	48	H	104	68	h
9	09	Horizontal tab	41	29)	73	49	I	105	69	i
10	0A	Line feed	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage return	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	47	2F	/	79	4F	O	111	6F	o
16	10	Data link escape	48	30	0	80	50	P	112	70	p
17	11	Device control 1	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	50	32	2	82	52	R	114	72	r
19	13	Device control 3	51	33	3	83	53	S	115	73	s
20	14	Device control 4	52	34	4	84	54	T	116	74	t
21	15	Neg. acknowledge	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	54	36	6	86	56	V	118	76	v
23	17	Endtrans. block	55	37	7	87	57	W	119	77	w
24	18	Cancel	56	38	8	88	58	X	120	78	x
25	19	End of medium	57	39	9	89	59	Y	121	79	y
26	1A	Substitution	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	59	3B	;	91	5B	[123	7B	{
28	1C	File separator	60	3C	<	92	5C	\	124	7C	
29	1D	Group separator	61	3D	=	93	5D]	125	7D	}
30	1E	Record separator	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	63	3F	?	95	5F	_	127	7F	ˆ

7

Sorting the bird data

```
function sortedData = sortByWingspan (birds)
% sort the bird information by wingspan
[temp indices] = sort(birds{5});
% create an empty cell array and fill it with all the bird
% information in sorted order
sortedData = cell(1,10);
for i = 1:10
    sortedData{i} = birds{i}(indices);
end
```



Exercise: sort the birds alphabetically by name

8

On the flip side...

Suppose we have some nice, orderly information that we want to *scramble*

```
>> order = randperm(10)
```

```
order =
```

```
6 2 5 1 4 8 10 3 7 9
```

```
>> conditions = [2.0 -2.0 2.5 -2.5 3.0 -3.0 3.5 -3.5 4.0 -4.0];
```

```
>> newConditions = conditions(order)
```

```
newConditions =
```

```
-3.0 -2.0 3.0 2.0 -2.5 -3.5 -4.0 2.5 3.5 4.0
```



Exercise: write a function `makeAnagram` that has an input string and returns an anagram of the string