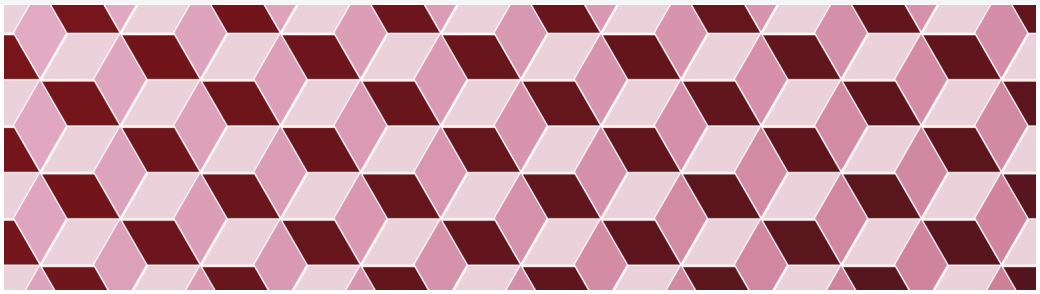


15-104 Introduction to Computing for Creative Practice

Fall 2020

34 Using Data Files

Instructor: Tom Cortina, tcortina@cs.cmu.edu, GHC 4117, 412-268-3514

1

Tabular Data

- Table data is commonly stored in a plain-text file referred to as a comma-separated values (csv) file.
- Data from such files can be stored as an object in your program for use in drawing.
- Data is indexed with a row and column number, starting from 0.

(0,0)	(0,1)	(0,2)	(0,3)	(0,4)
(1,0)	(1,1)	(1,2)	(1,3)	(1,4)
(2,0)	(2,1)	(2,2)	(2,3)	(2,4)
(3,0)	(3,1)	(3,2)	(3,3)	(3,4)

4 rows

5 columns

2

Example: Plotting Data

Requires a local server.

```
var stats;
var numRows;
function preload() {
    stats = loadTable("ortiz.csv");
}
function setup() {
    createCanvas(480, 120);
    numRows = stats.getRowCount();
}
function draw() {
    background(200);
    drawGrid();
    drawGraph();
    noLoop();
}
```

Don't put this line in preload!

year, homeruns, rbis, average

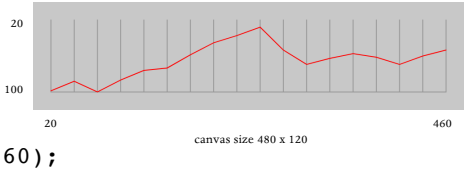
1997	1	6	0.327
1998	9	46	0.277
1999	0	0	0
2000	10	63	0.282
2001	18	48	0.234
2002	20	75	0.272
2003	31	101	0.288
2004	41	139	0.301
2005	47	148	0.3
2006	54	137	0.287
2007	35	117	0.332
2008	23	89	0.264
2009	28	99	0.238
2010	32	102	0.27
2011	29	96	0.309
2012	23	60	0.318
2013	30	103	0.309
2014	35	104	0.263

3

Example: Plotting Data (cont'd)

```
function drawGrid() {
    stroke(150); line(20, 100, 20, 20); line(20, 100, 460, 100);
    for (var i = 0; i < numRows; i++) {
        var x = map(i, 0, numRows-1, 20, 460);
        line(x, 20, x, 100);
    }
}
function drawGraph() {
    noFill(); stroke(255, 0, 0);
    beginShape();
    for (var i = 0; i < numRows; i++) {
        var x = map(i, 0, numRows-1, 20, 460);
        var y = map(stats.getNum(i, 1), 0, 60, 100, 20);
        vertex(x,y);
    }
    endShape();
}
```

map row number to value between 20 and 460 for x coordinate



canvas size 480 x 120

map homeruns value (row 1 in table) from the range 0 to 60 to the range 100 to 20.

4

Tables with Headers

- In some csv files, the first row is a header with column titles.
- We can use these titles to access column values rather than the column number.

zip	state	city	lat	lng
35004	AL	Acmar	33.584132	-86.51557
35005	AL	Adamsville	33.588437	-86.959727
35006	AL	Adger	33.434277	-87.167455
35007	AL	Keystone	33.236868	-86.812861
35010	AL	New Site	32.941445	-85.951086
35014	AL	Alpine	33.331165	-86.208934
35016	AL	Arab	34.328339	-86.489638
35019	AL	Baileyton	34.268298	-86.621299
35020	AL	Bessemer	33.409002	-86.947547
35023	AL	Hueytown	33.414625	-86.999607
35031	AL	Blountsville	34.092937	-86.568628
35033	AL	Bremen	33.973664	-87.004281
35034	AL	Brent	32.93567	-87.211387

5

Example: Drawing City Lights

Requires a local server.

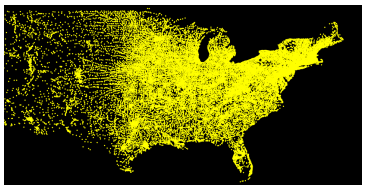
```
var cities;
var numRows;

function preload() {
    cities = loadTable("cities.csv", "header");
}

function setup() {
    createCanvas(480, 240);
    fill(255, 150, 0);
    noStroke();
    numRows = cities.getRowCount();
}
```

note the extra parameter

6

Example: Drawing City Lights (cont'd)

```
function draw() {
    background(0);
    var xoffset = map(mouseX, 0, width, -width*3, -width);
    translate(xoffset, -375); scale(7);
    for (var i = 0; i < numRows; i++) {
        var latitude = cities.getNum(i, "lat");
        var longitude = cities.getNum(i, "lng");
        setXY(latitude, longitude);
    }
}
function setXY(lat, lng) {
    var x = map(lng, -180, 180, 0, width);
    var y = map(lat, 90, -90, 0, height);
    ellipse(x, y, 0.25, 0.25);
}
```



7

JavaScript Object Notation (JSON)

- JSON is another format that stores objects in textual form by indicating their fields and corresponding values.

```
[
  {
    "title": "Breathless",
    "director": "Jean-Luc Godard",
    "year": 1960,
    "rating": 8.0
  },
  {
    "title": "Le Petit Soldat",
    "director": "Jean-Luc Godard",
    "year": 1960,
    "rating": 7.2
  },
  {
    "title": "A Woman Is a Woman",
    "director": "Jean-Luc Godard",
    "year": 1961,
    "rating": 7.7
  },
   ...etc...
]
```

8

Example: Film Ratings

Requires a local server.

```
var films = [];
var data;

function preload() {
    data = loadJSON("films.json");
}

function setup() {
    createCanvas(480, 120);
    for (obj in data) {
        films.push(data[obj]);
    }
}
```

append each object in the JSON data to the films array

9

Example: Film Ratings


```
function draw() {
    background(0);
    for (var i = 0; i < films.length; i++) {
        display(films[i], i*32 + 32, 105);
    }
    noLoop();
}
function display(film, x, y) {
    var ratingGray = map(film.rating, 6.5, 8.1, 102, 255);
    push();
    translate(x, y); rotate(-QUARTER_PI);
    fill(ratingGray); text(film.title, 0, 0);
    pop();
}
```

10