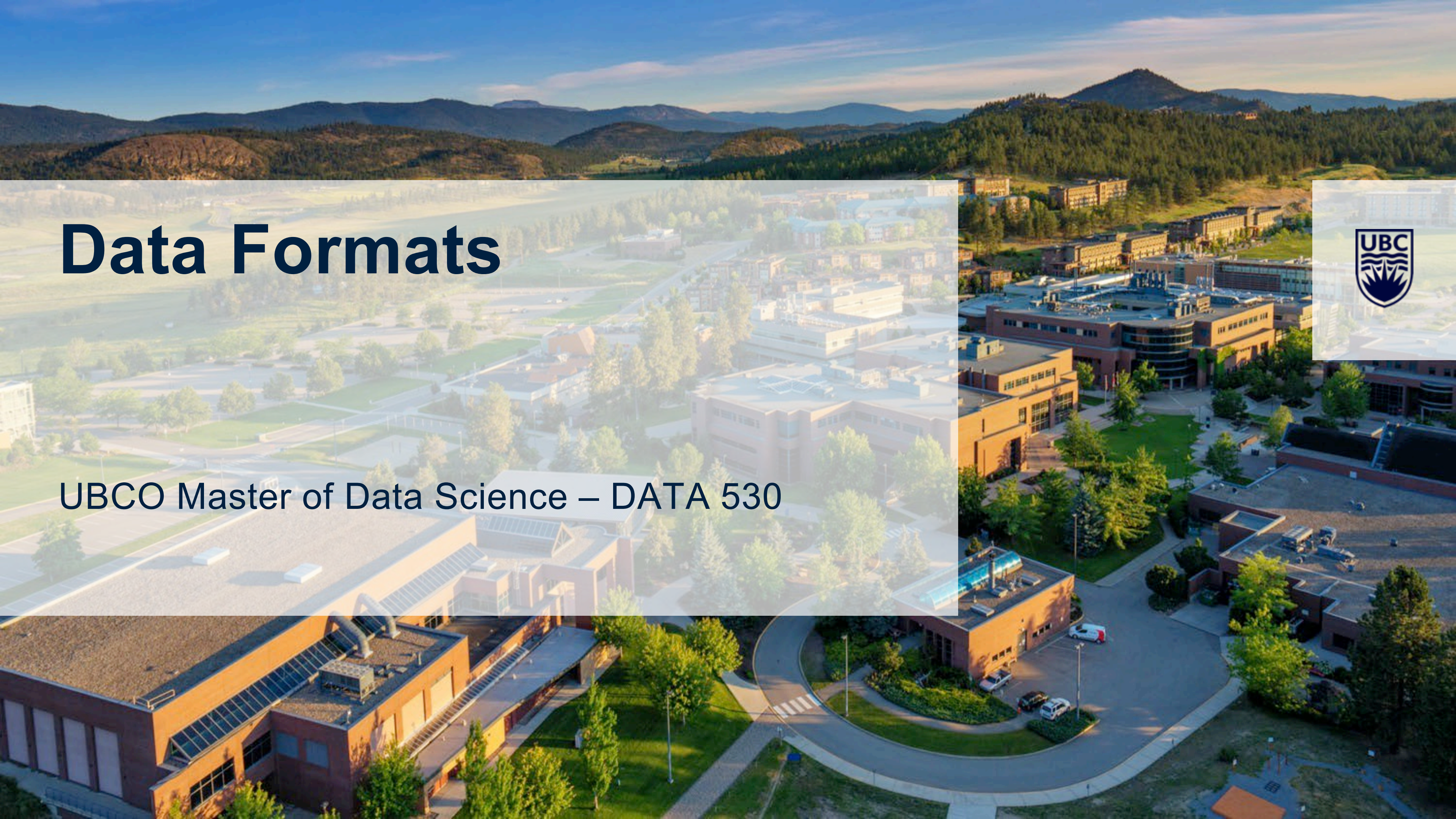
Data Formats

UBCO Master of Data Science – DATA 530



Learning Objectives

- Explain why it is important to understand and use correct terminology.
- Define: computer, software, memory, data, memory size/data size, cloud
- Explain "Big Data" and describe data growth in the coming years.
- Compare and contrast: digital versus analog
- Briefly explain how integers, doubles, and strings are encoded.
- Explain why ASCII table is required for character encoding.
- Explain why Unicode is used in certain situations instead of ASCII.
- Explain the role of metadata for interpreting data.
- Define: file, file encoding, text file, binary file
- Encode using the NATO broadcast alphabet.
- Discuss the time-versus-space tradeoff.

Basic Computer Terminology

A ***computer*** is a device that can be programmed to solve problems.

- ***Question:*** Is a cell phone a computer? **A)** yes **B)** no

Software is programs the computer follows to perform functions.

Memory is a device which allow the computer to store data either temporarily or permanently (data is preserved even when no power).

- Many different technologies for storing data with varying performance.
- ***Flash memory*** used in mobile devices (e.g. USB drives, phones) is permanent.
- ***Question:*** Does a hard drive store data permanently? **A)** yes **B)** no

"The Cloud"

"The Cloud" is not part of your computer but rather a network of distributed computers on the Internet that provides storage, applications, and services for your computer.

These systems and services simplify tasks that otherwise would be done by programs on your computer.

Examples:

- **Dropbox** is a cloud service that allows you to store your files on machines distributed on the Internet. Automatically synchronizes any files in folder with all your machines.
- **iCloud** is an Apple service that stores and synchronizes your data, music, apps, and other content across Apple devices.

What is Data?

Data is information before it has been given any context, structure and meaning.

Raw data is produced both by people during their interactions with computers (purchases, browsing, messaging) as well as by systems and sensors (logging, monitoring, automation).

How to Measure Data Size?

Data size is measured in bytes.

- Each **byte** contains 8 **bits** - a bit is either a 0 or a 1.
- A byte can store one character of text.

Larger units:

kilobyte (KB)	= 10^3 bytes
megabyte (MB)	= 10^6 bytes
gigabyte (GB)	= 10^9 bytes
terabyte (TB)	= 10^{12} bytes
petabyte (PB)	= 10^{15} bytes
exabyte (EB)	= 10^{18} bytes
zettabyte (ZB)	= 10^{21} bytes

Memory Size and Data Size

Memory size is a measure of memory storage capacity in bytes.

- It represents the *maximum* capacity of data in the device.

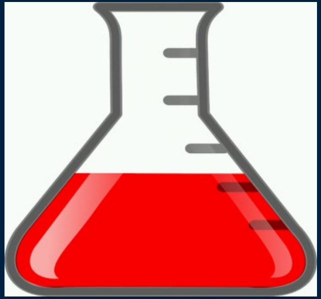
Question: Given this flask, assume the red liquid is data and each mark represents 100 MB of data. Select a true statement.

A) Memory size is 200 MB.

B) Flask can hold 0.5 GB of data.

C) Data size is about 200 KB.

D) Data size of 1000 KB would "overflow device".



Massive Growth of Data – "Big Data"

Big Data is the general term used to describe the explosion of data collected and the opportunities to transform this data into useful insights to benefit society.

- "Big" as the data size challenges how the data can be processed.

Data facts:

- Over 90% of the data in world history was created over the last few years.
- Estimated that 2.5 quintillion bytes (2.5 EB) generated per day.
- Global mobile data traffic estimated at 84.1 million TB per month in 2022.
    - Projected to reach 603.5 million TB per month by 2030
- Google processes about 3.5 billion requests/day and stores about 10 EB of data.
- Facebook collects 500 TBs/day (~2.5 billion items) and stores 100+ PB of photos.

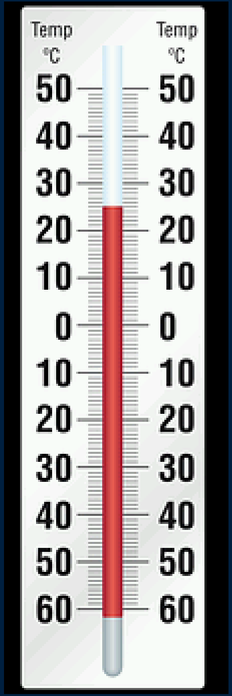
Representing Data on a Computer

Computers represent data ***digitally*** (in discrete bits).

The real-world is ***analog*** where the information is encoded on a continuous signal (spectrum of values).

Any data on a computer must be ***encoded*** as bits.

Analog versus Digital Thermometer Example



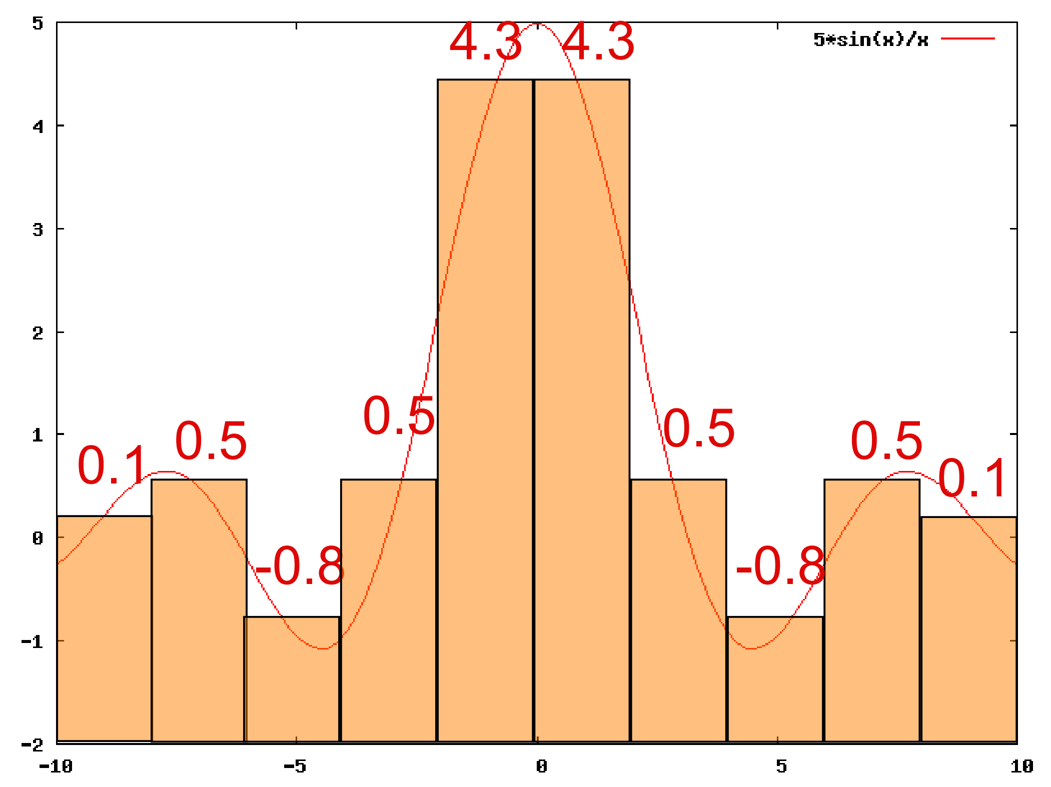
A thermometer contains liquid which expands and contracts in response to temperature changes.

The liquid level is analog, and its expansion continuous over the temperature range.

By adding marks and units to the thermometer, we are digitizing the information.

Conversion from Analog to Digital

How would you digitize this analog data into 10 discrete points?



Representing Data: Integers

An integer is a whole number. It is encoded in a computer using a fixed number of bits (usually 32 or 64).

- The first bit is a sign bit (0=positive, 1=negative).
- Negative numbers are represented in *two's complement notation*.
  - To get two’s complement, write out digit in binary, invert digits, add one to result

Example: 123,456,789 as a 32-bit integer:

Memory Address	0001	0002	0003	0004
	00000111	01011011	11001101	00010101

Example: 28 as an 8-bit integer: 00011100

-28 as an 8-bit integer: invert digits 11100011 then add 1

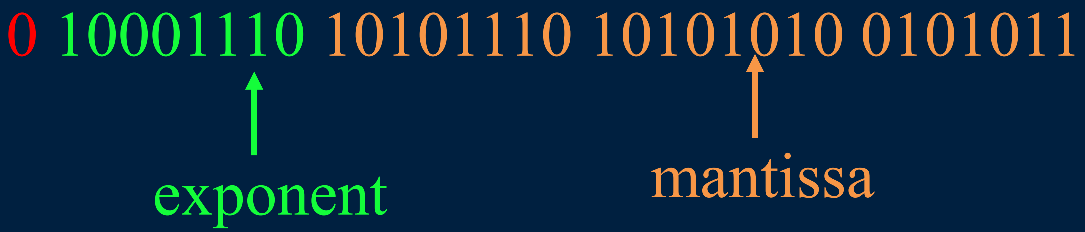
-28 = 11100100

Representing Data: Doubles and Floats

A number with a decimal may be either stored as a *double* (8 bytes) or *float* (4 bytes). Values are stored using a standard IEEE 754 format:

- Represent numbers in scientific format: N = m * 2^e
  - m - mantissa, e - exponent, 2 - radix
  - Note that converting from base 10 to base 2 is **not always precise**, since real numbers cannot be represented precisely in a fixed number of bits.

Example: The number 55,125.17 stored as 4 consecutive bytes is:



0 10001110 10101110 10101010 0101011

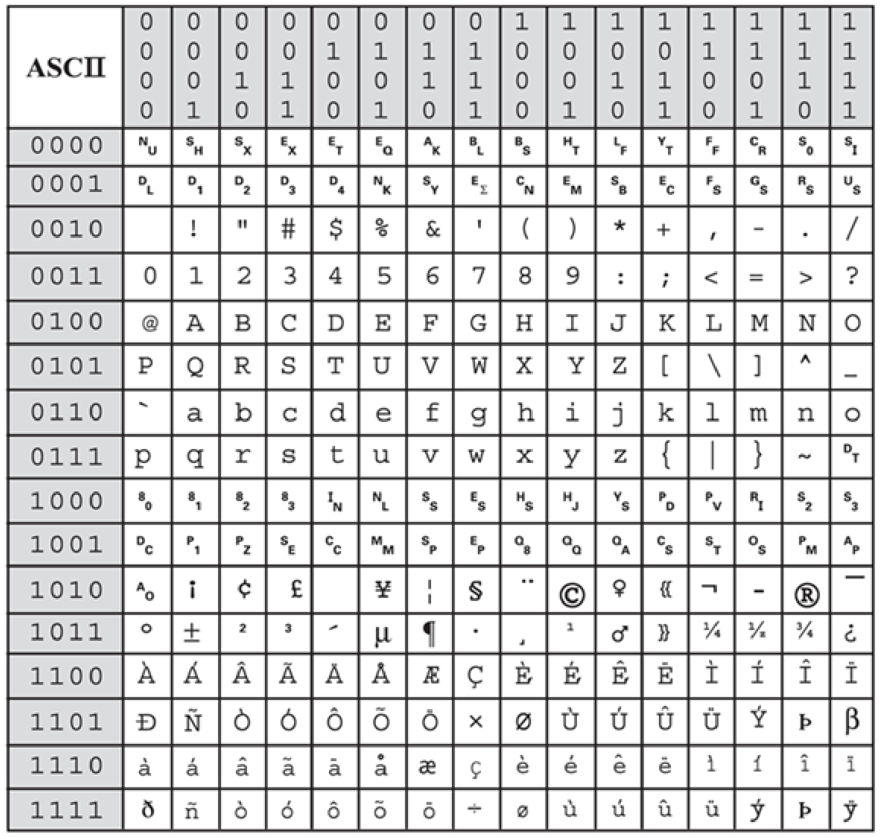
- Stored value is: 55125.168. Note the **lack of precision** which may be important in scientific applications.

Representing Data: Characters

A character is mapped to a sequence of bits using a *lookup* or *translation table*.

A common encoding is ***ASCII*** *(American Standard Code for Information Interchange)*, which uses 8 bits to represent characters.

ASCII Table



		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	ASCII	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0	0000	NU	SH	SX	EX	ET	EQ	AK	BL	BS	HT	LF	YT	FF	CR	S0	SI
1	0001	DL	D1	D2	D3	D4	NK	SY	EΣ	CN	EM	SB	EC	FS	GS	RS	US
2	0010		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0011	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	0100	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	0101	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	0110	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	0111	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DT
8	1000	80	81	82	83	IN	NL	SS	ES	HS	HJ	YS	PD	PV	RI	S2	S3
9	1001	DC	P1	P2	SE	CC	MW	SP	EP	Q8	QQ	QA	CS	ST	OS	PM	AP
A	1010	A0	¡	¢	£		¥	¦	§	¨	©	♀	«	¬	-	®	¯
B	1011	°	±	²	³	´	µ	¶	·	¸	¹	♂	»	¼	½	¾	¿
C	1100	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D	1101	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E	1110	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F	1111	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

Next 4 bits
(least significant)

First 4 bits
(most
significant)

Question:
Write your name in ASCII.

My name Oma =
01001111 01101101
01100001

ASCII Encoding

Question: What ASCII character is `0100 0100`?

A) `A`

B) `!`

C) `@`

D) `D`

ASCII Encoding

Question: What is `Test` encoded in ASCII?

A) 01110100 01100101 01110011 01110100

B) 01010100 01100101 01110011 01110100

C) 01000101 01010110 00110111 01000111

D) 01010100 01000101 01010011 01010100

Representing Text
Beyond ASCII - Unicode

Although ASCII is suitable for English text, many world languages, including Chinese, require a larger number of symbols to represent their basic alphabet.

The ***Unicode standard*** uses patterns of 16-bits (2 bytes) to represent the major symbols used in all languages.

- First 256 characters exactly the same as ASCII.
- Maximum # of symbols: 65,536.

Aside: Character Fonts

When a character is displayed on a screen, a particular font is used.

The font is how to present the data (the character).

Note that the font itself is data where each character has a mapping to a sequence of bytes representing pixels (image) on how to draw the character.

Representing Data: Strings

A ***string*** is a sequence of characters.

A string has a terminator to know when it ends:

- ***Null-terminated string*** - last byte value is 0 to indicate end of string.
- ***Byte-length string*** - length of string in bytes is specified (usually in the first few bytes before string starts).

Representing Data: Dates and Times

A ***date*** value can be represented in multiple ways:

- Integer representation - number of days past since a given date
  - Example: Julian Date (astronomy) – number of days since noon, January 1, 4713 BC
- String representation - represent a date's components (year, month, day) as individual characters of a string
  - Example: YYYYMMDD or YYYYDDD

A ***time*** value can also be represented in similar ways:

- Integer representation - number of seconds since a given time
  - Example: # of seconds since Thursday, January 1, 1970 (UNIX) (Year 2038 problem)
- String representation - hours, minutes, seconds, fractions
  - Example: HHMMSSFF

Encoding Other Data

We have seen how we can encode characters, numbers, and strings using only sequences of bits (and translation tables).

The documents, music, and videos that we commonly use are much more complex. However, the principle is exactly the same. We use sequences of bits and ***interpret*** them based on the ***context*** to represent information.

As we learn more about representing information, always remember that everything is stored as bits, it is by interpreting the context that we have information.

Metadata

Metadata is data that describes other data.

Examples of metadata:

- names of files
- column names in a spreadsheet
- table and column names and types in a database

Metadata helps you understand how to interpret and manipulate the data.

Files

A ***file*** is a sequence of bytes on a storage device.

- A file has a name.
- A computer reads the file from a storage device into memory to use it.

The operating system manages how to store and retrieve the file bytes from the device.

The program using the file must know how to interpret those bytes based on its information (e.g. metadata) on what is stored in the file.

File Encoding

A ***file encoding*** is how the bytes represent data in a file.

A file encoding is determined from the file extension (e.g. .txt or .xlsx) which allows the operating system to know how to process the file.

- The extension allows the OS to select the program to use. The program understands how to process the file in its format.

File Encodings: Text Files

A ***text file*** is a file encoded in a character format such as ASCII or Unicode. These files are readable by humans.

There are many different text file encodings:

- CSV (comma-separated value) file – each line is a record, fields separated by commas
- tab-separated file – each line is a record, fields separated by tabs
- JSON (JavaScript Object Notation) file – data encoded in JSON format
- XML (Extensible Markup Language) file – data encoded in XML format

Data analytics will often involve processing text files.

File Encodings: Text File Examples

CSV (comma-separated) file:

```
Id,Name,Province,Balance
1,Joe Smith,BC,345.42
```

CSV (tab-separated) file:

```
Id  Name    Province  Balance
1   Joe Smith BC  345.42
```

JSON file:

```
{"Id":1, "Name":"Joe Smith", "Province":"BC", "Balance":345.42}
```

XML file:

```
<customers><customer>
   <id>1</id>   <name>Joe Smith</name>
    <province>BC</province> <balance>345.42</balance>
</customer></customers>
```

Question:
In these file encodings, what is data and what is metadata?

Note: XML and JSON are case-sensitive.

File Encodings: Binary File

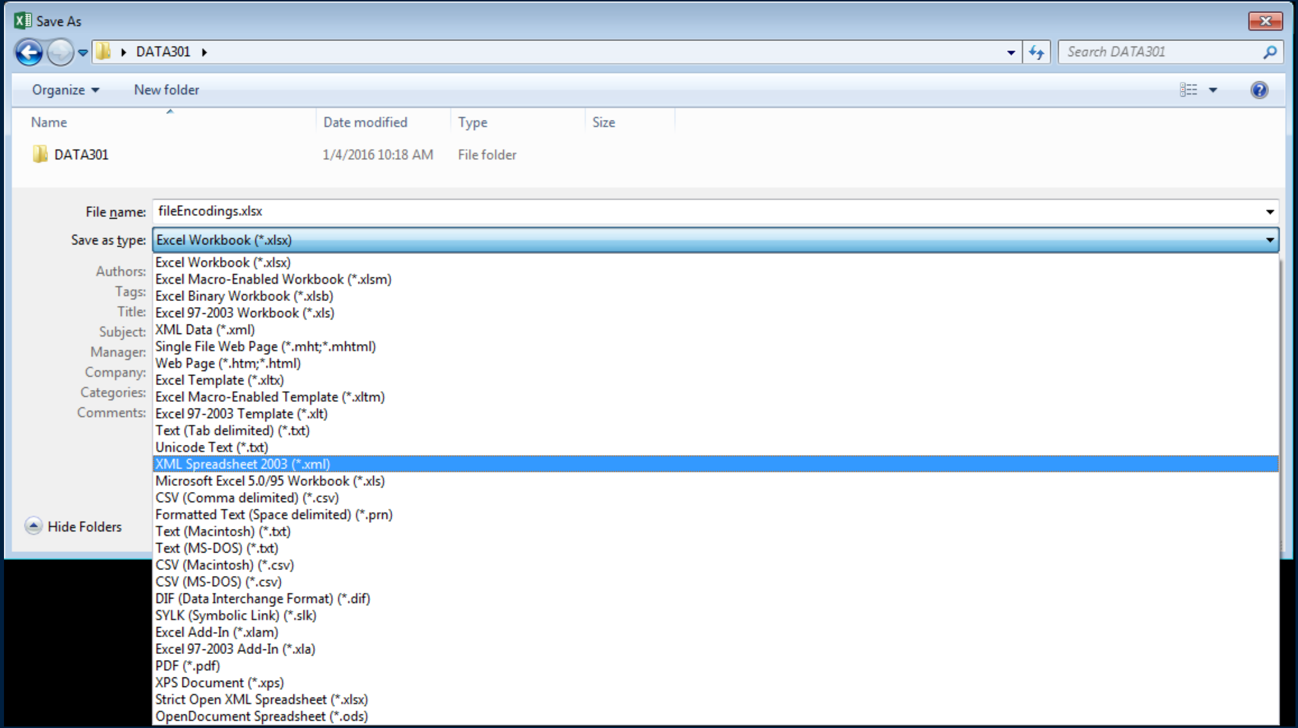
A ***binary file*** encodes data in a format that is not designed to be human-readable and is in the format used by the computer.

Binary files are often faster to process as they do not require translation from text form and may also be smaller.

Processing a binary file requires the user to understand its encoding so that the bytes can be read and interpreted properly.

Try it: File Encodings

Question: Use the `fileEncodings.xlsx` file and save the file as CSV, tab-separated, and XML. Look at each file in a text editor.



Name	Date modified	Type	Size
DATA301	1/4/2016 10:18 AM	File folder	

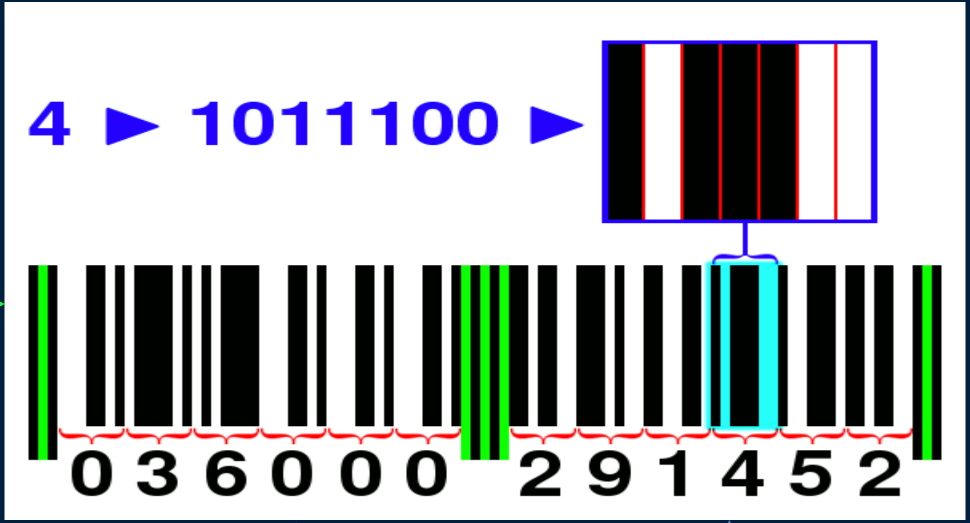
File name: fileEncodings.xlsx

Save as type: Excel Workbook (*.xlsx)

Save As
DATA301
Search DATA301
Organize
New folder
Authors:
Tags:
Title:
Subject:
Manager:
Company:
Categories:
Comments:
Hide Folders
Excel Workbook (*.xlsx)
Excel Macro-Enabled Workbook (*.xlsm)
Excel Binary Workbook (*.xlsb)
Excel 97-2003 Workbook (*.xls)
XML Data (*.xml)
Single File Web Page (*.mht;*.mhtml)
Web Page (*.htm;*.html)
Excel Template (*.xltx)
Excel Macro-Enabled Template (*.xltm)
Excel 97-2003 Template (*.xlt)
Text (Tab delimited) (*.txt)
Unicode Text (*.txt)
XML Spreadsheet 2003 (*.xml)
Microsoft Excel 5.0/95 Workbook (*.xls)
CSV (Comma delimited) (*.csv)
Formatted Text (Space delimited) (*.prn)
Text (Macintosh) (*.txt)
Text (MS-DOS) (*.txt)
CSV (Macintosh) (*.csv)
CSV (MS-DOS) (*.csv)
DIF (Data Interchange Format) (*.dif)
SYLK (Symbolic Link) (*.slk)
Excel Add-In (*.xlam)
Excel 97-2003 Add-In (*.xla)
PDF (*.pdf)
XPS Document (*.xps)
Strict Open XML Spreadsheet (*.xlsx)
OpenDocument Spreadsheet (*.ods)

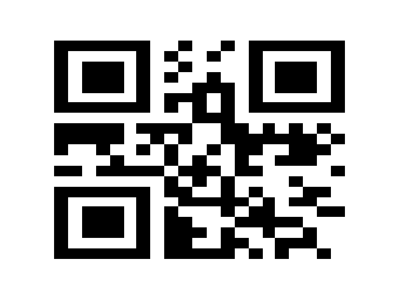
UPC Barcodes

Universal Product Codes (UPC) encode manufacturer on left side and product on right side. Each digit uses 7 bits with different bit combinations for each side (can tell if upside down).



QR Codes

A **QR** (**Q**uick **R**esponse) code is a 2D optical encoding developed in 1994 by Toyota with support for error correction.



Hello World!

Part of Syllabus

Make your own codes at: www.qrstuff.com.

NATO Broadcast Alphabet

The code for broadcast communication is purposefully inefficient, to be distinctive when spoken amid noise.

A	Alpha	J	Juliet	S	Sierra
B	Bravo	K	Kilo	T	Tango
C	Charlie	L	Lima	U	Uniform
D	Delta	M	Mike	V	Victor
E	Echo	N	November	W	Whiskey
F	Foxtrot	O	Oscar	X	X-ray
G	Golf	P	Papa	Y	Yankee
H	Hotel	Q	Quebec	Z	Zulu
I	India	R	Romeo		

Question: Broadcast your name to a partner using the NATO broadcast alphabet.

Advanced: The Time versus Space Tradeoff

A fundamental challenge in computer science is encoding information efficiently both in terms of space and time.

At all granularities (sizes) of data representation, we want to use as little space (memory) as possible. However, saving space often makes it harder to figure out what the data means (think of compression or abbreviations). In computer terms, the data takes longer to process.

The ***time versus space tradeoff*** implies that we can often get a faster execution time if we use more memory (space). Thus, we often must strive for a balance between time and space.

Review: Memory Size

Question: Which is bigger?

A) 10 TB

B) 100 GB

C) 1,000,000,000,000 bytes

D) 1 PB

Review: Metadata and Data

Question: Select a **TRUE** statement.

A) It is possible to have data without metadata.

B) Growth rates of data generation are decreasing.

C) A double value represents decimal numbers precisely on a computer.

D) A character encoded in Unicode uses twice as much space as ASCII.

Conclusion

All ***data*** is encoded as bits on a computer. ***Metadata*** provides the context to understand how to interpret the data to make it useful.

- Memory capacity of devices and data sizes are measured in bytes.

Files are sequences of bytes stored on a device. A ***file encoding*** is how the bytes are organized to represent the data.

- Text files (comma/tab separated, JSON, XML) are often processed during data analytics tasks. Binary files are usually only processed by the program that creates them.

As a data analyst, understanding the different ways of representing data is critical as it is often necessary to transform data from one format to another.

THE UNIVERSITY OF BRITISH COLUMBIA