

LABORATORY OF DATA SCIENCE

Data Access: Files

Two issues

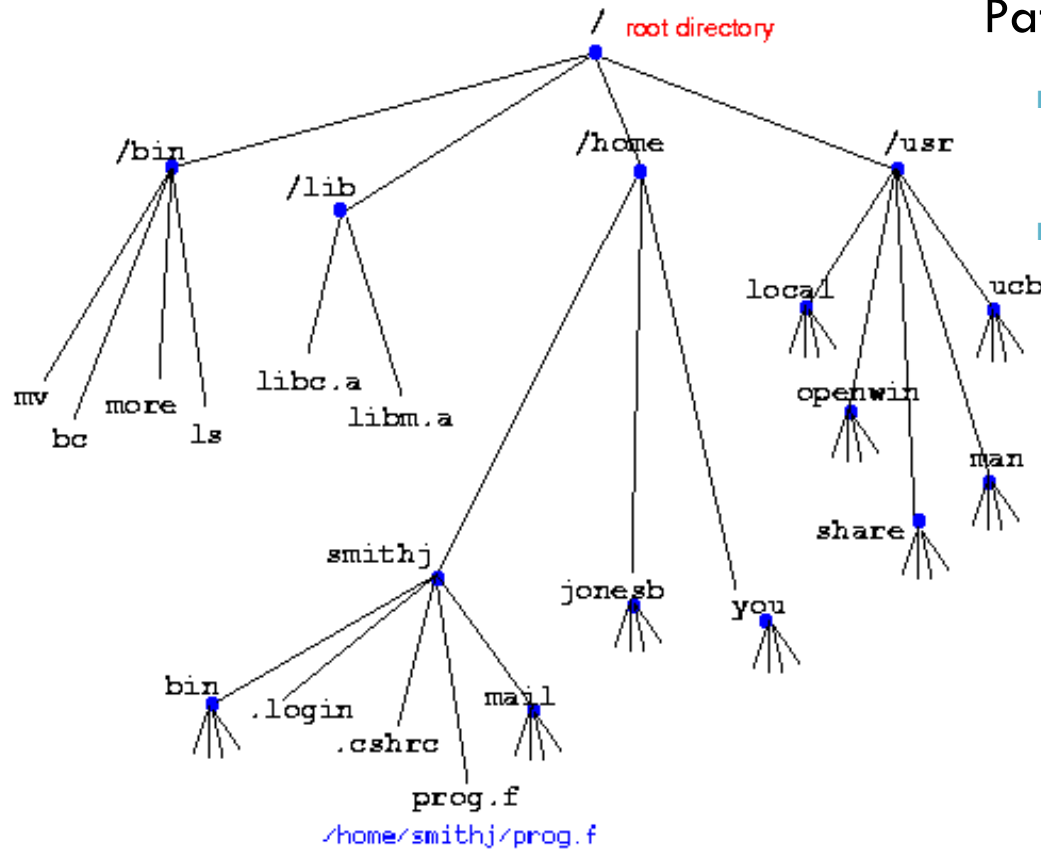
2

- **Where** are my files?
 - Local file systems
 - Distributed file systems
 - Network protocols

- Which **format** is data in?
 - Text
 - CSV, ARFF
 - XML
 - Binary, Compressed, ...

Local file system

3



Path of a **resource**

■ Windows:

■ C:\Program Files\Office\sample.doc

■ Linux:

■ /usr/home/r/ruggieri/sample.txt

Local file system

4

A logical abstraction of persistent mass memory

- hierarchical view (tree of directories and files)
- types of resources (file, directory, pipe, link, special)
- resource attributes (owner, rights, hard links)
- services (indexing, journaling)

Sample file system:

- Windows
 - NTFS, FAT32
- Linux
 - EXT2, EXT3, JFS, XFS, REISERFS, FAT32

Disk file systems [\[edit\]](#)

Disk file systems are usually block-oriented. Files in a

- **ADFS** – Acorn's Advanced Disc filing system, such as on the Acorn RISC PC
- **AdvFS** - Advanced File System, designed by Digital Equipment Corporation
- **AFS** (Not to be confused with Andrew File System)
- **AFS** - Ami File Safe, a commercial file system shipped with AmigaOS
- **AosFS** - File System used by the Oberon and A2000
- **AthFS** - AtheOS File System, a 64-bit [journaled](#) file system
- **BFS** - the Boot File System used on System V release 3
- **BFS** – the Be File System used on BeOS, occasionally on Linux
- **Btrfs** - is a [copy-on-write](#) file system for Linux announced in 2007
- **CBMFS** – The filesystem used on most Commodore 64 computers
- **CMDFS** – A filesystem extension added to CBMFS
- **CP/M** file system — Native filesystem used in the CP/M operating system
- **DDFS** – Data Domain File System, the [data deduplication](#) filesystem
- **DTFS** – Desktop File System, featuring file compression
- **DOS 3.x** - Original floppy operating system and file system
- **EAFS** – Extended Acer Fast Filesystem, used on Acer Aspire notebooks
- **Extent File System** (EFS) – an older block filing system
- **ext** – Extended file system, designed for Linux systems
- **ext2** – Second extended file system, designed for Linux
- **ext3** – A [journaled](#) form of ext2.
- **ext4** – A follow up for ext3 and also a [journaled](#) file system
- **ext3cow** – A [versioning file system](#) form of ext3.
- **FAT** – File Allocation Table, used on DOS and Microsoft Windows
 - **VFAT** – Optional layer on Microsoft Windows
 - **FATX** – A modified version of Microsoft Windows
- **FFS** (Amiga) – Fast File System, used on Amiga systems
- **FFS** – Fast File System, used on *BSD systems

Local file system

5

Physical view

□ Disk partition

- collection of contiguous blocks on a disk

□ File system driver

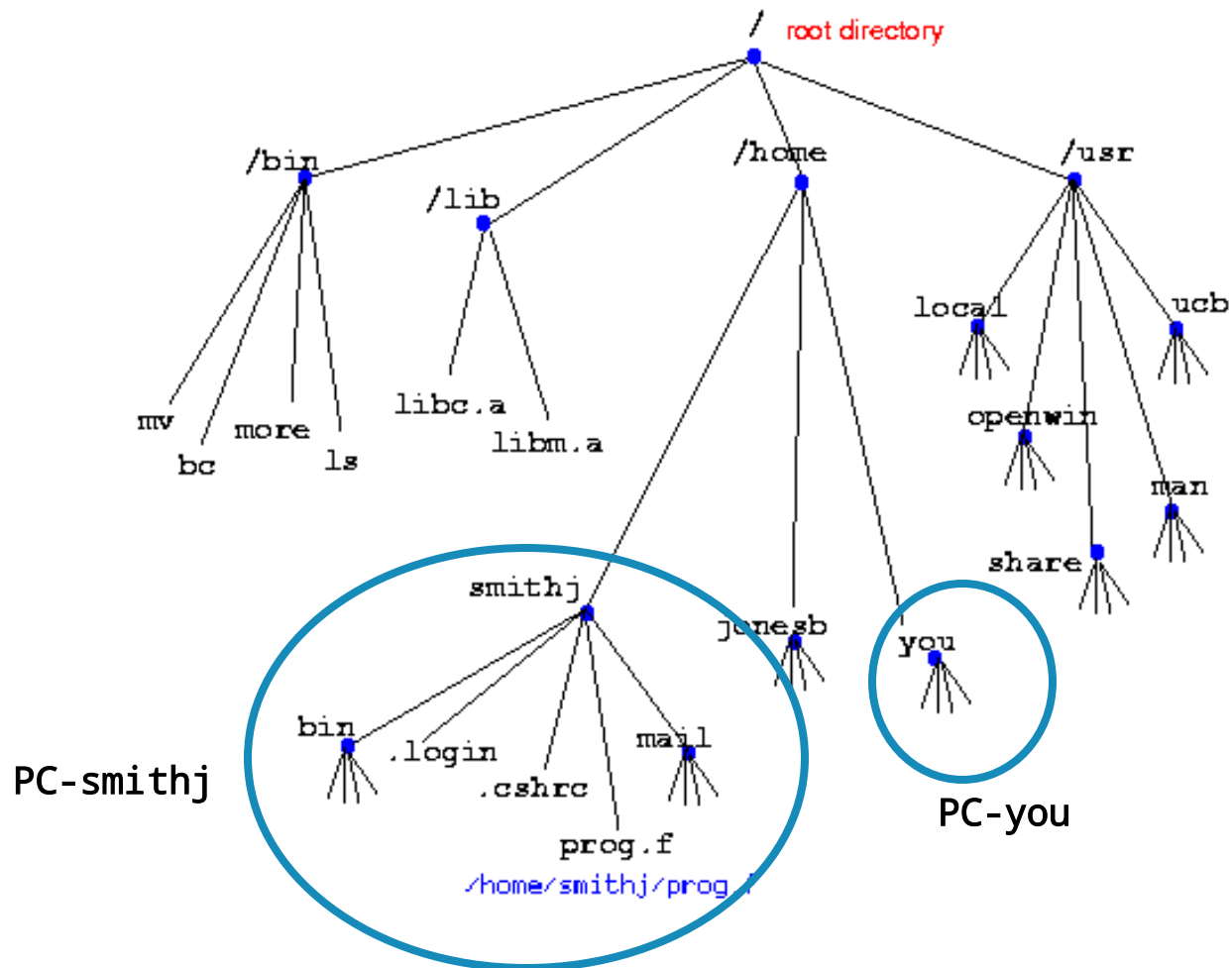
- software abstracting a file system on a partition
- Maps a file system to each partition

□ Mount

- starting a file system driver on a partition
- Windows (start up typically is automatic):
 - at startup for NTFS and FAT partitions
 - names of partitions: A: ... Z:
- Linux
 - at startup for partitions in `/etc/fstab`
 - `> mount -t ext3 /dev/hda2 /mnt/mydisk`

Distributed file system

6



Distributed file system

7

Acts as a client for a remote file access protocol

- logical abstraction of remote persistent mass memory

Sample file system:

- Samba (SMB)
or Common Internet File System (CIFS)
- Network File System (NFS)

Distributed file systems [\[edit\]](#)

See also: [Comparison of distributed file systems](#)

Distributed file systems are also called network file

- 9P, the Plan 9 from Bell Labs and Inferno distributed file system
- Amazon S3
- Andrew File System (AFS) is scalable and local
- Apple Filing Protocol (AFP) from Apple Inc.. A
- DCE Distributed File System (DCE/DFS) from Microsoft
- File Access Listener (FAL) is an implementation of AFS
- Microsoft Office Groove shared workspace, uses Groove File System
- NetWare Core Protocol (NCP) from Novell is used by NetWare
- Network File System (NFS) originally from Sun
- OS4000 Linked-OS provides distributed filesystem
- Secure File System (SFS)
- Self-certifying File System (SFS), a global network file system
- Server Message Block (SMB) originally from IBM for file sharing and authentication.

Network protocols

8

- Files accessed through **explicit** request/reply
- A **local copy** has to be made before accessing data
- Resource naming:
 - Uniform Resource Locator (URL)
 - `scheme://user:password@host:port/path`
 - `http://bob:bye@www.host.it:80/home/idx.html`
 - scheme = protocol name (http, https, ftp, file, jdbc, ...)
 - port = TCP/IP port number

HTTP Protocol

9

□ HyperText Transfer Protocol

- URL: <http://user:pwd@www.di.unipi.it>
- State-less connections
- Crypted variant: Secure HTTP (HTTPS)

□ Windows clients

□ Any browser

□ > wget

- GNU <http://www.gnu.org/software/wget/>
- W3C <http://www.w3.org/Library>

□ Linux clients

□ Any browser

□ > wget

FTP Protocol

10

□ File Transfer Protocol

- URL: <ftp://user:pwd@ftp.apa.unip.it/myfile>
- State-less connections
- Commands: get / put / mget
- Crypted variant: Secure FTP (SFTP)

□ Windows clients

- FTP: > ftp or any browser
- SFTP:
 - PuTTY <http://www.chiark.greenend.org.uk/~sgtatham/putty>
 - SSH Secure Shell <http://www.ssh.com>

□ Linux clients

- FTP: > ftp > sftp > gftp (GUI)

SCP Protocol

11

□ Secure Copy

- `> scp data.zip user@alice.cli.di.unip.it:datacopy.zip`
- File copy from/to a remote account
- File paths must be known in advance

□ Client

□ command line:

- `> scp/pscp` `> scp2`

□ Windows GUI

- WinSCP <http://winscp.sourceforge.net>
- SSH Secure Shell

□ Linux GUI

- SCP: default

Two issues

12

- **Where** are my files?
 - Local file systems
 - Distributed file systems
 - Network protocols

- Which **format** is data in?
 - Text
 - CSV, ARFF
 - XML
 - Binary, Compressed, ...

What is a file?

13

- File = sequence of bytes

67	73	65	79	10	83	10	...	
----	----	----	----	----	----	----	-----	--

How bytes are mapped to chars?

14

- Character set = alphabet of characters
- Coding bytes by means of a character set
 - ASCII, EBCDIC (1 byte per char)
 - UNICODE (1 / 2 / 4 bytes per char)

American Standard Code for Information Interchange

CODE	CHAR	CODE	CHAR	CODE	CHAR	CODE	CHAR	CODE	CHAR
0	NUL	26	SUB	52	4	78	N	104	h
1	SOH	27	ESC	53	5	79	O	105	i
2	STX	28	FS	54	6	80	P	106	j
3	ETX	29	GS	55	7	81	Q	107	k
4	EOT	30	RS	56	8	82	R	108	l
5	ENQ	31	US	57	9	83	S	109	m
6	ACK	32	SP	58	:	84	T	110	n
7	BEL	33	!	59	;	85	U	111	o
8	BS	34	"	60	<	86	V	112	p
9	HT	35	#	61	=	87	W	113	q
10	LF	36	\$	62	>	88	X	114	r
11	VT	37	%	63	?	89	Y	115	s
12	FF	38	&	64	@	90	Z	116	t
13	CR	39	'	65	A	91	[	117	u
14	SO	40	(	66	B	92	\	118	v
15	SI	41	)	67	C	93	]	119	w
16	DLE	42	*	68	D	94	^	120	x
17	DC1	43	+	69	E	95	_	121	y
18	DC2	44	,	70	F	96	`	122	z
19	DC3	45	-	71	G	97	a	123	{
20	DC4	46	.	72	H	98	b	124	
21	NAK	47	/	73	I	99	c	125	}
22	SYN	48	0	74	J	100	d	126	~
23	ETB	49	1	75	K	101	e	127	DEL
24	CAN	50	2	76	L	102	f		
25	EM	51	3	77	M	103	g		

Text file = file+character set

16

- Text file = sequence of characters

C	I	A	O	\n	S	\n	...	
---	---	---	---	----	---	----	-----	--

Viewing text files

17

- By a text editor
 - Emacs, Notepad++, TextEdit, UltraEdit, Vi, etc.
- “Carriage return” character
 - Start a new line
 - Coding
 - Unix: 1 char ASCII(0A) ('\n' in Java)
 - Windows: 2 chars ASCII(0D 0A) (“\r\n” in Java)
 - Mac: 1 char ASCII(0D) ('\r' in Java)
 - Conversions
 - > **dos2unix**
 - > **unix2dos**

Text file = file+character set

18

- Text file = sequence of **lines**

C	I	A	O
S			
...			

Tabular data format

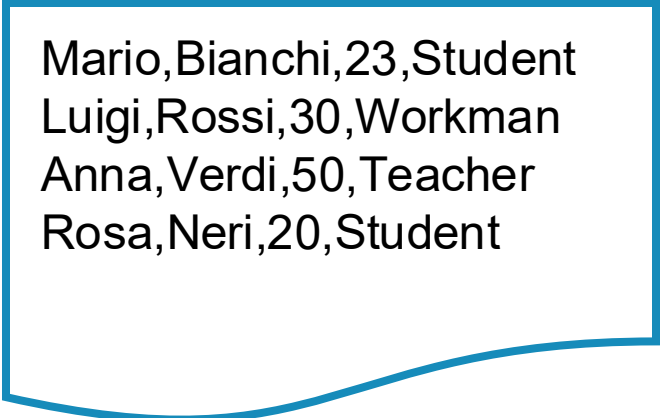
19

Row	Column			
	Mario	Bianchi	23	Student
	Luigi	Rossi	30	Workman
	Anna	Verdi	50	Teacher
	Rosa	Neri	20	Student

Representing tabular data in text files

20

- **Comma Separated Values (CSV)**
 - A row per line
 - Column values in a line separated by a special character
 - Delimiters: comma, tab, space



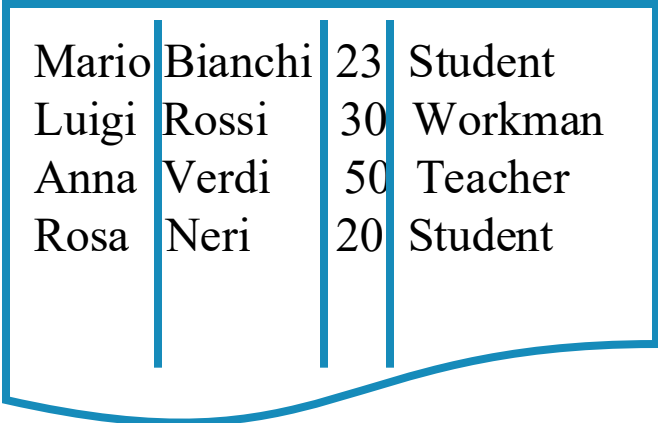
```
Mario,Bianchi,23,Student
Luigi,Rossi,30,Workman
Anna,Verdi,50,Teacher
Rosa,Neri,20,Student
```

Representing tabular data in text files

21

□ Fixed Length Values (FLV)

- A row per line
- Column values occupy a fixed number of chars
 - Allow for random access to elements
 - Higher disk space requirements



Mario	Bianchi	23	Student
Luigi	Rossi	30	Workman
Anna	Verdi	50	Teacher
Rosa	Neri	20	Student

Quoting

22

- What happens in CSV if a delimiter is part of a value?
 - Format error
- Solution: **quoting**
 - Special delimiters for start and end of a value (ex. “ ... “)

Mario Bianchi 23 Student
Luigi Rossi 30 Workman
Anna Verdi 50 Teacher
Rosa Neri 20 Student

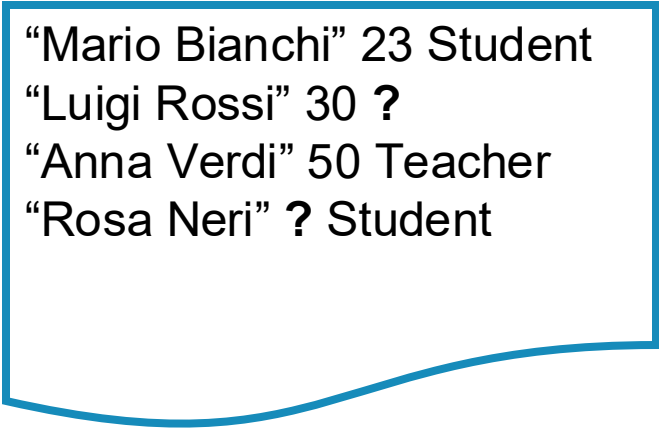


“Mario Bianchi” 23 Student
“Luigi Rossi” 30 Workman
“Anna Verdi” 50 Teacher
“Rosa Neri” 20 Student

Missing values

23

- How to represent missing values in CSV or FLV?
 - A reserved string: “?”, “null”, “”



“Mario Bianchi” 23 Student
“Luigi Rossi” 30 ?
“Anna Verdi” 50 Teacher
“Rosa Neri” ? Student

Meta-data

24

- Describe properties of data
 - Table name, column name, column type

name	surname	age	occupation
string	string	int	string
Mario	Bianchi	23	Student
Luigi	Rossi	30	Workman
Anna	Verdi	50	Teacher
Rosa	Neri	20	Student

Meta-data: ARFF data types

25

- ARFF (Attribute-Relation File Format)
 - ◆ real / integer / numeric
 - they are synonyms and cover numeric types
 - ◆ String
 - covers strings of any length
 - ◆ { name-1, ..., name-n }
 - enumerated type
 - covers an enumeration of values
 - Ex., {high, medium, low} {Play, Don't Play}
 - ◆ date "yyyy-MM-dd HH:mm:ss"
 - date and time
 - Ex., "2001-04-03 12:12:12"

How to represent meta-data in text files?

26

- Two rows: names and types

name	surname	age	occupation
string	string	int	string



name,surname,age,occupation
string,string,int,string

How to represent meta-data in text files?

27

- n rows, with two columns: name and type

name	surname	age	occupation
string	string	int	string



name	type
name	string
surname	string
age	int
occupation	string



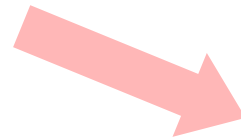
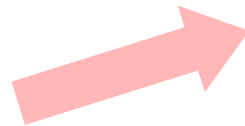
name,string
surname,string
age,int
occupation,string

Meta-data and data in text files

28

- Two distinct files
 - Eg., C4.5 format with .names and .data

name	surname	age	occupation
string	string	int	string
Mario	Bianchi	23	Student
Luigi	Rossi	30	Workman
Anna	Verdi	50	Teacher
Rosa	Neri	20	Student



name,string
surname,string
age,int
occupation,string

Mario,Bianchi,23,Student
Luigi,Rossi,30,Workman
Anna,Verdi,50,Teacher
Rosa,Neri,20,Student

Meta-data and data in text files

29

- In the same file
 - Meta-data first, then data

name	surname	age	occupation
string	string	int	string
Mario	Bianchi	23	Student
Luigi	Rossi	30	Workman
Anna	Verdi	50	Insegnante
Rosa	Neri	20	Studente



nome,cognome,eta',professione
string,string,int,string
Mario,Bianchi,23,Studente
Luigi,Rossi,30,Operaio
Anna,Verdi,50,Insegnante
Rosa,Neri,20,Studente

Meta-data and data in text files

30

- In the same file
 - Meta-data first, then data
 - A delimiter line may be required

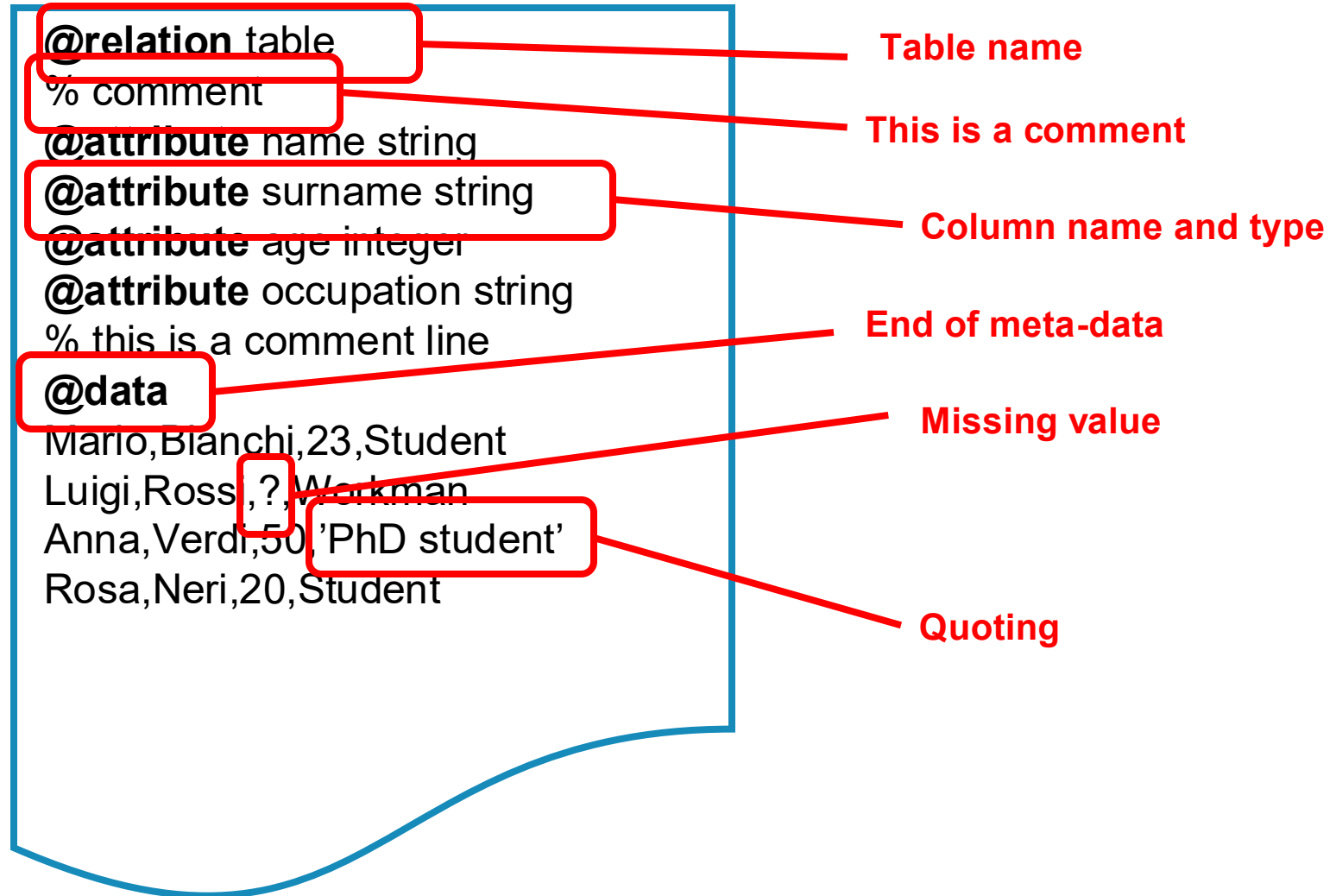
nome	cognome	eta'	professione
string	string	int	string
Mario	Bianchi	23	Studente
Luigi	Rossi	30	Operaio
Anna	Verdi	50	Teacher
Rosa	Neri	20	Student



```
name,string
surname,string
age,int
occupation,string
@data
Mario,Bianchi,23,Student
Luigi,Rossi,30,Workman
Anna,Verdi,50,Teacher
Rosa,Neri,20,Student
```

Weka ARFF format

31



The diagram illustrates the Weka ARFF format with a sample dataset. Red boxes highlight specific parts of the text, and red lines connect them to explanatory labels on the right. The sample data is as follows:

```
@relation table
% comment
@attribute name string
@attribute surname string
@attribute age integer
@attribute occupation string
% this is a comment line
@data
Mario,Blanchi,23,Student
Luigi,Ross,?,Workman
Anna,Verdi,50,'PhD student'
Rosa,Neri,20,Student
```

Annotations:

- Table name**: Points to `@relation table`.
- This is a comment**: Points to `% comment`.
- Column name and type**: Points to `@attribute surname string`.
- End of meta-data**: Points to `@data`.
- Missing value**: Points to `?` in the second data row.
- Quoting**: Points to the single quotes around `'PhD student'` in the third data row.

Two issues

32

- **Where** are my files?
 - Local file systems
 - Distributed file systems
 - Network protocols

- Which **format** is data in?
 - Text
 - CSV, ARFF
 - XML
 - Binary, Compressed, ...

Data representation in XML

33

- XML = **eXtensible Markup Language**
- XML allows for the definition of markup languages that represent structured data
 - Markup: marking, tagging, highlighting the meaning of a data element

The diagram shows a sample of text with various XML-like annotations. The text is: "Fourscore and seven years ago our fathers brought forth on this continent a new nation, conceived in liberty, and dedicated to the propositions that all men are created equal. Now we are engaged in a great civil war, testing whether that nation, or any nation". Handwritten annotations include: "enlarged font" pointing to the first 'F'; "Indent and bold, up to 'our'" pointing to the first sentence; "put in italics" pointing to "men are created equal."; "new paragraph" and "skip a line" pointing to the start of the second sentence; and "align text to both margins" pointing to the second sentence.

enlarged font
Fourscore and seven
years ago our fathers
brought forth on this
continent a new nation,
conceived in liberty,
and dedicated to the
propositions that all
men are created equal.
new paragraph
Now we are engaged in a
great civil war,
skip a line
testing whether that
nation, or any nation

Indent and bold, up to "our"
put in italics
align text to both margins

Why using markup languages?

34

- Problem: **data interchange** between applications
 - Proprietary data format do not allow for easy interchange
 - CSV with different delimiters, or column orders
 - Similar limitations of FLV, ARFF, binary data, etc.
- Solution:
 - definition of an interchange format...
 - ... marking data elements with their meaning ...
 - ... so that any other party can easily interpret them.

XML by example

35

```
<?xml version="1.0" encoding="UTF-8"?>
<Music>
  <CD number="1" >
    <song track="1">
      <artist>Iron Maiden</artist>
      <album>Killers</album>
      <year>1980</year>
      <title>The Ides of March</title>
      <length>1:55</length>
    </song>
    <!-- this is a comment -->
    <song track="4">
      <artist>Iron Maiden</artist>
      <album>Powerslave</album>
      <title>Another Life</title>
      <length>3:12</length>
    </song>
  </CD>
  ...
</Music>
```

Prologue: XML declaration

36

```
<?xml version="1.0" encoding="UTF-8"?>
```

- Mandatory at the beginning of the document
- Attributes:
 - *version*: (mandatory) XML version of the document.
 - *encoding*: (optional) character encoding (default: UTF-8)
 - *standalone*: (optional) if set to yes then the document does not refer to external documents (default: no)

Elements

37

- An **element** is a piece of data, delimited by and identified by a **tag name**.



Elements

38

- **Tag open** syntax :

<*name* *attributes***>**

- *name* is the name of the element.
- *attributes* is an *optional* list of attribute-values

- **Tag close** syntax:

</*name***>**

- *name* is the name of the element

- Elements with no content:

<*name* *attributes* **/>**

- There exists one and only one **root element**

Attributes

39

- They allow for specifying properties of elements using the syntax **attribute = “value”**

<name attribute="value">

- *<CD number="1" >*
- Attributes appear in the tag open
 - Order is not relevant
 - The “attribute or inner element?” dilemma

Text

40

- Reserved chars: '>', '<' and '&'
 - Meta-characters for reserved chars
 - `>` (greater-than sign: >);
 - `<` (less-than sign: <);
 - `&` amp (ampersand);
 - Character entities: 'à'
 - `à`
- CDATA sections
 - Bunch of textual data
 - `<![CDATA[ here any text with no XML meaning ]]>`

Tabular data, again

42

name	surname	age	occupation
string	string	int	string
Mario	Bianchi	23	Student
Luigi	Rossi	30	Workman
Anna	Verdi	?	Teacher
Rosa	Neri	20	Student

How to represent tabular data in XML?

43

□ Format “**Row**”

- an element `<row>` for every row, with an attribute for every non-missing column value

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <row name="Mario" surname="Bianchi" age="23" ocpt="Student" />
  <row name="Luigi" surname="Rossi" age="30" ocpt="Workman" />
  <row name="Anna" surname="Verdi" ocpt="Teacher" />
  <row name="Mario" surname="Bianchi" age="23" ocpt="Student" />
</root>
```

How to represent tabular data in XML?

44

- Format “**Elements**”
 - an element **<row>** with an inner element for every non-missing column value

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <row>
    <name>Mario</name>
    <surname>Bianchi</surname>
    <age>23</age>
    <ocpt>Studente</ocpt>
  </row>
  <row>
    <name>Luigi</name>
    <surname>Rossi</surname>
    <age>30</age>
    <ocpt>Operaio</ocpt>
  </row>
</root>
```

How to represent meta-data in XML?

45

- An element `<schema>` with an inner element `<attribute>` for every column

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <schema>
    <attribute name="name" type="string"/>
    <attribute name="surname" type="string"/>
    <attribute name="age" type="int"/>
    <attribute name="ocpt" type="string"/>
  </schema>
  <row name="Mario" surname="Bianchi" age="23" ocpt="Student" />
  <row name="Luigi" surname="Rossi" age="30" ocpt="Workman" />
  <row name="Anna" surname="Verdi" ocpt="Teacher" />
  <row name="Mario" surname="Bianchi" age="23" ocpt="Student" />
</root>
```

ARFF+XML = XRFF

46

- eXtensible attribute-Relation File Format
- XML version of ARFF
 - with additional column data types

```
- <dataset name="iris" version="3.5.3">
- <header>
- <attributes>
  <attribute name="sepalength" type="numeric" class="no" />
  <attribute name="sepalwidth" type="numeric" class="no" />
  <attribute name="petallength" type="numeric" class="no" />
  <attribute name="petalwidth" type="numeric" class="no" />
- <attribute class="yes" name="class" type="nominal">
- <labels>
  <label>Iris-setosa</label>
  <label>Iris-versicolor</label>
  <label>Iris-virginica</label>
</labels>
</attribute>
</attributes>
</header>
- <body>
- <instances>
  - <instance type="normal">
    <value missing="no">5.1</value>
    <value missing="no">3.5</value>
    <value missing="no">1.4</value>
    <value missing="no">0.2</value>
    <value missing="no">Iris-setosa</value>
  </instance>
```