Task one: Distance and similarity calculation (10 points)

1. [5 points]

Given two objects represented by the tuples (22, 1, 42, 10) and (20, 0, 36, 8):

(a) Compute the *Euclidean distance* between the two objects.

(b) Compute the *Manhattan distance* between the two objects.

a) Euclidian distance : $\sqrt{\sum_{i=1}^{n} (a_i - b_i)^2}$

$a = (22, 1, 42, 10)$, $b = (20, 0, 36, 8)$

$d(a,b) = \sqrt{(22-20)^2 + (1-0)^2 + (42-36)^2 + (10-8)^2}$

$= \sqrt{2^2 + 1^2 + 6^2 + 2^2} = \sqrt{4+1+36+4}$

$= \sqrt{45}$

$= 3\sqrt{5}$

b) Manhattan distance : $\sum_{i=1}^{n} |a_i - b_i|$

$a = (22, 1, 42, 10)$, $b = (20, 0, 36, 8)$

$d(a,b) = (22-20) + (1-0) + (42-36) + (10-8)$

$= 2 + 1 + 6 + 2 = 11$

2. [5 points] Suppose we have the following data set:

document/term	t_1	t_2	t_3	t_4	t_5	t_6	t_7
d_1	0	4	10	8	0	5	0
d_2	5	19	7	16	0	0	32
d_3	15	0	0	4	9	0	17
d_4	22	3	12	0	5	15	0
d_5	0	7	0	9	2	4	12

Find the similarity between documents d_1 and d_2 using cosine similarity.

$\cos(d_1, d_2) = \frac{(d_1 \cdot d_2)}{\|d_1\| \|d_2\|}$

$d_1 \cdot d_2 = 0*5 + 4*19 + 10*7 + 8*16 + 0*0 + 5*0 + 0*32$

$= 76 + 70 + 128 = 274$

$\|d_1\| = \sqrt{0^2 + 4^2 + 10^2 + 8^2 + 0^2 + 5^2 + 0^2} = 16 + 100 + 64 + 25 = \sqrt{205}$

$\|d_2\| = \sqrt{5^2 + 19^2 + 7^2 + 16^2 + 0^2 + 0^2 + 32^2}$

$= \sqrt{25 + 361 + 49 + 256 + 1024}$

$= \sqrt{1715}$

$\cos(d_1, d_2) = \frac{274}{\sqrt{(205)(1715)}} = 0.4621$

↳ not similar //

Task two: Probability and Bayes rule (20 points)

The following application of Bayes rule often occurs in actual medical practice. Suppose you have tested positive for a disease. What is the probability you actually have the disease? It depends on the sensitivity and specificity of the test, and on the prevalence (prior probability) of the disease.
We'll denote:

a positive test as Test = pos
a negative test as Test = neg
presence of disease as Disease = true
absence of disease as Disease = false

We know from clinical studies done on the test before FDA approval that the sensitivity and specificity of the test are:

p(Test = pos | Disease = true) = 0.95 (true positive rate, or sensitivity)
p(Test = neg | Disease = false) = 0.90 (true negative rate, or specificity)

From which we can also deduce:

p(Test = neg | Disease = true) = 0.05 (false negative rate)
p(Test = pos | Disease = false) = 0.10 (false positive rate)

We also know from public health surveys that the disease is relatively rare. The prevalence in the general population is:

p(Disease = true) = 0.01

From which we can deduce:

p(Disease = false) = 0.99

1. [18 points] Use Bayes rule to calculate *p*(Disease = true | Test = pos), i.e. the probability you actually have the disease, given the test was positive.

$P(\text{true} \mid \text{pos}) = ?$

$P(\text{true} \mid \text{pos}) = \frac{P(\text{pos} \mid \text{true})\, P(\text{true})}{P(\text{pos})}$

$P(\text{pos}) = P(\text{pos} \mid t)\, P(t) + P(\text{pos} \mid f)\, P(f)$

$0.95\ (0.01) = .0095$

$0.1\ (0.99) = .0990$

0.1085

$= \frac{0.95\ (0.01)}{0.1085}$

$= 0.08755 \simeq 8.75\%$

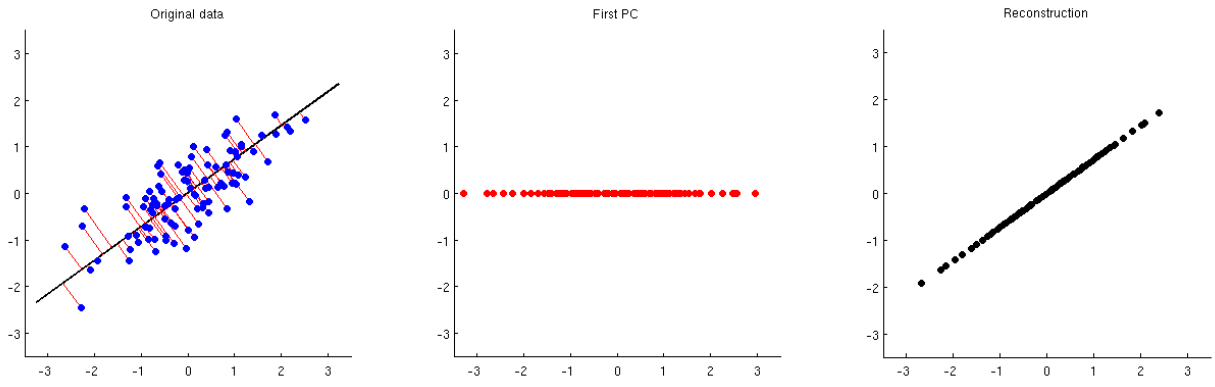
2. [2 points] Calculate the ratio *p*(Disease = true | Test = pos) / *p*(Disease = true). [In Bayesian statistics, a ratio like this is interpreted as the effect of new evidence on our beliefs about a probability. In this case, we are concerned with the probability of disease, and the new evidence is the test result.]

$$\frac{p(\text{true} \mid \text{pos})}{p(\text{true})} = \frac{0.08755}{0.01} = 8.755$$

Task three: PCA analysis simulation (25 points)

Let *X* be the *m*n* data matrix with *m* rows (data points) and *n* columns (centered variables, or features). Let *V* be the *n*p* matrix of *p* eigenvectors that with the largest *p* eigenvalues. Then the *m*p* matrix of PCA projections will be simply given by $Z = XV$.

This is illustrated in the figure below: the first subplot shows some simulated data points (and their projections on the first principal axis). The second subplot shows only the values of this projection on the new axis. The dimensionality of the data has been reduced from two to one:



In order to be able to reconstruct the original data points by using this one principal component, we can map it back to n dimensions with V^T. Indeed, the values of each PC should be placed on the same vector as was used for projection; compare subplots 1 and 3. The result is then given by $X\_reconstruction = ZV^T = XVV^T$, which is displayed on the third subplot above. **Please complete the code below using python**

(15 points), also write a short report to show the steps, results, plots and your thoughts on this task (10 points), you don't need to draw the projection trajectory (red lines as shown in the first subplot) but it's fine if you want.

[2 points] # generate 1000 randomly distributed data points with x and y two features/dimensions

```
#Creating pandas dataframe
data = pd.DataFrame(columns = ['x', 'y'], index = range(1, 1001))
#populating dataframe with data
for row in data.index:
    data.loc[row, 'x'] = np.random.normal(0, 1) #less variance
    data.loc[row, 'y'] = np.random.normal(0, 10) #more variance
data.head()
```

	x	y
1	-0.327571	7.07509
2	0.47992	8.43052
3	1.06336	-28.3915
4	0.396386	-3.31826
5	2.37286	3.15532

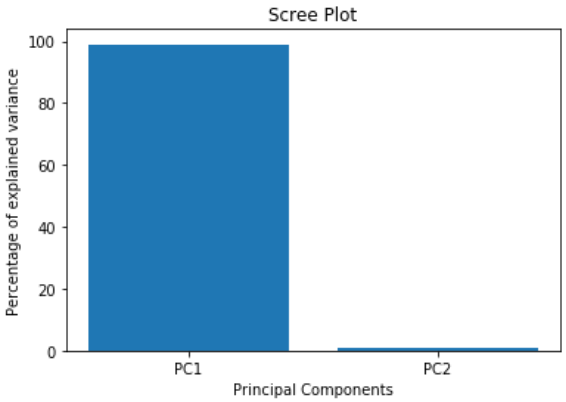
[10 points] # calculate the first principle component and project the data points onto it

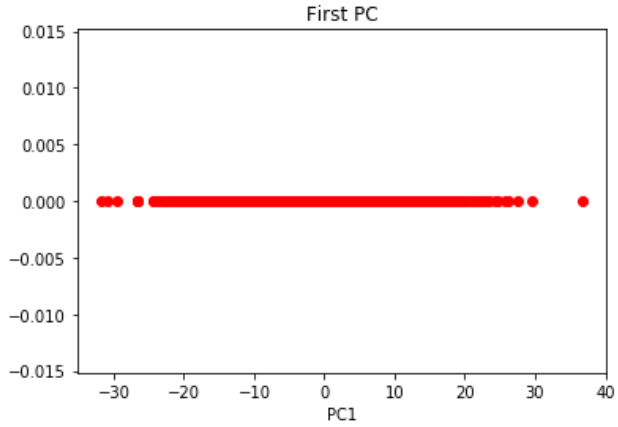
```
#Create PCA Object
pca = PCA()
pca.fit(data)
pca_data = pca.transform(data)

#Get Eigenvalues / PC Score
per_var = np.round(pca.explained_variance_ratio_ * 100, decimals = 1)
per_var #array([99.,  1.])

labels = ['PC'+str(x) for x in range(1, len(per_var) + 1)]
```

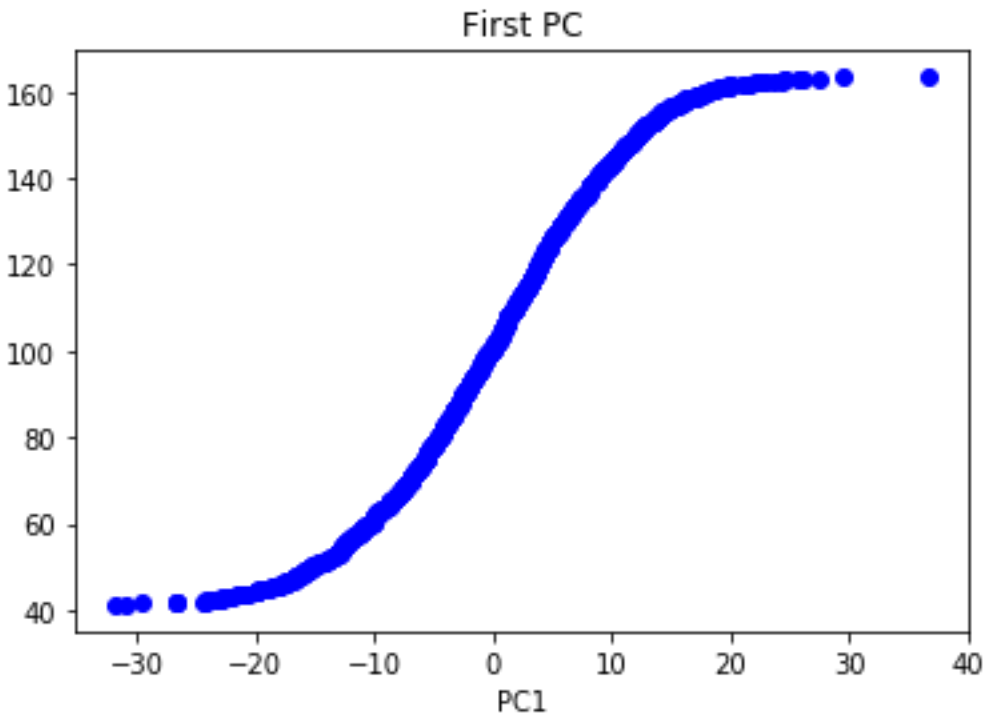
```
#Scree plot to look at percentage variance captured
plt.bar(x = range(1, len(per_var) + 1), height = per_var,
tick_label = labels)
plt.ylabel('Percentage of explained variance')
plt.xlabel('Principal Components')
plt.title('Scree Plot')
plt.show()
```





[3 points] # reconstruct the original points from this one principal component

```
#reconstruct PC onto original datapoints
y_new = np.linspace(np.min(y), np.max(y), len(y))
plt.scatter(np.sort(pca_data[:, 0]), y_new, color = 'blue')
plt.xlabel('PC1')
plt.title('First PC')
```


First PC
160
140
120
100
80
60
40
−30
−20
−10
0
10
20
30
40
PC1