PDST
Professiona Deve opment Service for Teachers | An tSeirbhís um Fhorbairt Ghairmiúi do Mhúinteoirí

An Roinn Oideachais
Department of Education

National Workshop 3

Day 2 Session 3

Overview of the Session

Part 1	Python libraries
Movement Break	
Part 2	NCCA examples
Movement Break	
Part 3	Curriculum planning

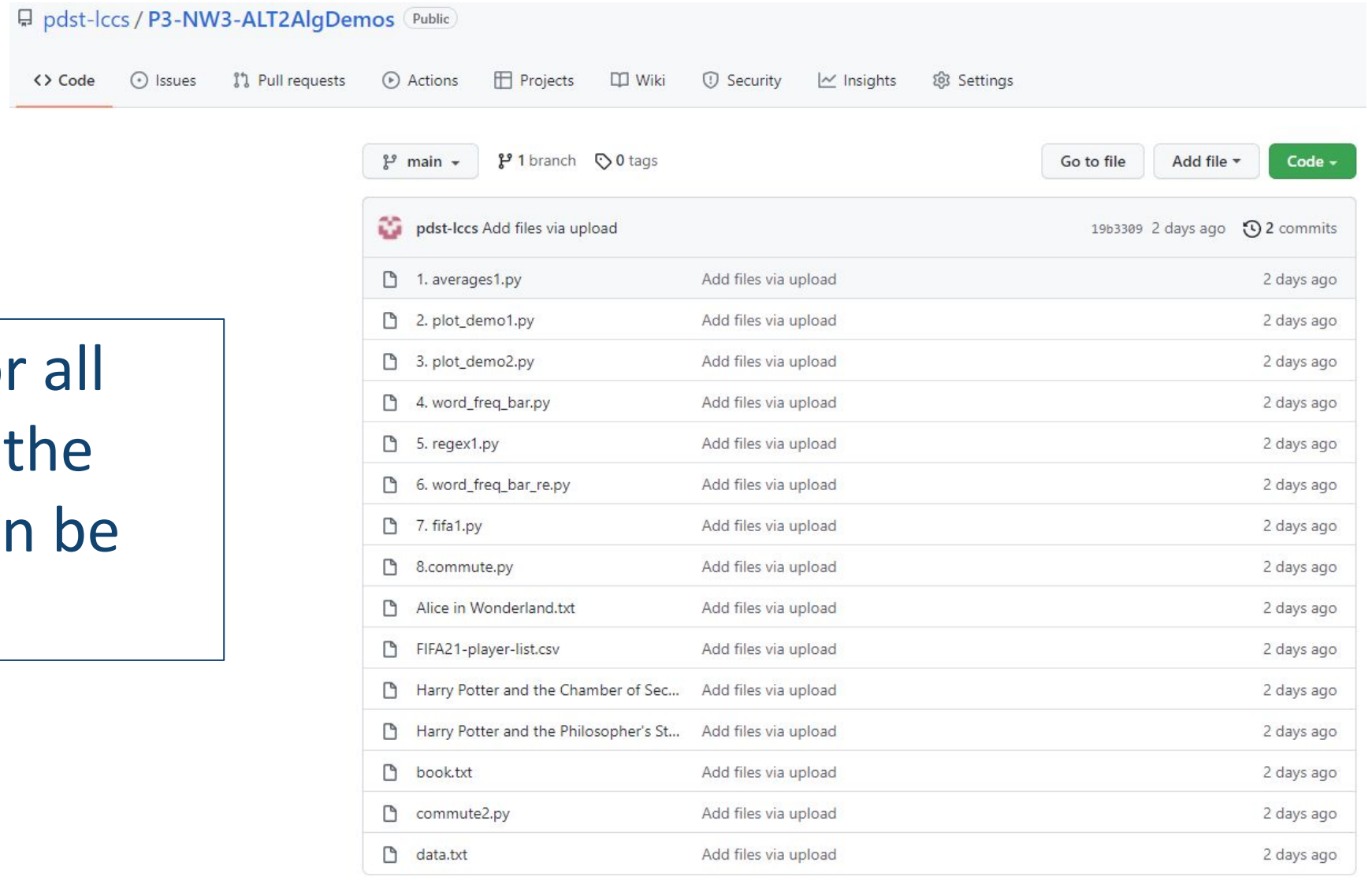
PDST
Professional Development Service for Teachers | An tSeirbhís um Fhorbairt Ghairmiúil do Mhúinteoirí

Section I

Python libraries for ALT2

GitHub

The source code for all the files shown on the upcoming slides can be found on GitHub



https://github.com/pdst-lccs/P3-NW3-ALT2AlgDemos

Measures of Central Tendency

```
# A simple program to calculate and display averages
from statistics import *

# Initialise a list of values
values = [2,3,5,2,4]

# Compute the 3 averages
arithmetic_mean = mean(values)
median_value = median(values)
modal_value = mode(values)

# Display the answers
print("The mean is ", arithmetic_mean)
print("The median and mode are %d and %d" %(median_value, modal_value))
```

When the program is run the output looks like this:

```
The mean is  3.2
The median and mode are 3 and 2
>>>
```

Measures of Central Tendency

Check out the online documentation

Averages and measures of central location

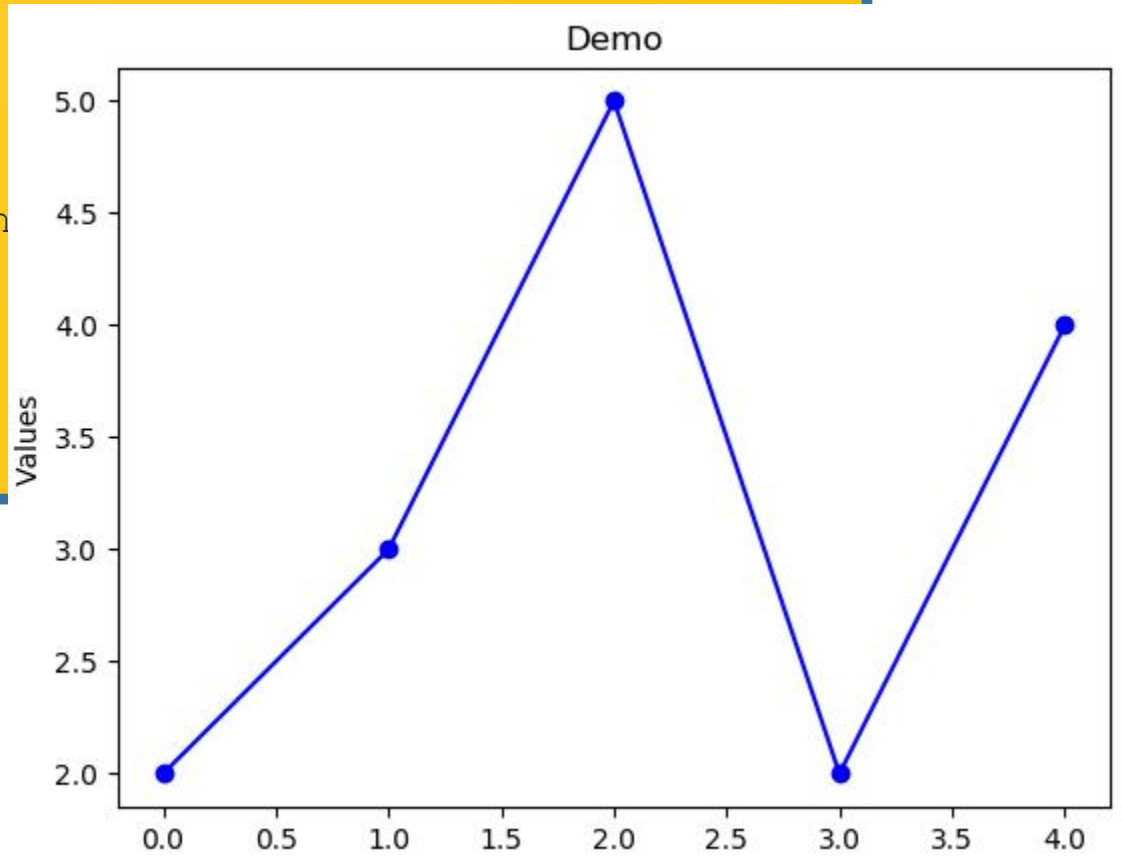
These functions calculate an average or typical value from a population or sample.

`mean()`	Arithmetic mean ("average") of data.
`fmean()`	Fast, floating point arithmetic mean.
`geometric_mean()`	Geometric mean of data.
`harmonic_mean()`	Harmonic mean of data.
`median()`	Median (middle value) of data.
`median_low()`	Low median of data.
`median_high()`	High median of data.
`median_grouped()`	Median, or 50th percentile, of grouped data.
`mode()`	Single mode (most common value) of discrete or nominal data.
`multimode()`	List of modes (most common values) of discrete or nomimal data.
`quantiles()`	Divide data into intervals with equal probability.

https://docs.python.org/3/library/statistics.html

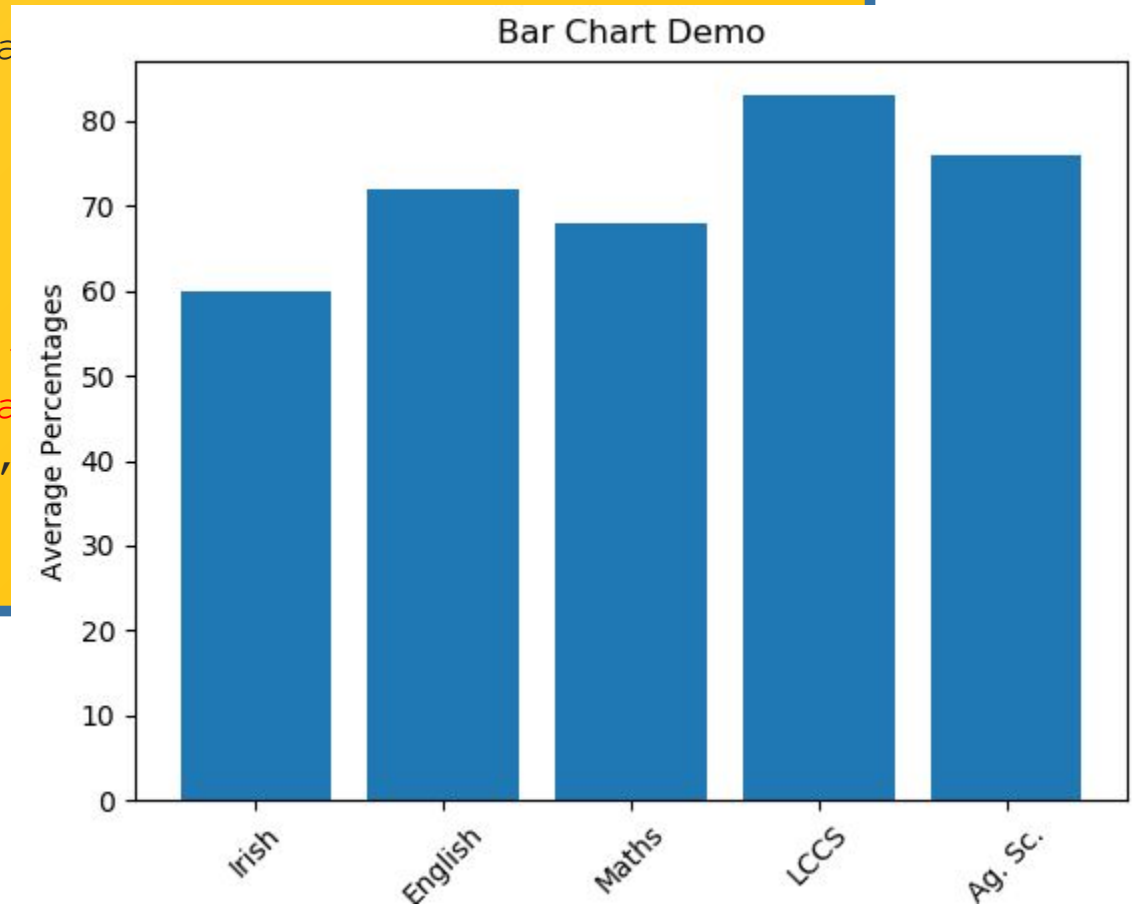
Demonstration of `matplotlib`

```
# A simple program to demonstrate use of matplotlib
from matplotlib import pyplot as plt

# Initialise a list of values
values = [2,3,5,2,4]

# Intervals for the x-axis
x_axis = [0, 1, 2, 3, 4]

plt.plot(x_axis, values, color='blue', lin

plt.title("Demo") # graph title
plt.ylabel("Values") # label the y-axis
plt.show() # Display the plot
```



Demonstration of `matplotlib`

```
# A simple program to demonstrate use of matplotlib
from matplotlib import pyplot as plt

# Initialise a list of subjects
subjects = ['Irish', 'English', 'Maths', 'LCCS', 'Ag. Sc.']

percentages = [60, 72, 68, 83, 76] # Avera

# Plot a bar chart
plt.bar(subjects, percentages)

plt.title("Bar Chart Demo") # graph title
plt.ylabel("Average Percentages") # label
# put the names of the subjects on the x-a
plt.xticks(range(len(subjects)), subjects,

plt.show() # Display the plot
```



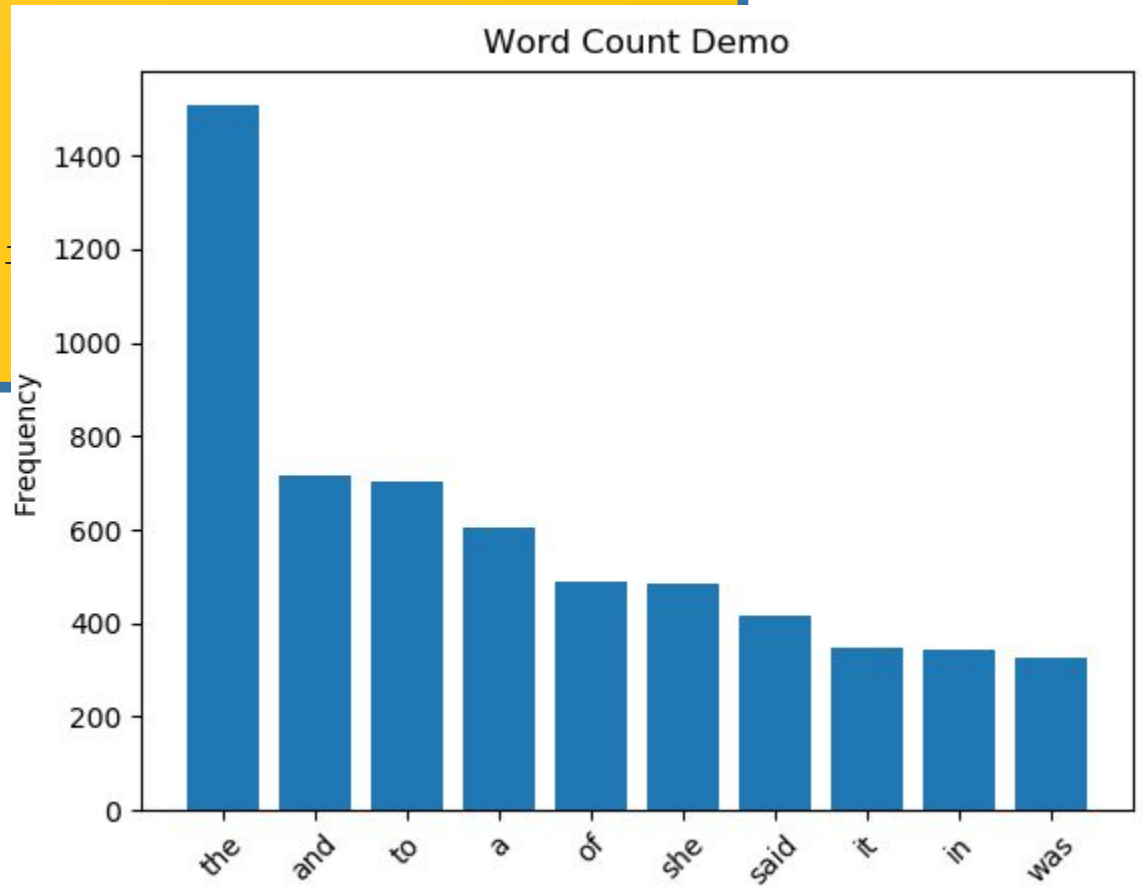
Text Analysis – word frequency

```
# A program to visualise the most common words in a file
from matplotlib import pyplot as plt
from collections import Counter

# IMPORTANT: Make sure book.txt exists in runtime directory
bookFile = open("book.txt","r") # Open the file
text = bookFile.read() # read the file
bookFile.close() # close the file
text_list = text.split() # create a list

# use counter to return the most common words
# format is .... [('the', 1507), ('and', 714), etc
most_common_words = Counter(text_list).most_common(10)

words = [] # an empty list of words
word_count = [] # an empty list of counts

# Build up the lists
for word, count in most_common_words:
    words.append(word) # append the word to the words list
    word_count.append(count)

# Now create and display the chart ….
```

Text Analysis – word frequency

… continued from previous slide

```
# Now create and display the chart ….

# Create the chart
plt.bar(words, word_count)
plt.title("Word Count Demo") # graph title
plt.ylabel("Frequency") # label the y-axis
# put the words on the x-axis
plt.xticks(range(len(words)), words, rotati
plt.show() # display the chart
```



Regular Expressions

A language that enables us to look for patterns in strings

```
import re

text1 = "THERE are 99 RED balloons"
print(re.sub('[0-9]', '', text1)) # remove digits
print(re.sub('[A-Z]', '', text1)) # remove uppercase
print(re.sub('[A-Z0-9]', '', text1)) # remove uppercase and digits
print(re.sub('[^a-z]', '', text1)) # leave lowercase
print(re.sub('[^a-zA-Z ]', '', text1)) # leave letters and spaces
print(re.sub('[^a-zA-Z0-9]', ' ', text1)) # leave letters and digits
print(re.sub(r'\b\w{1,4}\b', '', text1)) # remove words of length 1-3

text1 = "$%**$%joe*&$%^&"
print(re.sub('[^a-zA-Z0-9]', '', text1))
```

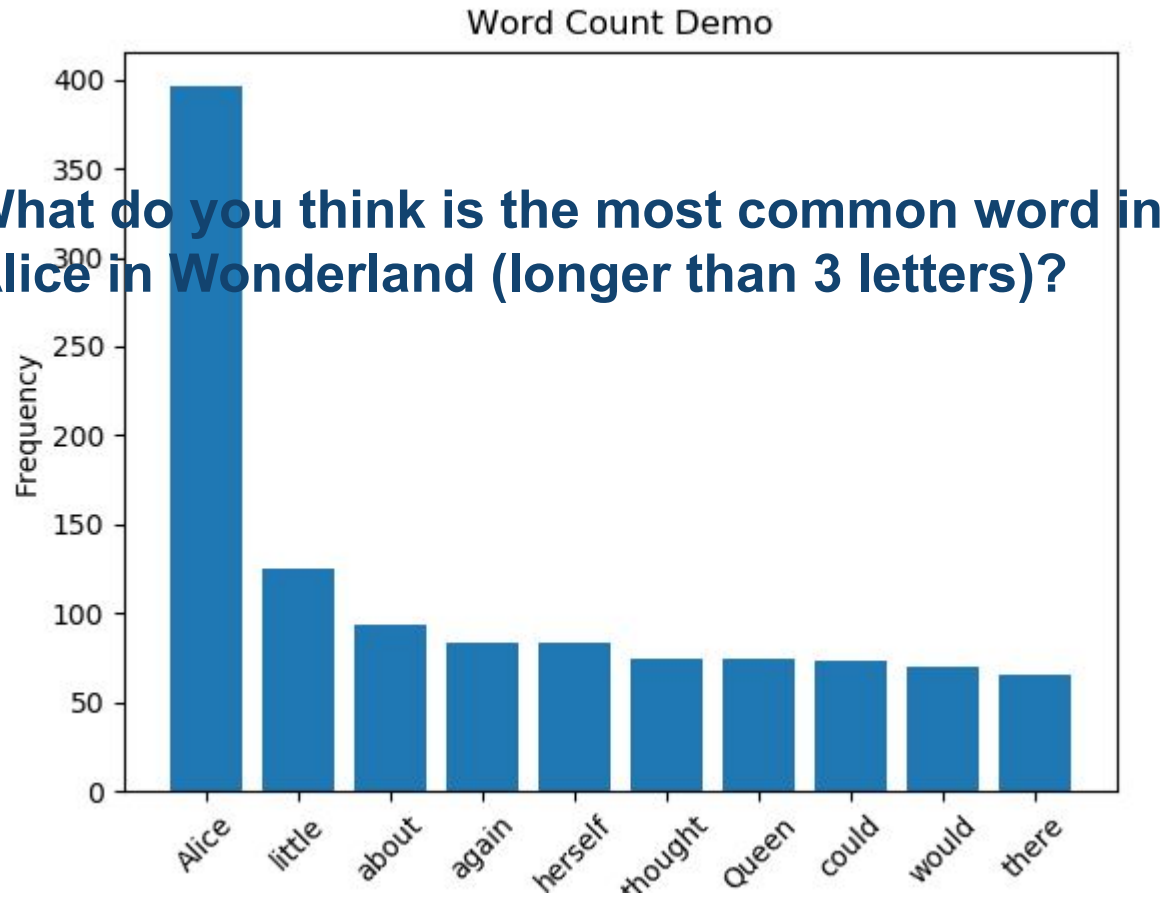
Output

```
THERE are  RED balloons
 are 99  balloons
 are   balloons
areballoons
THERE are  RED balloons
THERE are 99 RED balloons
THERE    balloons


joe
```

Text Analysis – word frequency

Eliminate words of three letters or less … use Regular Expressions

```
# A program to visualise the most common words in a file
from matplotlib import pyplot as plt
from collections import Counter
import re

# IMPORTANT: Make sure book.txt exists in ru
bookFile = open("book.txt","r") # Open the fi
text = bookFile.read() # read the file
bookFile.close() # close the file

text = re.sub('[^a-zA-Z0-9 \n]', ' ', text)
text = re.sub(r'\b\w{1,4}\b', '', text)

text_list = text.split() # create a list

# Continue as before …
```



What do you think is the most common word in Alice in Wonderland (longer than 3 letters)?

Import the `re` library

Use the `sub` method

Pandas

Useful for very large files … this file was sourced on Kaggle

1	short_name	age	dob	height_cn	weight_kg	nationalit	club_nam	value_eur	wage_eur	player_po	preferred
2	L. Messi	33	24/06/1987	170	72	Argentina	FC Barcelc	67500000	560000	RW, ST, CF	Left
3	Cristiano Ronaldo	35	05/02/1985	187	83	Portugal	Juventus	46000000	220000	ST, LW	Right
4	J. Oblak	27	07/01/1993	188	87	Slovenia	AtlÃ©ticc	75000000	125000	GK	Right
5	R. Lewandowski	31	21/08/1988	184	80	Poland	FC Bayern	80000000	240000	ST	Right
6	Neymar Jr	28	05/02/1992	175	68	Brazil	Paris Saint	90000000	270000	LW, CAM	Right
7	K. De Bruyne	29	28/06/1991	181	70	Belgium	Manchest	87000000	370000	CAM, CM	Right

… …

18911	C. Pizarro	20	18/09/1999	176	70	Chile	UniÃ³n La	45000	500	CB	Right
18912	Shan Huanhuan	21	24/01/1999	185	70	China PR	Dalian YiF	50000	2000	ST	Right
18913	R. Dinanga	18	06/12/2001	182	73	Republic c	Cork City	45000	500	ST	Right
18914	J. Browne	19	10/09/2000	180	73	Republic c	Finn Harp:	45000	500	ST	Right
18915	P. McGarvey	16	02/08/2003	180	76	Republic c	Finn Harp:	30000	500	GK	Right
18916	Xie Xiaofan	22	15/03/1998	177	75	China PR	Jiangsu Su	45000	2000	CM	Right
18917	Wang Haijian	19	02/08/2000	185	67	China PR	Shanghai (	45000	1000	CM	Right
18918	A. Cetiner	18	20/07/2001	175	70	Republic c	Shelbourr	40000	500	CM	Right
18919	Huang Jiahui	19	07/10/2000	186	74	China PR	Dalian YiF	40000	1000	CB	Right
18920	A. Phelan	19	20/06/2001	176	72	Republic c	Waterforc	40000	500	CM	Right
18921	J. Akintunde	24	29/03/1996	175	75	England	Derry City	40000	550	ST	Right

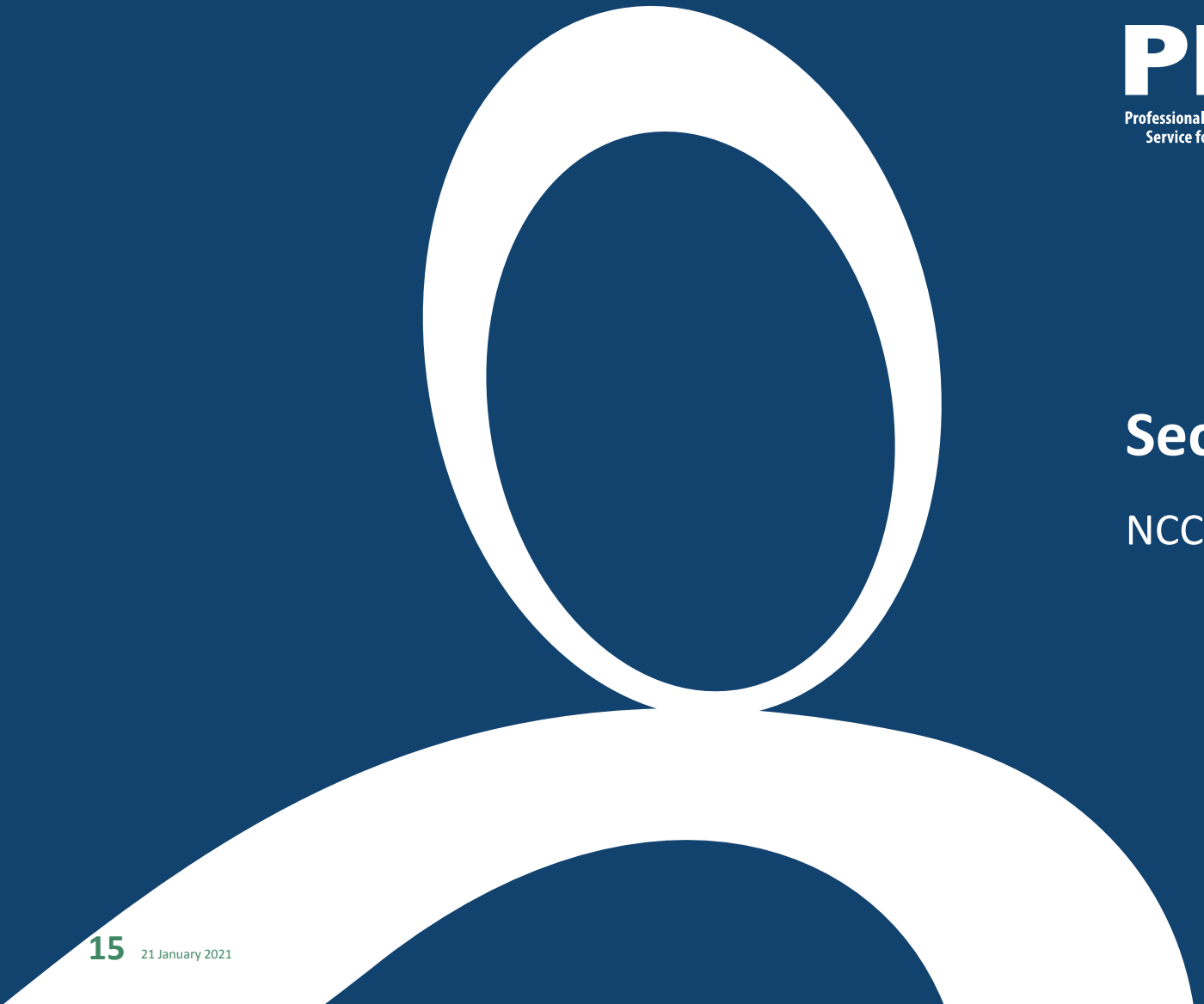
Let’s explore the player’s value

Pandas

```
# Using pandas - recommended for larger files
import statistics
import pandas

# Read the entire CSV file into a pandas DataFrame
df = pandas.read_csv('FIFA21-player-list.csv')

# Filter out the column, value_eur
player_values = df['value_eur']

# Compute and display the mean
mean_value = round(statistics.mean(player_values), 2)
print("Mean Value:", mean_value)

# Compute and display the median
median_value = statistics.median(player_values)
print("Median Value:", median_value)

# Compute and display the min and max values
print("Min: €%f, Max: €%f" %(min(player_values),max(player_values)))
```

Output looks like this:

```
Mean Value: 2224813.29
Median Value: 650000.0
Min: €0.000000, Max: €105500000.000000
```



PDST
Professional Development Service for Teachers | An tSeirbhís um Fhorbairt Ghairmiúil do Mhúinteoirí

Section II

NCCA examples

Demonstration of Samples

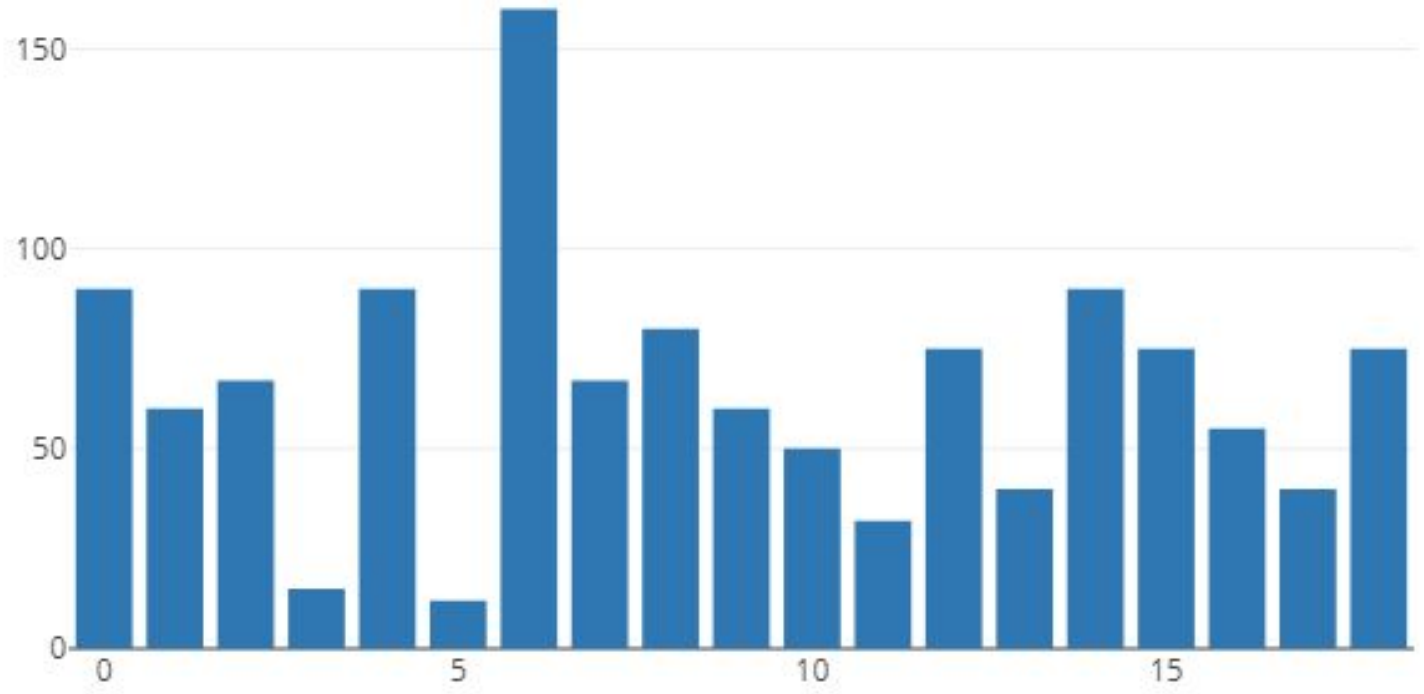
Commute Times

"Our topic is travel times, our data source are the other groups working and our hypothesis is that the average travel time will be 50 minutes and no one will have traveled for longer than 2 hours."

data.txt - Notepad

File Edit Format View Help

```
90
60
67 minutes
15
90
12 minutes
160
67 minutes
80
60 minutes
50
32
32
32
32
75
40 minutes
90 minutes
75 minutes
55 minutes
40
75 minutes
```



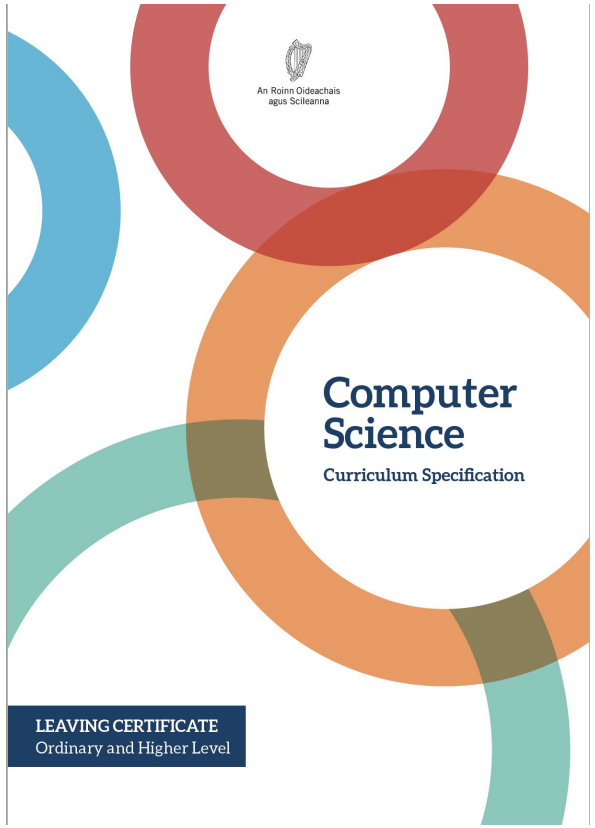
```
# Sample ALT2 - Commute times
import statistics
import re
import plotly.plotly
from plotly.graph_objs import Bar, Layout

# Open and read the data file
file = open("data.txt","r")
string = file.read()
file.close()

# Scrub the data
clean_string = re.sub(' minutes', '', string)
clean_string = re.sub(' ', '', clean_string)
string_array = clean_string.split('\n')

# Convert all the strings to integers
int_array = [int(i) for i in string_array]

# Determine and display the averages
mean_value = statistics.mean(int_array)
median_value = statistics.median_grouped(int_array, 1)
mode_value = statistics.mode(int_array)
print("Mean: %.2f, Median %d, Mode %d" %(mean_value, median_value, mode_value))

plotly.offline.plot({"data": [Bar(y=int_array)],
    "layout": Layout(title="word count")
})
```

PDST
Professional Development Service for Teachers | An tSeirbhís um Fhorbairt Ghairmiúil do Mhúinteoirí

Section III

Curriculum planning

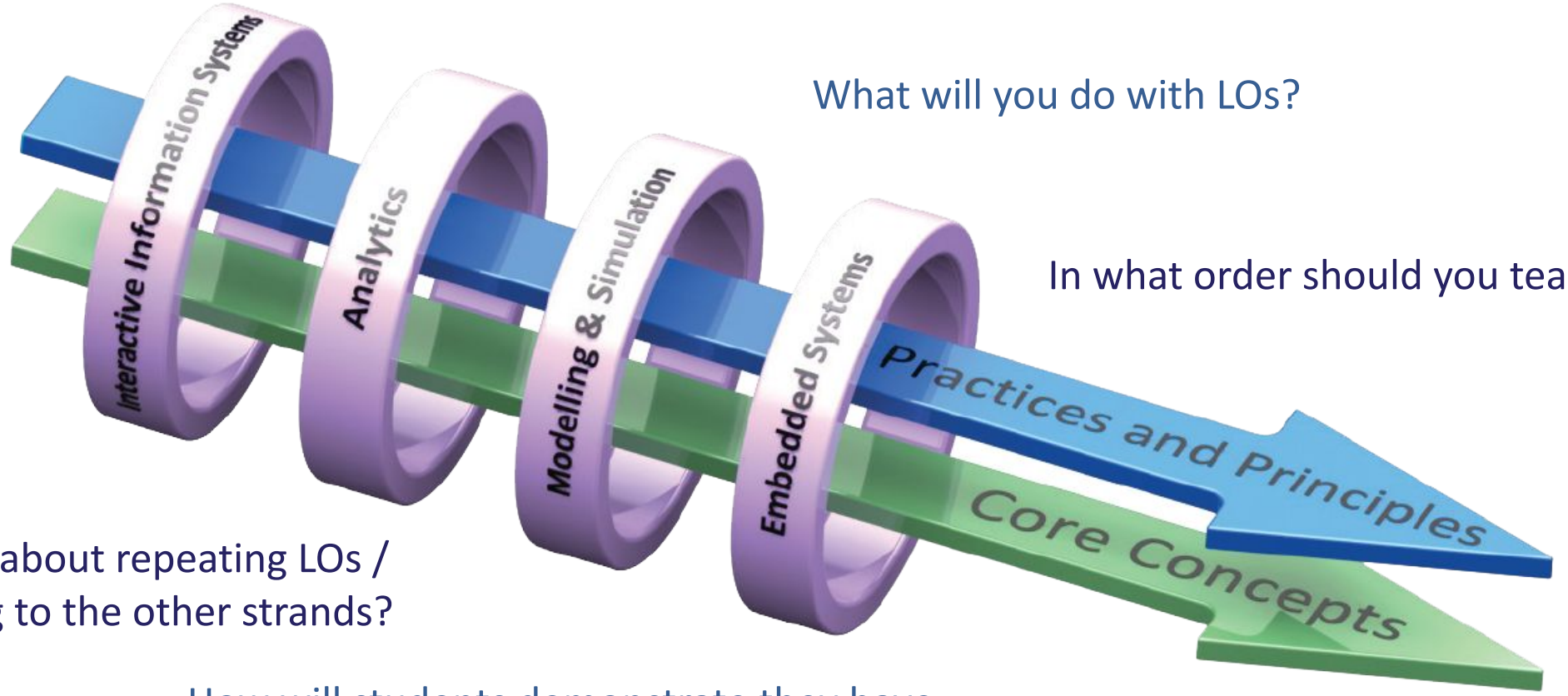
Considering curriculum planning



*'Learning outcomes can best be defined as statements of what a learner **knows**, **understands** and is **able to do** after completion of learning.'*

CEDEFOP (2009)

Considering curriculum planning



What will you do with LOs?

In what order should you teach them?

What about repeating LOs / linking to the other strands?

How will students demonstrate they have achieved the LOs?

What content or resources do you need?

Key message to remember:

Explore and teach the LOs through the lens of ALTs.

There are several ways to achieve this.

Group Activity

Develop a curriculum plan for January to April

Focus on ALT2

Group activity - instructions

Discuss your next steps in relation to curriculum planning.

Focus on ALT 2. Remember to teach the LOs through the lens of the ALTs - there are numerous ways to achieve this.

Consider topics, LOs, build up to ALT2, ALT2, equipment, resources, assessment, differentiation, etc.

What will you do with the LOs for ALT2?

In what order should you teach them?

What about repeating LOs / linking to the other strands?

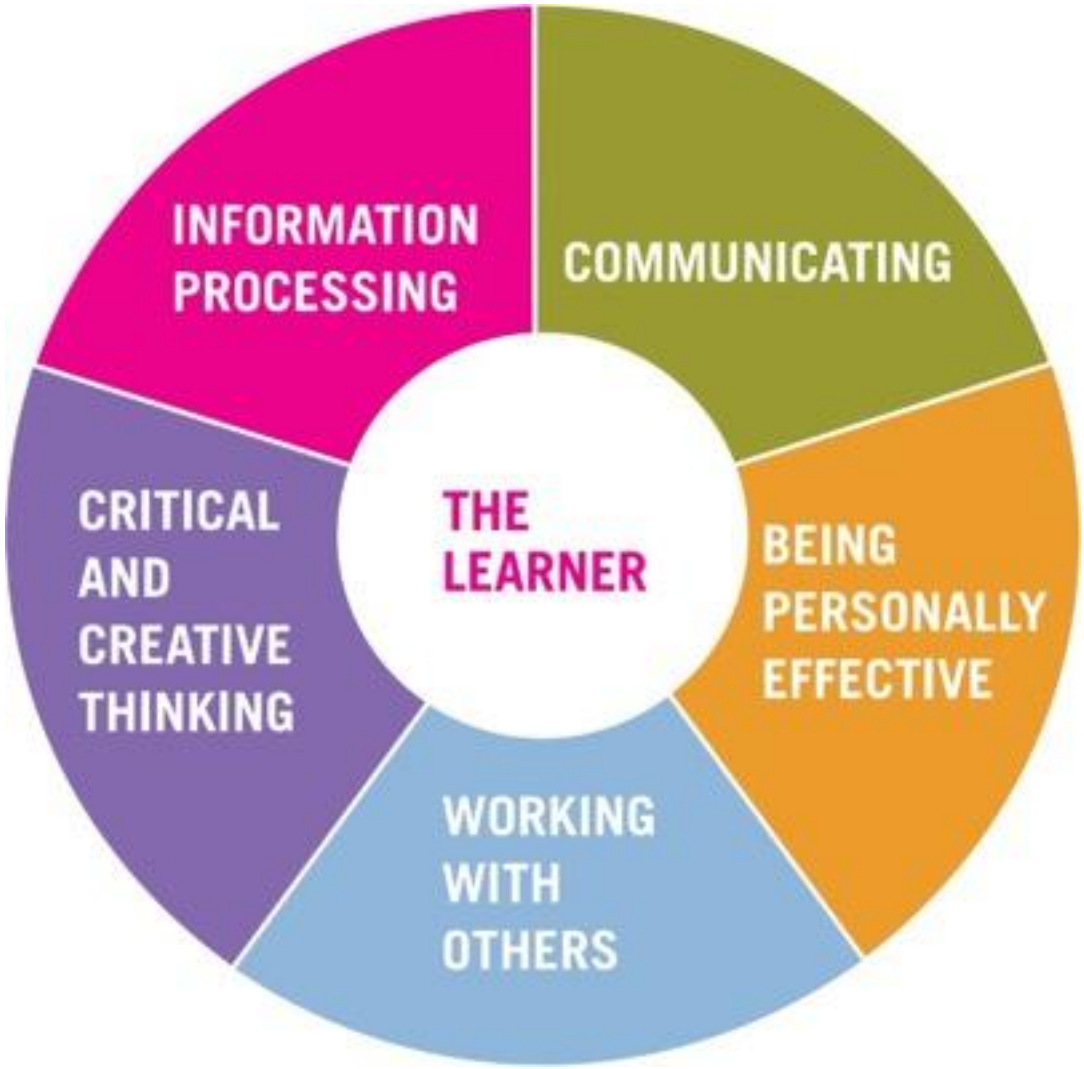
How will students demonstrate they have achieved the learning outcomes?

What content or resources will you need?

What can you include for Ordinary Level students?

Are there any considerations you should make for your students with SEN?

What about differentiation and extension of tasks?



Key Skills of Senior Cycle

LCCS Specification: p12

Group activity - instructions

Discuss your next steps in relation to curriculum planning.

Focus on ALT 2. Remember to teach the LOs through the lens of the ALTs - there are numerous ways to achieve this.

Consider topics, LOs, build up to ALT2, ALT2, equipment, resources, assessment, differentiation, etc.

Feedback

An Roinn Oideachais
Department of Education