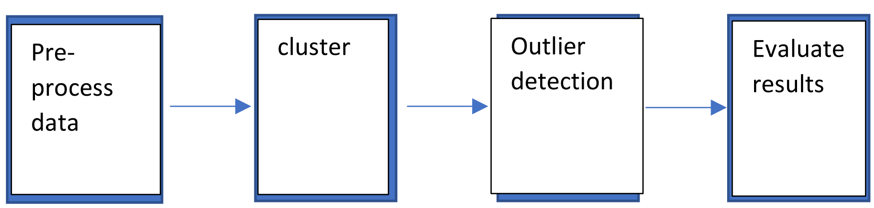
Question 1. Application scenario:

My example is an application that detects tax fraud. The data that would be input into the application, would a dataset of businesses and their reported profits/losses in each quarter, along with other key data such as the size of the company, the location, the sector it is operating in, how old the company is, and so on.

It is possible for businesses to be clustered by a clustering function, to place similar performing companies together in the same cluster. After this, an outlier detection function can be run on the new dataset of clusters. If there was a sharp drop in sales revenue in one company from that cluster, but not any of the others, it can be detected as an outlier. This outlier could have occurred for a number of reasons, such as human error, deliberate falsification of data for fraud, etc. It may be true that this company is under-reporting its sales to hide its taxable profits, moreover, if the rest of the metrics seem to remain unchanged, this becomes more likely. Another case could be that the company is over-reporting its profits, in order to secure more funding from investors. This is another fraudulent behaviour that can be detected by an anomaly detection function. Additionally, if there were a change to the industry that caused a drop in sales, this change would have to have been recorded amongst all the other companies in that cluster and so the precision with which the clusters are made is very important.

The next step would be to investigate this and maybe uncover whether it is tax fraud or not. In this way, the output of running a clustering function on the data, is used as an input for the outlier detection function, to detect outliers within each cluster. The program will only successfully function, if both of them are adequate and if the clusters formed by the first function are accurate enough to run a successful outlier detection function on.



Question 2.

2.1:

CMPT 459

Dist	P.J.T	B.L	
0-3	166	155	321
3-10	101	93	194
10-16	14	45	59
16-3pt	8	20	28
3pt	711	687	1398
	1000	1000	2000

Expected values.

Dist	P.J.T	B.L
0-3	160.5	160.5
3-10	97	97
10-16	29.5	29.5
16-3pt	14	14
3pt	699	699
	1000	1000

Dist	PJ.T	B.L
0-3	5.5^2	5.5^2
3-10	4^2	4^2
10-16	15.5^2	15.5^2
16-3pt	6^2	6^2
3pt.	12^2	12^2

Dist	PJ.T	B.L
0-3	0.1885	0.1885
3-10	0.1649	0.1649
10-16	8.1441	8.1441
16-3pt	2.5714	2.5714
3pt	0.2060	0.2060

adding up all values –
Chi-squared: 22.5499.

Degrees of freedom (D.F) = $(5-1) \times (2-1) = 4$.

Using df and chi-square we get p = 0.000155733
p is less than 0.05, we can determine that the 2 datasets are not independent. Players follow same distribution

Chi- square = 22.5499

P [obtained from tables] = 0.000155733.

Datasets are not independent and so the players follow the same distribution (p<0.05 significance level).

2.2:

kL- Divergence:

$$\int_{-\infty}^{\infty} p(t) \cdot \log\left(\frac{p(t)}{q(t)}\right) dt$$

Using Brook Lopez as base and by using python for calculations.

KL-divergence = 0.0929 diverging from Brook Lopez.

2.3:

KL divergence of 0.0929 tells us how much PJ Tucker diverges from Brook Lopez in shooting frequency. Chi-square tells us that the two datasets are not independent and shows how similar the two datasets are. Both values being small shows us that the two datasets are almost the same in terms of their distribution. The two players are very alike when it comes to their shooting.

Question 3:

3.1:

3231 unique words – D1

1889 unique words – D2

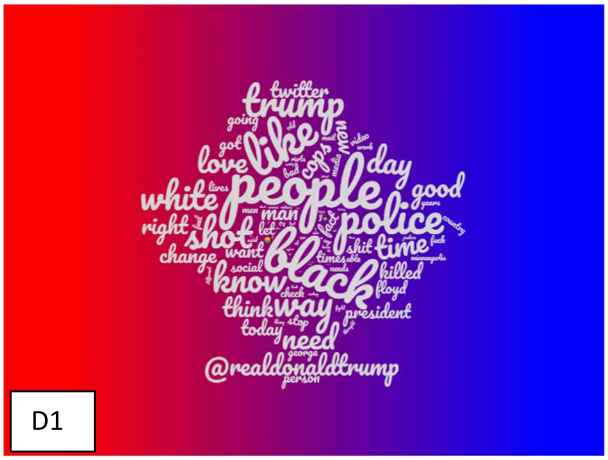
3.2:

D1	Token	Token Frequency			D2	Token	Token Frequency
	people	0.142				covid-19	0.255
	black	0.1				people	0.04
	like	0.065				pandemic	0.04
	police	0.062				deaths	0.036
	trump	0.048				covid	0.035
	shot	0.043				cases	0.031
	way	0.042				19	0.028
	know	0.041				trump	0.026
	white	0.041				death	0.025
	day	0.039				americans	0.024
	need	0.038				patients	0.023
	love	0.038				new	0.021
	time	0.037				need	0.02
	good	0.035				world	0.02
	#NAME?	0.033				nursing	0.02
	cops	0.033				health	0.02
	think	0.032				died	0.019
	new	0.031				crisis	0.019
	man	0.031				like	0.019
	right	0.031				asian	0.018
	change	0.03				social	0.017
	twitter	0.029				cuomo	0.017
	killed	0.028				u	0.017
	want	0.028				24	0.016
	today	0.028				hours	0.016
	president	0.027				got	0.016
	shit	0.026				spread	0.016
	got	0.026				covid19	0.016
	fact	0.025				homes	0.016
	let	0.025				democrat	0.015
		0.025				real	0.015
	going	0.025				virus	0.015
	floyd	0.025				positive	0.015

	social	0.024				attacks	0.015
	times	0.024				beginning	0.015
	person	0.023				george	0.015
	stop	0.023				floyd	0.015
	check	0.022				murder	0.015
	bad	0.022				hospital	0.015
	george	0.022				s	0.015
	lives	0.022				good	0.014
	2	0.021				okay	0.014
	men	0.021				black	0.014
	video	0.021				vaccine	0.013
	media	0.021				passionately	0.013
	country	0.02				speaking	0.013
	able	0.02				xenophobic	0.013
	needs	0.02				suddenly	0.013
	years	0.02				public	0.013
	feel	0.02				twitter	0.012
	fuck	0.02				president	0.012
	mail	0.019				want	0.012
	y’	0.019				country	0.012
	riots	0.019				house	0.012
	work	0.019				workers	0.012
	old	0.019				better	0.011
	minneapolis	0.019				state	0.011
	said	0.018				masks	0.011
	cop	0.018				look	0.011
	😂	0.018				acted	0.011
	justice	0.017				covid_19	0.011
	car	0.017				wear	0.011
	nt	0.017				record	0.01
	different	0.017				economy	0.01
	thought	0.017				patient	0.01
	fight	0.017				single	0.01
	thing	0.017				work	0.01
	look	0.016				coronavirus	0.01
	women	0.016				quickly	0.01
	best	0.016				broke	0.01
	seeing	0.016				tweets	0.01
	rioting	0.016				face	0.01
	care	0.016				order	0.01
	u	0.016				government	0.01
	violence	0.016				help	0.01
	😭	0.016				needs	0.01

	house	0.016				elderly	0.01
	history	0.015				watch	0.009
	federal	0.015				sharing	0.009
	10	0.015				dead	0.009
	thank	0.015				lockdowns	0.009
	free	0.015				recorded	0.009
	support	0.015				spike	0.009
	end	0.015				blame	0.009
	help	0.015				case	0.009
	life	0.015				viruses	0.009
	things	0.015				months	0.009
	❤️	0.015				racism	0.009
	family	0.014				tested	0.009
	brother	0.014				today	0.009
	killing	0.014				fact	0.009
	maybe	0.014				lives	0.009
	use	0.014				individuals	0.009
	protest	0.014				testing	0.009
	murdered	0.014				unable	0.009
	matter	0.014				texas	0.009
	kill	0.014				data	0.009
	state	0.014				continues	0.009
	🙏	0.014				states	0.009
	🏻	0.014				florida	0.009

3.3:



D1

D2

I made the cloud words using the online tool: https://www.wordclouds.com I imported a .csv file to the website containing the tokens and the frequency of the tokens. The size of the words is based on the how frequently the word has been used in my data set. The image on the left is from dataset D1, and the one on the right is of D2.

3.4:

The method I have used to compare the two data sets, is to make filtered data sets of my data, with a new D1 and D2 that do not contain any of the overlapping words in them. The third new data set is of all the overlapping words in D1 and D2. I gave these words a token frequency that is an average of their frequencies in their original respective datasets (for ex: 'trump' in D1= 0.5 and 'trump' in D2= 0.8; then token frequency in the overlapping dataset would be 0.65). Compared with the baseline, my method helps you to visualize what the common words are, making it more introspective. It does more than just putting two datasets together, as it also shows us the unique words in each dataset.